Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité AUTOMATIQUE, SIGNAL, PRODUCTIQUE, ROBOTIQUE

Préparée au sein de l'Université de Rouen Normandie

Onboard/offboard extended perception for autonomous navigation

Présentée et soutenue par
ANTOINE CAILLOT

Thèse soutenue le 22/11/2022 devant le jury composé de		
MME VÉRONIQUE CHERFAOUI	PROFESSEUR DES UNIVERSITES, UNIV TECHNOLOGIE COMPIEGNE UTC COMPIEGNE	Rapporteur du jury
M. JEAN-PHILIPPE LAUFFENBURGER	PROFESSEUR DES UNIVERSITES, UNIVERSITE DE HAUTE ALSACE	Rapporteur du jury
M. FABIO MORBIDI	MAITRE DE CONFERENCES, UNIVERSITE AMIENS PICARDIE JULES VERNE	Membre du jury
MME SAFA OUERGHI	, ESIGELEC	Membre du jury
M. PASCAL VASSEUR	PROFESSEUR DES UNIVERSITES, UNIVERSITE AMIENS PICARDIE JULES VERNE	Membre du jury
M. THIERRY CHATEAU	PROFESSEUR DES UNIVERSITES, UNIVERSITE CLERMONT AUVERGNE CLERMONT AUVERGNE	Président du jury
M. YOHAN DUPUIS	DIRECTEUR DE RECHERCHE, CESI	Directeur de thèse
M. RÉMI BOUTTEAU	PROFESSEUR DES UNIVERSITES, Université de Rouen Normandie	Co-directeur de thèse

Thèse dirigée par YOHAN DUPUIS (INSTITUT DE RECHERCHE EN SYSTEMES ELECTRONIQUES EMBARQUES) et RÉMI BOUTTEAU (Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes)

UNIVERSITÉ DE ROUEN NORMANDIE

CESI ÉCOLE D'INGÉNIEURS

MIIS

irseem UR 4353

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat
Spécialité Automatique, signal, production, robotique
Préparée au sein de l'Université de Rouen Normandie

Onboard/offboard extended perception for autonomous navigation

Présentée et soutenue par
Antoine Caillot

Thèse soutenue publiquement le 22/11/2022

devant le jury composé de

Véronique CHERFAOUI	Professeure des Universités, Université de Technologie de Compiègne	Rapporteure
Jean-Philippe LAUFFENBURGER	Professeur des Universités, Université de Haute-Alsace	Rapporteur
Thierry CHATEAU	Professeur des Universités, Université Clermont Auvergne	Examinateur
Fabio MORBIDI	Maître de Conférences, Université Picardie Jules Verne	Examinateur
Safa OUERGHI	Enseignante-Chercheuse, ESIGELEC	Encadrante
Pascal VASSEUR	Professeur des Universités, Université Picardie Jules Verne	Encadrant
Rémi BOUTTEAU	Professeur des Universités, Université de Rouen Normandie	Codirecteur de thèse
Yohan DUPUIS	Directeur de recherche, CESI	Directeur de thèse

Thèse dirigée par Yohan DUPUIS (LINEACT CESI) et Rémi BOUTTEAU (Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes)

ESIGELEC ÉCOLE D'INGÉNIEURS-ES GÉNÉRALISTES

UNIVERSITÉ DE ROUEN NORMANDIE

irseem

ED 590 MIIS Mathématiques, Information, Ingénierie des Systèmes

Antoine Caillot: *Onboard/offboard extended perception for autonomous navigation* , November 2022

You have never been defeated until you give up.

— *The Weekly Democrat* of Natchez, Mississippi, 1910

Dedicated to *Michel Caillot*, *Serge Lainé*, *Jean-Pierre Levasseur* and *Cyrille Caillot*.

REMERCIEMENTS

Je tiens à remercier chaleureusement *Yohan DUPUIS* et *Rémi BOUTTEAU* d'avoir dirigé cette thèse ainsi que *Pascal VASSEUR* et *Safa OUERGHI* pour leurs encadrements. Merci à vous quatre pour votre confiance, votre disponibilité, vos conseils et votre aide pendant ces trois années parsemées de difficultés aussi diverses qu'improbables.

Je souhaite aussi remercier *Thierry CHATEAU* d'avoir présidé mon jury de thèse et *Fabio MORBIDI* pour son travail d'examinateur, *Véronique CHERFAOUI* et *Jean-Philippe LAUFFENBURGER* d'avoir accepté le travail de rapporteur et qui m'ont fourni des remarques pertinentes aussi bien sur mon manuscrit que sur mes travaux. Je vous remercie tous les quatre pour la bienveillance dont vous avez fait preuve envers moi lors de nos échanges.

Un grand merci à *Nicolas RAGOT* et *Guillaume CARON* de m'avoir offert cette opportunité professionnelle incroyable qui fut un moteur de motivation m'ayant permis de surmonter les difficultés.

Là encore, un très grand merci à mes ~~collègues~~ ***amis*** du département Systèmes Embarqués de l'ESIGELEC. Merci à tous pour votre accueil, votre gentillesse et pour tous nos moments de partages sur des sujets très variés. Merci à *Isabelle RIGIDEL*, aussi appelée "maman des doctorants", pour son soutien tant moral que technique. Je dédie une mention spéciale à *Louis LECROSNIER* et *Antoine MAURI* pour nos moments de complicité et pour leur soutien indéfectible, quelquefois dans des moments terriblement difficiles.

Enfin, je conclurai en adressant mes remerciements à ma famille et en particulier à mes parents qui m'ont donné les moyens de développer ma curiosité, d'explorer les domaines qui me passionnaient tout en m'encourageant toujours à donner le meilleur de moi-même. Merci à eux aussi pour leur accompagnement et leur soutien durant ces trois années. À ma mère, en première ligne de combat contre mes fautes de français, à mon père et ma sœur pour avoir su me changer les idées lorsque c'était nécessaire. Merci à tous de m'avoir permis d'aller jusqu'ici.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

GLOSSARY

2D	2-Dimensional vi, 11–13, 21, 22, 83, 84, 89–92, 107, 110, 120, 136
3D	3-Dimensional vi, 11–13, 19, 22, 83, 84, 90–92, 110, 120, 133, 136
AGPS	Assisted Global Positioning System (GPS) 33
AMCW	Amplitude Modulated Continuous-Wave 35
AOA	Angle Of Arrival 5, 8, 36, 52
BBA	Basic Belief Assigment 14, 86–88, 96, 98, 107, 109, 112–115, 124, 136
BP	Belief Propagation 61
BPA	Basic Probability Assigment 107, 109, 112–114, 124, 136
BSM	Basic Safety Messages 41, 42
C-ITS	Cooperative-ITS 6, 41
CAM	Cooperative Awareness Messages 41
CR	Correct Ratio 19, 20, 126, 127
CV	Connected Vehicle v, 85, 86, 130–133
DARPA	Defense Advanced Research Projects Agency 29
DENM	Distributed Environment Notification Messages 41, 43
DGPS	Differential GPS 33
DSRC	Dedicated Short-Range Communication viii, 6, 41–43

DST Dempster-Shafer Theory 13–21, 86, 87, 94, 95, 107, 109, 112–115, 124, 125, 127, 128, 131–133

EKF Extended Kalman Filter 8, 54, 55, 61

Emap Enhanced Maps 65

ETSI European Telecommunications Standards Institute 10, 66

FIM Fisher Information Matrix 61

FMCW Frequency Modulated Continuous-Wave 35

FN False Negative 126, 127

FOV Field of View 120

FP False Positive 126, 127

GCDC grand Cooperative Driving Challenge 29, 44, 47, 65, 74

GM General Motors 29

GMPHD Gaussian Mixture Probability Hypothesis Density 9, 61

GNSS Global Navigation Satellite System 4, 5, 8, 33, 36, 50, 54, 55

GOG Global Occupancy Grid 88, 94

GPS Global Positioning System 4, 7, 33, 35, 49–52, 75, 84, 85

GPS RTK Real-Time Kinematic GPS 4, 33

HD High Definition 69, 70

iCLCM i-GAME Cooperative Lane Change Message 41

ICP Iterative Closest Point 10, 69, 70

IMM Interacting Multiple Model 54

IMU	Inertial Measurement Unit 4, 5, 33, 35, 36, 75, 76
IoU	Intersection over Union ix, 15, 19–21, 126–128, 131, 132
IPM	Inverse Perspective Mapping 68, 89
ITS	Intelligent Transportation System 43
LDM	Local Dynamic Maps 10, 66, 69, 70
LiDAR	Light Detection and Ranging 4, 5, 8, 11, 34, 35, 44, 47, 49, 50, 52, 53, 68, 73, 74, 76, 136
LOG	Local Occupancy Grid 86, 88, 89, 92–94, 96
LUT	Look-Up Table viii, ix, 113, 114
mmWAVE	Millimeter Wave 5, 43, 44, 73
Mo-Cap	Motion Capture 7, 48, 49
NTP	Network Time Protocol 7, 48, 49
OBU	On-Board Unit 43
OG	Occupancy Grid 88, 92
PF	Particle Filter 8, 54
PoV	Point of View vi, 81, 83, 84, 95, 105, 107, 112, 115, 119–122, 124, 130, 131, 133
RADAR	Radio Detection and Ranging 4, 5, 8, 34, 37, 52, 59, 76, 136
RDS	Radio Data System 49
RMS	Root Mean Square 33, 34
RMSE	Root Mean Square Error 33, 34, 53
ROS	Robot Operating System 6, 42, 85, 92
RSSI	Received Signal Strength Indication 5, 8, 36, 75
RSU	Road Side Unit v, 12, 43, 44, 85, 86, 88, 107

SDN Software-Defined Network 42, 44, 73

SFM Structure From Motion 48

SLAM Simultaneous Mocalization And Mapping 4, 5, 34

SLAT Simultaneous Localization And Tracking 9, 61

SOOP Omnipresent Signals of Opportunity 61

SPaT Signal Phase and Timing 41, 42

SPOD Sparse Point-cloud Object Detection 75

SWOT Strengths, Weaknesses, Opportunities, and Threats 32, 73

TDOA Time Difference Of Arrival 5, 8, 36, 50, 52

TN True Negative 126, 127

TOA Time Of Arrival 5, 8, 36, 50

ToF Time of Flight 35

TP True Positive 126, 127

UTC Coordinated Universal Time 49

UWB Ultra Wide Band 5, 36, 43

V2I Vehicle-to-Infrastructure 11, 32, 72, 81, 83

V2V Vehicle-to-Vehicle 11, 32, 45, 61, 81, 83, 119

V2X Vehicle-to-Everything 30, 71

VANET Vehicular Ad-hoc Network viii, 5, 6, 41, 42

VIAC VisLab Intercontinental Autonomous Challenge 29

VLC Visible Light Communication 5, 43

WAVE Wireless Access in Vehicular Environments 5, 41, 43

Wi-Fi Wireless Fidelity 5, 43

YOLOv3 You Only Look Once Version 3 59

PERCEPTION ÉTENDUE EMBARQUÉE/DÉBARQUÉE POUR LA NAVIGATION AUTONOME

R

RÉSUMÉ EN FRANÇAIS

R.1 INTRODUCTION

Avec le développement du transport automobile et de l'informatique, les systèmes d'aide à la conduite se sont multipliés au fil des années pour améliorer la sécurité et le confort d'utilisation des véhicules. Dans un premier temps, ces systèmes d'aide à la conduite utilisaient essentiellement des capteurs proprioceptifs et ne permettaient pas d'analyser l'environnement dans lequel le véhicule évoluait. Au début des années 2000, les systèmes d'aide à la conduite se sont complexifiés pour devenir des systèmes d'aide à la conduite avancés (ADAS) offrants de de nouvelles fonctions se basant sur la perception de l'environnement. Pour répondre à ce besoin, les véhicules ont été équipés de capteurs extéroceptifs et de système de traitement pour comprendre l'environnement proche dans lequel évolue le véhicule. Naturellement, la multiplication des ADAS a porté l'idée de véhicules autonomes et souligné le besoin de comprendre l'environnement le plus précisément possible. Au début des années 2010, une approche coopérative a émergé afin d'outrepasser les limitations des capteurs ainsi que les problématiques d'occultations. Cette approche fut initialement abordée par la coopération entre véhicules jusqu'à l'arrivée d'infrastructures. Toutefois, les méthodes de coopération sont encore disparates et soulèvent de nouvelles problématiques techniques. Ainsi, les travaux effectués pendant cette thèse s'orientent sur deux axes organisés en trois chapitres. Le premier constitue la réalisation d'un état de l'art sur la perception coopérative dans le contexte automobile et occupera le premier chapitre. Le second axe consiste en deux contributions occupant chacune un chapitre. L'une se concentrant sur la génération d'une carte des obstacles au niveau d'une intersection via une architecture faisant coopérer les véhicules et une infrastructure, l'autre sur la mise à jour vers une carte sémantique.

R.2 LA PERCEPTION COOPÉRATIVE DANS UN CONTEXTE AUTOMOBILE

R.2.1 *Les bases de la coopération*

Pour effectuer de la perception coopérative, on observe que trois éléments sont obligatoires et doivent être choisis minutieusement, car ils impacteront les résultats suivants. Ces trois éléments sont : les capteurs, la communication et l'architecture. Toutefois, malgré le soin apporté au choix de ces trois éléments, l'aspect coopératif impose de nouveaux défis qu'il est important de prendre en compte lors de la réalisation d'un système coopératif.

R.2.1.1 *Capteurs*

Les capteurs sont à la racine des systèmes de perception, qu'ils soient coopératifs ou non. Il est important de connaître leurs performances seul ou au sein d'un système de perception multimodal non coopératif pour avoir un point de comparaison sur les systèmes de perception coopératifs. Nous allons donc dans les prochaines lignes étudier les capteurs les plus communs dans un contexte automobile. Le tableau 1.1 compare les capteurs et leurs performances.

SYSTÈME DE NAVIGATION GLOBALE PAR SATELLITE Le positionnement par satellite, souvent appelé Global Positioning System (GPS) ou Global Navigation Satellite System (GNSS), est sans doute l'un des capteurs les plus répandus, puisqu'utilisés par les conducteurs eux-mêmes en guise d'assistance à la conduite. Alors que le GPS seul obtient une erreur allant jusqu'à 20 m [104], des évolutions comme le Real-Time Kinematic GPS (GPS RTK) permettent une précision de quelques centimètres [65]. Pour augmenter la fréquence d'estimation de la position, l' GNSS est souvent associé à un Inertial Measurement Unit (IMU) [74, 125].

CAMÉRA Les caméras sont des capteurs fréquents sur les véhicules et permettent plusieurs taches comme la détection d'objets [6] ou l'estimation de trajectoire et la création de cartes via le Simultaneous Mocalization And Mapping (SLAM) [94]. Les caméras sont souvent associées à des capteurs de distances, car elles ne fournissent pas directement d'informations de profondeur [65].

RADAR Les Radio Detection and Ranging (RADAR)s font partie des capteurs de distances fréquemment utilisés [48, 111, 114]. Leurs prix sont assez faibles et permettent d'obtenir la distance d'un objet, mais souffre d'une mauvaise résolution angulaire.

LIDAR Les Light Detection and Ranging (LiDAR) sont, comme les RADAR, des capteurs de distance qui sont de plus en plus présents sur les véhicules malgré leur prix élevé [23, 77]. Ils sont fréquemment associés à des caméras pour apporter l'aspect de profondeur aux images. Ainsi, on les retrouve dans les mêmes tâches que pour les caméras, c'est-à-dire de la détection et classification d'objets, estimation de trajectoire et cartographie comme le SLAM [21, 24, 68, 69].

CAPTEUR ULTRASONIC Ces capteurs sont très commun grâce à leur faible coût, mais ne fonctionne qu'à basse vitesse [65].

MÉTHODES BASÉES RF Ces méthodes se basent sur les communications entre le véhicule et une infrastructure ou d'autres véhicules [30]. On distingue quatre méthodes pour se positionner à partir de plusieurs points définis. Elles se basent sur le Received Signal Strength Indication (RSSI), le Time Of Arrival (TOA), le Time Difference Of Arrival (TDOA) et le Angle Of Arrival (AOA). On distingue une

autre approche qui permet de reconnaitre l'environnement radio à partir d'une carte réalisée en amont et de l'empreinte radio de chaque localisation.

INSTALLATION TYPIQUE Aujourd'hui, on trouve au sein d'un système des couples de capteurs. On retrouve notamment fréquemment le couple GNSS-IMU pour la localisation et les couples caméra-LiDAR et caméra-RADAR pour offrir les informations de profondeurs robustes manquantes sur les images [48, 119].

R.2.1.2 *Communication*

Dans un système coopératif avec plusieurs agents, les moyens de communication sont incontournables. Ceux-ci doivent offrir une infrastructure robuste ainsi que divers standards permettant aux véhicules de communiquer entre eux.

INFRASTRUCTURE DE COMMUNICATION pour les communications à courtes distances, le standard IEEE 802.11p, extension du Wireless Fidelity (Wi-Fi) est fréquemment utilisé sous la norme américaine Wireless Access in Vehicular Environments (WAVE) ou la norme européenne ITS-G5, mais offre des performances réduites [12]. Une autre solution consiste en l'utilisation du réseau cellulaire, et notamment du réseau 5G [56] et de ses liens haut débit Millimeter Wave (mmWAVE) [73, 83]. Cependant, la mise en place de ce réseau est toujours en cours et des tests de robustesse doivent être effectués. Enfin, d'autres solutions sont envisagées comme celles se basant sur l'Ultra Wide Band (UWB) [52, 62] et le Visible Light Communication (VLC) [65] mais sont encore très expérimentales.

TRANSPORT DES DONNÉES Pour transporter des données, il est nécessaire de les empaqueter. Le protocole Vehicular Ad-hoc Network (VANET) [38] est assez répandu et se constitue d'une norme européenne Cooperative-ITS (C-ITS) [40] et d'une norme américaine Dedicated Short-Range Communication (DSRC) [60]. Elles apportent des solutions sur les couches du modèle OSI, énoncé dans le tableau 1.2. En dehors du modèle VANET, le *framework* proposé avec Robot Operating System (ROS) [88] permet une communication presque transparente sur un réseau, peu importe sa taille [4, 63, 64, 73, 110].

R.2.1.3 *Architecture*

L'architecture d'un système coopératif indique comment les agents vont communiquer entre eux. On remarque deux approches : l'approche centralisée et l'approche distribuée.

APPROCHE CENTRALISÉE L'approche centralisée se définie par un point par lequel toutes les données transitent. Cette approche est souvent utilisée dès lors qu'une infrastructure en bord de voie est présente et traite des données [48, 77]. Cette approche permet

d'agréger plus de données et d'avoir un point de vue plus global sur la scène tout en pouvant offrir plus de puissance de calcul. Cependant, cette approche induit un délai entre les données issues des capteurs et les données après traitement. De plus, si l'élément central tombe en panne, l'intégralité du système cesse de fonctionner.

APPROCHE DISTRIBUÉE Cette approche est quant à elle plus commune dans les scénarii où des véhicules communiquent les uns avec les autres [73, 119, 126]. Cette approche a pour avantage d'être toujours disponible dès qu'au moins deux véhicules sont disponibles et est résiliente. Toutefois, le réseau est moins optimisé et des délais peuvent impacter la synchronisation des agents.

R.2.1.4 *Défis*

Dès lors que nous utilisons plusieurs capteurs, certains défis apparaissent comme la multimodalité ou le calibrage des capteurs. Ces défis sont exacerbés dans un système coopératif et d'autres s'adjoignent à eux.

MULTIMODALITÉ La multimodalité est l'un des défis les plus connus, car l'un des défis les plus fréquemment rencontrés dès lors que plusieurs capteurs de différents types sont associés. Deux solutions sont apportées. La première consiste à créer des objets topologiques pour chaque objet détecté dans la scène et d'enrichir ou d'affiner successivement ses caractéristiques en utilisant chaque type de capteurs séparément [56]. Une autre solution est de traiter chaque objet indépendamment et dans une représentation finale commune afin de les fusionner [77, 119].

CALIBRAGE Le calibrage consiste à trouver la pose des capteurs les uns par rapport aux autres (les paramètres extrinsèques). Dans un système où les capteurs sont fixes les uns par rapport aux autres, ce calibrage peut être effectué manuellement [4, 77]. Or, ce n'est pas le cas lorsque des agents sont dynamiques et aucune méthode ne donne de réelle solution pour effectuer une calibration dynamique.

SYNCHRONISATION La synchronisation est une étape primordiale, car un délai induit impacte significativement la précision de l'estimation de pose des objets dynamiques. Bien qu'il soit possible de déclencher physiquement toutes les acquisitions simultanément sur un système unique, cette tâche est impossible dès lors que plusieurs agents sont dynamiques. Une solution est d'utiliser le protocole Network Time Protocol (NTP) [120] ou l'horloge transmise par les satellites GPS [112].

POINTS DE VUES Dans un système coopératif, les points de vues peuvent être extrêmement différents. Ainsi, un même objet observé par différents points de vues peut avoir un aspect radicalement différent. Actuellement, ce qui semble se rapprocher le plus de cette problé-

matique sont les systèmes de Motion Capture (Mo-Cap) moyennant l'installation d'amers singulière sur les objets facilitant l'association. Aujourd'hui, ce problème n'est pas encore réellement exploré.

ASSOCIATION Enfin, en rapport avec la différence de points de vue, il est nécessaire pour toutes les observations d'un objet de les associer ensemble. Une solution consiste à utiliser sa position ou encore des paramètres comme sa vitesse [63]. Une autre solution proposée consiste en l'utilisation de graphes bipartites [81].

R.2.2 *Localisation*

La localisation coopérative est un sujet très actif dans la littérature. Celui-ci consiste à localiser le véhicule ainsi que les autres agents connectés. On remarque deux approches : l'estimation de pose sans à priori et l'optimisation de pose. Le tableau 1.4 compare les performances des méthodes de localisation.

R.2.2.1 *Estimation de pose coopérative*

L'estimation de pose coopérative, sans à priori, permet de trouver sa position à partir d'autres points dont la position est connue. Ce procédé trouve toute son utilité dans des lieux non couverts par les infrastructures de positionnement comme le GNSS. D'ailleurs, nous pouvons noter que le système de positionnement par satellite est lui-même un système de localisation coopératif satellite-utilisateur.

MULTILATÉRATION Le principe de la multilatération est d'utiliser la distance entre des ancres (mobiles ou non) localisées. C'est sur ce principe que fonctionne le GNSS où les satellites sont des ancres mobiles avec une position parfaitement connue ainsi que les méthodes basées sur le RSSI, le TOA et le TDOA fonctionnent [2, 3, 86, 91].

TRIANGULATION Le principe de la triangulation est similaire à celui de la multilatération, mais se base sur des mesures d'angles au lieu des mesures de distances. C'est sur ce principe que se base l'AOA [54].

APPROCHE GÉOMÉTRIQUE L'approche géométrique consiste à effectuer une estimation de pose relative à une ancre localisée grâce à des caméras, LiDAR ou RADAR [4, 35]. Cette estimation peut être effectuée par une ancre ou par le véhicule. Toutefois, il a été observé que les performances ne sont pas équivalentes en fonction de qui fait l'estimation relative [55].

R.2.2.2 *Optimisation de l'estimation de pose coopérative*

L'optimisation de l'estimation de pose consiste à affiner la pose des différents utilisateurs. Cette partie peut utiliser les trois procédés de l'estimation de pose et va chercher à minimiser les erreurs [43]. On

trouve alors trois méthodes principales. Celles basées sur les Extended Kalman Filter (EKF) [81], celles basées sur les Particle Filter (PF) [58] et celles basées sur la théorie des graphes citegulati2016vehicle, gulati2017graph.

R.2.3 *Détection et suivi d'objets*

Une autre tâche de la perception coopérative est la détection et le suivi d'objets. Ce sujet est moins exploré que celui de la localisation coopérative, mais y est aussi très lié dans la partie de suivie. Le tableau 1.5 récapitule les méthodes et leurs performances.

R.2.3.1 *Détection et classification*

La détection et la classification coopérative est peut-être la partie la moins explorée par la communauté. En général, cette tâche est effectuée localement sur les agents et sur des objets topologiques qui sont distribués en vue d'être fusionnés [4, 77]. C'est ce qu'on appelle la fusion tardive, qui est l'un des trois schémas de fusion pour la détection et la classification [6, 27].

FUSION TARDIVE Comme énoncé dans les lignes précédentes, cette forme de fusion utilise la capacité individuelle de chaque agent observateur pour obtenir des objets déjà traités pour les fusionner.

FUSION PRÉCOCE Cette méthode de fusion compte sur le partage de données brutes et uniformes pour densifier les observations et aider à la détection et à la classification.

FUSION PROFONDE Cette méthode se base sur le fait que les algorithmes de classification et de détection sont aujourd'hui majoritairement constitués de réseaux de neurones [49]. L'idée est de faire passer les données brutes dans les premières couches d'un réseau de neurones et de partager des valeurs intermédiaires. Toutefois, cela nécessite que tous les agents aient le même réseau de neurones ayant reçu le même entrainement [23].

R.2.3.2 *Suivi*

Le suivi et la localisation coopérative sont intrinsèquement liés [29]. Ainsi, dans beaucoup de travaux, ces deux notions sont confondues comme avec les méthodes de Simultaneous Localization And Tracking (SLAT) [113] ou de Gaussian Mixture Probability Hypothesis Density (GMPHD) [4, 25, 56]. Pour suivre des objets topologiques au travers d'une scène, il est aussi possible de prendre des paramètres constituant l'objet comme sa vitesse ou sa pose [63, 64, 77, 81].

R.2.4 *Génération de cartes*

Globalement, les cartes se distinguent en deux familles : les cartes géométriques et les cartes volumétriques. La génération ou la mise à jour de cartes coopératives s'applique à ces deux familles.

R.2.4.1 *Cartes géométriques*

Les cartes géométriques contiennent des éléments décrit par des paramètres [4, 8, 11, 77, 119]. En tant qu'utilisateurs, nous sommes déjà familiers des cartes géométriques coopératives. En effet, de nombreuses applications de navigation participative mettent à profit les observations des utilisateurs pour mettre à jour certains éléments de la carte. C'est notamment sur ce principe que se basent les cartes utilisant la formalisation Local Dynamic Maps (LDM) proposée par l'European Telecommunications Standards Institute (ETSI) [39]. Cette formalisation prend la forme d'une carte formée de 4 couches. La première couche constitue les informations statiques comme les voies de circulation et leur vitesse associée. La seconde est constituée des informations à long terme comme des zones de travaux. La troisième couche est constituée d'informations à moyen terme comme des véhicules en stationnement, des bouchons ou des conditions météorologiques particulières. Enfin, la quatrième et dernière couche concerne les éléments à court terme comme les véhicules en circulation ou le statut des feux de circulation.

R.2.4.2 *Cartes volumétriques*

Contrairement aux cartes géométriques, les cartes volumétriques représentent l'environnement dans une forme discrétisée [96]. Les grilles d'occupation et ses dérivées sont un exemple très commun de carte volumétrique dans lequel l'environnement est discrétisé en cellules et ont l'avantage de facilement pouvoir être fusionnées [14, 63, 64]. Chaque cellule donne une indication sur son statut occupé ou libre. Bien que certains auteurs génèrent ces cartes à partir de caméras, elles sont majoritairement construites à partir de capteurs de distances. Cependant, les scanners laser offrent naturellement des cartes sous la forme de nuages de points pouvant être fusionnés notamment grâce à l'algorithme d'Iterative Closest Point (ICP) et de ses variantes [116, 122].

R.2.5 *La perception coopérative aujourd'hui*

Ces dernières années, la perception coopérative gagne de plus en plus en intérêt et de nombreux projets tentent de répondre à un ensemble de scénarii.

R.2.5.1 *Scénarii et expérimentations*

Généralement, la perception coopérative répond à la problématique du manque de visibilité dû aux limitations des capteurs ou aux occul-

tations. La gestion des intersections est un des scénarii qui bénéficie d'une grande part des travaux. Viennent ensuite les dépassements, rabattement et les insertions dans lesquels les autres véhicules forment de multiples occultations [4, 16, 42, 64, 77, 119].

R.2.5.2 *Jeux de données*

Bien que les projets de perception coopérative se multiplient et que l'effort de recherche s'intensifie dans cet axe, on observe que les jeux de données sont très peu nombreux. On note la présence du dataset T&J [23] où plusieurs véhicules équipés de LiDAR s'observent. Pour obtenir des jeux de données coopératifs, il faut utiliser un simulateur comme celui proposé par CARLA [33, 89, 92, 98].

R.3 GÉNÉRATION COOPÉRATIVE DE GRILLES D'OCCUPATION ÉVIDENTIELLES

Après avoir effectué notre état de l'art, nous avons remarqué que les efforts de recherches se concentrent sur les architectures Vehicle-to-Vehicle (V2V) Vehicle-to-Infrastructure (V2I) unidirectionnelles. En effet, lorsqu'une infrastructure est présente dans un système, seules les données issues de cette dernière et de l'ego-véhicule sont utilisées. Nous souhaitons donc utiliser ces points de vue jusqu'ici ignorés pour générer une carte. Cette carte sera ensuite redistribuée à tous les utilisateurs. Afin de rester dans les spécifications des liens de communication, nous n'utiliserons que des boîtes englobantes 2-Dimensional (2D) issues des images acquises par les différents points de vue.

R.3.1 *Idée générale*

L'idée générale est que nous pouvons faire une projection inverse des boîtes englobantes dans l'espace 3-Dimensional (3D). Puisque nous n'avons aucune information sur la distance, ces projections inverses prennent la forme de frustum. Dès lors qu'un objet est observé par plusieurs points de vue, nous pouvons estimer que les frustums se croiseront et que la position de l'objet dans l'espace se situe à cette intersection.

Puisque les véhicules évoluent sur le sol, nous pouvons estimer la position de celui-ci en cherchant l'intersection avec le frustum associé à sa détection et le plan du sol. Cette intersection entre le frustum et le sol forme une silhouette similaire à une ombre projetée. Ainsi, plus un véhicule est observé, plus les silhouettes projetées au sol s'accumuleront et plus l'estimation de l'emprise réelle du véhicule sera précise.

R.3.2 *Architecture du système*

D'un point de vue macroscopique, notre architecture est constituée de trois éléments. Les agents observateurs, le Road Side Unit (RSU) et

les utilisateurs qui reçoivent la carte générée sans nécessairement être contributeurs. Les agents envoient des données au RSU qui génère une carte et l'envoie aux utilisateurs selon la figure 2.3.

AGENTS Les agents observateurs peuvent aussi bien être des véhicules ou un point de vue de l'infrastructure. En fait, nous les considérons simplement comme des points de vue transmettant une boîte englobante 2D ainsi que les paramètres intrinsèques et la pose du capteur d'images.

RSU Le RSU reçoit les données transmises par les agents pour en créer une carte sous la forme d'une grille d'occupation. Les données suivent le cheminement décrit dans la figure 2.4. Dans un premier temps, on duplique ces données en de multiples particules sur lesquelles on ajoute du bruit sur tous les paramètres afin de modéliser les incertitudes. Ces particules sont projetées et discrétisées avant d'être fusionnées dans une grille d'occupation locale. Nous obtenons donc une grille d'occupation locale pour chaque point de vue sur laquelle les silhouettes représentant l'emprise au sol des véhicules ainsi que l'incertitude qui leur est associée. Ces grilles sont ensuite synchronisées puis fusionnées avant d'être partagées.

R.3.3 *Méthodes*

Les deux points les plus importants dans le traitement des données effectuées par le RSU réside dans la création des grilles d'occupation locales ainsi que la fusion de ces dernières dans une grille d'occupation finale à partager.

R.3.3.1 *Génération de grilles d'occupation locales*

Pour obtenir les silhouettes des boîtes englobantes sur le sol, nous devons trouver l'intersection entre le frustum associé aux boîtes englobantes avec le plan du sol avec pour à priori la pose approximative des caméras.

FRUSTUM Pour retrouver les paramètres décrivant le frustum, nous avons besoin de quatre rayons. Puisque les boîtes englobantes sont issues du plan image de la caméra, nous considérerons le modèle du sténopé inverse. Le modèle du sténopé permet de prendre un point 3D et de le projeter sur le plan image en un point 2D. Néanmoins, avec cette opération nous perdons l'information de profondeur. Par conséquent, en effectuant l'opération inverse, pour un point 2D du plan image, nous obtenons une multitude de points 3D suivant une ligne passant par le centre optique de la caméra et le point 3D réel duquel provient le point 2D sur l'image. Suivant ce principe, avec les quatre coins formant une boîte englobante, on obtient quatre lignes 3D formant le frustum. Ces quatre lignes sont décrites en utilisant deux points 3D appartenant à la projection inverse d'un point 2D avec

deux distances arbitraires différentes. On considérera par exemple le centre optique de la caméra et un point reprojeté à 1 m de la caméra.

OBTENTION DE SILHOUETTES Puisque les lignes du frustum sont décrites par deux points, nous utilisons le système de coordonnées de Plücker pour décrire les lignes. Le plan du sol est quant à lui décrit par 3 points ou le vecteur normal du plan et une distance par rapport à son origine. Ainsi, on retrouve le point 3D $\boldsymbol{P_{intersection}}$ à l'intersection d'une ligne $\boldsymbol{L}$ et d'un plan $\boldsymbol{\pi}$ via une multiplication décrite dans l'équation R.1.

$$\boldsymbol{P_{intersection} = L\pi} \tag{R.1}$$

Pour un ensemble de quatre lignes formant le frustum, nous obtenons donc quatre points sur le plan du sol formant un polygone correspondant à la silhouette de la boîte englobante 2D. Ce polygone est discrétisé afin d'assigner aux cellules une valeur d'occupation dans le but d'obtenir une grille d'occupation.

R.3.3.2 *Fusion des grilles*

Pour fusionner les grilles d'occupation, nous avons testé deux approches. La première se base sur la théorie bayésienne tandis que l'autre se base sur la théorie de l'évidence aussi appelée la théorie de Dempster-Shafer ou Dempster-Shafer Theory (DST).

FUSION BAYÉSIENNE Puisque les observations entre deux points de vue sont indépendantes et ne s'influencent pas les unes les autres, nous pouvons utiliser le principe de probabilité jointe décrite dans l'équation R.2.

$$P(o_1 \cap o_2) = P(o_1) \times P(o_2) \tag{R.2}$$

Où o_1 et o_2 représentent des observations distinctes. Ainsi, pour deux cellules de même coordonnées, mais issues d'observations différentes, la probabilité jointe de leur occupation est le produit de la probabilité d'occupation selon les deux observations. Puisque cette opération est associative, nous pouvons fusionner les cellules de même coordonnées pour un nombre arbitraire d'observations sans se préoccuper de l'ordre. On renouvellera cette opération pour toutes les coordonnées de la carte.

FUSION ÉVIDENTIELLE La DST se base sur un système de masses associé à des éléments focaux. Puisque nous avons une grille d'occupation, nous considérons deux états pour une cellule : occupée $\mathcal{O}$ ou libre $\mathcal{F}$. Cet univers de possibilité est noté Ω. Les éléments focaux proviennent de l'ensemble des sous parties de Ω et est noté 2^Ω où $2^\Omega = \{\varnothing, \mathcal{O}, \mathcal{F}, \Omega\}$. Une masse est associée à chacun des éléments focaux suivant une fonction nommée Basic Belief Assigment (BBA). Nous avons défini cette fonction suivant l'algorithme 1. Nous obtenons donc une grille évidentielle où, pour chaque coordonnée, une cellule contient les masses des quatre éléments focaux.

FIGURE R.1 : Champs de vue de tous les agents du dataset.

Pour fusionner ces grilles évidentielles, nous avons utilisé la règle de combinaison de Dempster définie dans l'équation R.3 où m_1 et m_2 correspondent aux masses pour deux observations.

$$m_f(X) = m_1(X) \oplus m_2(X), \ X \in \Omega \tag{R.3}$$

$$m_f(X) = \frac{1}{1-K} \sum_{Y \cap Z = X \neq \varnothing} m_1(Y) m_2(Z) \tag{R.4}$$

$$K = \sum_{Y \cap Z = \varnothing} m_1(Y) m_2(Z), \ \forall Y, Z \in 2^{\Omega} \tag{R.5}$$

Cette opération est renouvelée pour toutes les coordonnées de la carte. La carte fusionnée ainsi obtenue est donc une grille évidentielle qu'il faudra retransformer en grille d'occupation. Nous avons donc utilisé les valeurs de $m(\mathcal{O})$ en guise de grille d'occupation.

R.3.4 *Résultats*

Puisqu'aucun jeu de données ne proposait un ensemble de véhicules tous instrumentés ainsi qu'une infrastructure avec des champs de vue se recouvrant, nous avons dû créer notre propre jeu de données pour tester notre approche. Nous avons ensuite fait une étude qualitative pour vérifier que nous avions des résultats cohérents puis nous avons fait une étude quantitative.

R.3.4.1 *Jeu de données*

Pour construire notre jeu de données, nous avons utilisé le simulateur CARLA. En guise de scénario, nous faisons transiter trois véhicules dans un rond-point surveillé par un point de vue infrastructure. Ces véhicules embarquent tous une caméra et s'observent les uns les autres. Tous les agents partagent leur point de vue simultanément. La figure R.1 montre les différents points de vue disponibles dans ce dataset.

R.3.4.2 *Résultats qualitatifs*

Les résultats qualitatifs nous ont permis d'isoler 6 scénarii ayant des configurations de véhicules différentes. Nous avons notamment remarqué que plus les champs de vue se recoupent, meilleurs sont les résultats. Nous avons pu aussi observer que les véhicules semblent bien positionnés sur la carte et que les résultats sont cohérents avec la vérité terrain comme le montre la figure R.2.

R.3.4.3 *Résultats quantitatifs*

Afin d'étudier quantitativement les résultats de notre approche, nous avoins choisi d'utiliser deux indicateurs : l'intersection sur l'union (Intersection over Union (IoU)) ainsi que le F1-score. Nous considérons une cellule comme occupée si sa probabilité est supérieure à 0.5. Pour la méthode basée sur la fusion évidentielle, nous obtenons un IoU maximal de 30.52 % sur l'une de nos séquences et un IoU global de 22.35 %. En revanche, pour la méthode de fusion bayésienne, les résultats sont nuls, car aucune cellule ne semble dépasser le seuil requis pour être considérée comme occupée.

R.4 GÉNÉRATION COOPÉRATIVE DE GRILLES SÉMANTIQUES ÉVIDENTIELLES

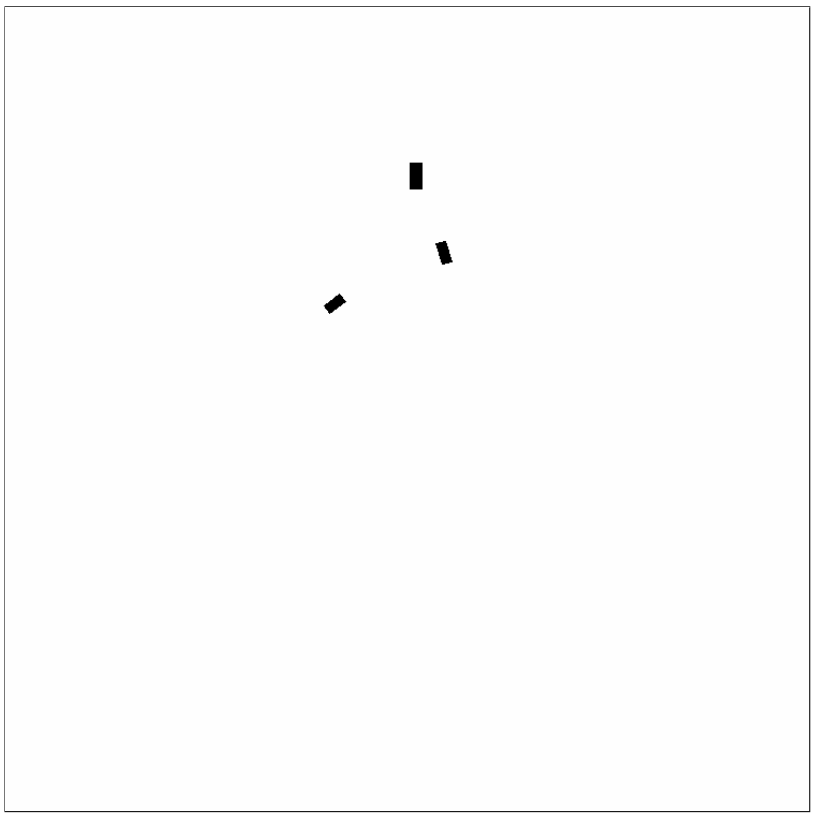
Jusqu'ici, nous n'avons pas abordé l'aspect sémantique sur nos cartes. En fait, nous avons validé que notre approche apporte des résultats intéressants malgré une implémentation rudimentaire. Nous allons donc ajouter l'aspect sémantique dans nos travaux et incrémentalement améliorer notre implémentation. Nous testerons aussi notre approche sur une plus grande variété de datasets.

R.4.1 *Nouvelle architecture*

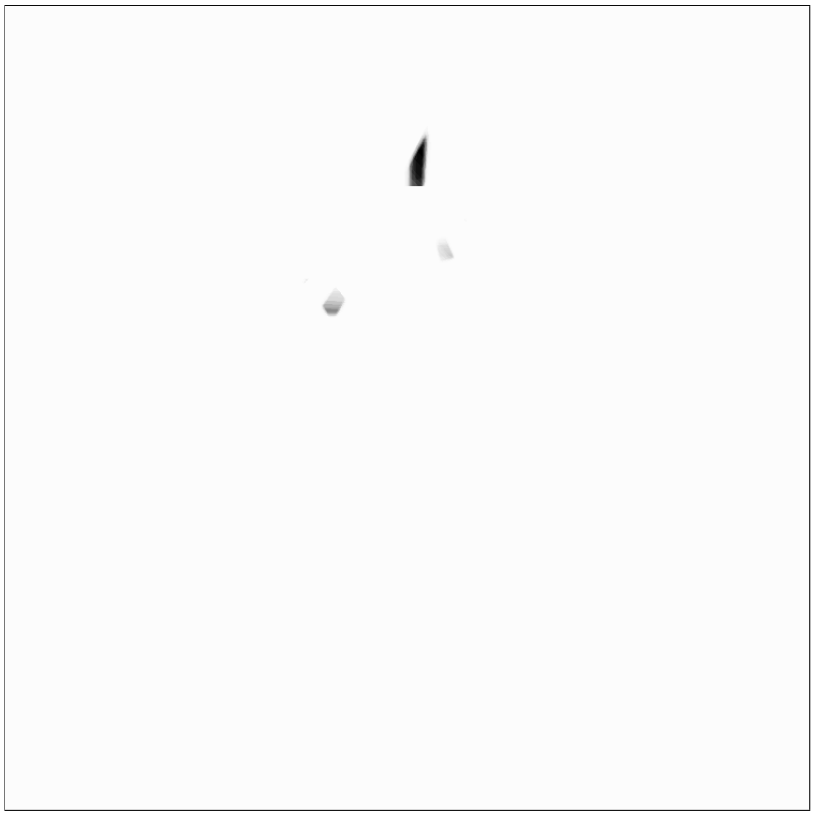
L'architecture globale de notre approche n'évolue que très peu. Les agents observateurs partagent les boîtes englobantes avec le label associé à l'objet observé avec les paramètres intrinsèques et la pose du capteur d'images. Une carte sous forme de grille sémantique locale est créée à partir des boîtes englobantes puis est transformée en grille d'occupation sémantique ou en grille somatique évidentielle avant d'être fusionnée soit par une méthode bayésienne, soit par une méthode basée sur la DST. Enfin, la carte issue de la fusion de tous les points de vue est transformée en carte sémantique par un bloc de prises de décision.

R.4.2 *Grilles locales*

La création de la grille sémantique locale utilise le même principe d'obtention des silhouettes expliquées précédemment se basant sur l'intersection de frustums créés à partir de boîtes englobantes et du plan du sol. Toutefois, lors de la discrétisation, les cellules ne se voient



(a) Vérité terrain



(b) Carte après fusion par DST

FIGURE R.2 : Cartes à la trame 150 permetant une étude qualitative. Les véhicules sont détéctés mais plus grand que leur taille réelle.

pas assigner une valeur de probabilité, mais le label associé à la boîte englobante d'origine pour former une grille sémantique.

Puisque nous connaissons maintenant les labels associés aux silhouettes, nous pouvons réduire leurs tailles dans des dimensions raisonnables. En effet, une voiture à rarement une emprise au sol supérieur à 6 m et un piéton une emprise supérieure à 1 m sur l'axe le plus long de la silhouette. Nous réduisons donc la taille des silhouettes et assignons le label *non observé* dans la partie que nous avons supprimé de la partie silhouette originale.

R.4.2.1 *Assignement des valeurs*

Selon si nous effectuons une fusion dans le cadre bayésien ou dans le cadre evidentiel, l'assignation de valeurs diffère.

GRILLE D'OCCUPATION SÉMANTIQUE La grille d'occupation sémantique contient plusieurs couches : une par label. Dans notre cas, nous avons choisi de considérer trois labels : des piétons, des véhicules et le terrain. Ainsi, pour chaque cellule d'une carte sémantique, en fonction du label de la cellule, nous appliquons sur la cellule cellule associée à une position correspondante des valeurs prédéfinies pour les sous-cellules représentant un espace de label.

GRILLE SÉMANTIQUE ÉVIDENTIELLE Pour les grilles sémantiques évidentielles, le principe est similaire aux grilles d'occupation sémantique, mais le nombre de sous-cellules correspond aux éléments focaux formés par l'ensemble des parties de l'ensemble des labels. Ici aussi, pour chacune des observations un ensemble de valeurs à assigner à chacun des éléments focaux est prédéfini.

R.4.3 *Méthode de fusion*

Dans notre première implémentation de notre approche, nous avons observé que l'approche bayésienne ne donnait pas de résultats, car le seuil de probabilité de détection d'occupation des cellules n'était jamais dépassé. Cependant, puisque notre méthode de prise de décision change, nous pourrions avoir des résultats différents. C'est pourquoi nous implémentons aussi bien une version adaptée pour l'aspect sémantique d'une méthode de fusion se basant sur la théorie bayésienne qu'une basée sur la DST.

R.4.3.1 *Fusion bayésienne*

Nous considérons toujours que les observations entre tous les utilisateurs sont indépendantes et qu'elles ne sont pas influençables par les unes avec les autres. Par conséquent, nous utiliserons la propriété des probabilités jointes pour fusionner les sous-cellules de même label et de mêmes coordonnées de toutes les observations.

R.4.3.2 *Fusion évidentielle*

Jusqu'ici, nous avons utilisé uniquement la règle de combinaison de Dempster qui est normalisée par une mesure du conflit d'observation. Nous testerons ici une autre méthode, en plus de la règle de combinaison de Dempster : la règle de combinaison conjonctive. Cette règle de combinaison est très proche de celle de Dempster puisque seule la partie de normalisation est abandonnée comme décrite dans l'équation R.6.

$$m_1(X) \textcircled{\cap} m_2(X) = \sum_{Y \cap Z = X \in 2^\Omega} m_1(Y) m_2(Z) \tag{R.6}$$

Similairement à ce que nous avons expliqué dans la section précédente, les cellules de mêmes coordonnées et pour les trois labels *véhicule*, *piéton* et *terrain* sont fusionnées pour toutes les observations.

R.4.4 *Prises de décision*

L'aspect décisionnel a été fortement n'a pas été totalement exploré lors de notre première implémentation. C'est pourquoi nous avons développé de nouvelles solutions pour décider quel label assigner à une cellule à partir des cartes fusionnées.

R.4.4.1 *À partir d'une grille d'occupation*

Puisque la grille d'occupation sémantique finale contient les probabilités pour chacun des labels, nous avons décidé de choisir le label ayant la probabilité maximale. Puisque nous retrouvons une grille sémantique, nous ne devrions plus avoir à considérer un seuil d'occupation, mais une correspondance de label. Cela supprime donc la problématique rencontrée précédemment.

R.4.4.2 *À partir d'une grille évidentielle*

Pour prendre une décision à partir des éléments focaux d'une grille évidentielle, beaucoup de solutions existent que nous avons regroupées en trois familles.

MASSES Une méthode simple, s'inspirant de celle adoptée pour les grilles d'occupation sémantique, est de sélectionner le label ayant une masse maximale en ignorant les autres éléments focaux.

CROYANCE ET PLAUSIBILITÉ La DST vient avec deux notions : la *croyance* dont la fonction est notée *bel* et la *plausibilité* dont la fonction est notée *pl*. Les deux valeurs obtenues par ces fonctions encadrent la probabilité associée au label. Il est donc possible de choisir le label ayant une *croyance* ou une *plausibilité* maximale.

ESTIMATION DE PROBABILITÉS Une solution pour estimer la probabilité est de prendre la valeur se situant entre les valeurs de *croyance*

et de *plausibilité*. On choisira donc le label ayant la probabilité estimée maximale. De la même façon, il est possible d'ajouter un poids qui diminuera la probabilité estimée si l'écart entre la *croyance* et la *plausibilité* est grand.

Il est aussi possible de mesurer la probabilité estimée par le calcul de probabilité pignistique $BetP$ décrit dans l'équation R.7.

$$BetP(A) = \sum_{\varnothing \neq B \subseteq \Omega} \frac{m(B)}{1 - m(\varnothing)} \frac{|A \cap B|}{|B|}, \ \forall A \subseteq \Omega \tag{R.7}$$

La fonction $BetP$ est normalisée par le conflit. Par conséquent, elle devrait avoir les mêmes résultats après une fusion par la règle de combinaison conjonctive et la règle de combinaison de Dempster. On choisira ici aussi le label pour lequel la valeur de $BetP$ est maximale.

R.4.5 *Résultats*

Jusqu'ici, nous avons testé notre approche avec deux méthodes de fusion et un jeu de données limité. Par conséquent, nous allons tester notre approche avec plusieurs jeux de données de notre conception ainsi qu'avec différents paramètres.

R.4.5.1 *Jeux de données*

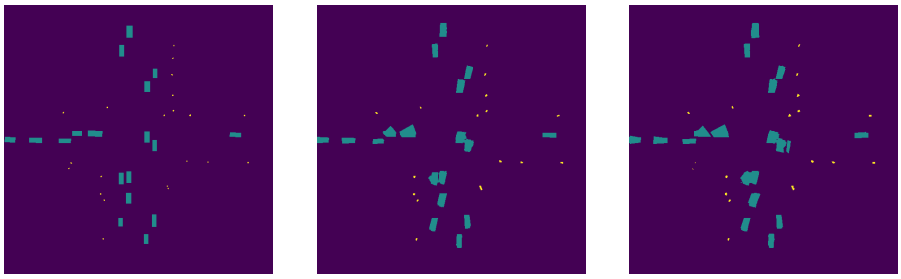
Puisqu'au début de nos travaux sur la mise à jour de notre approche, nous nous sommes encore heurtés à la problématique du manque de jeu de données coopératif, nous avons créé de nouveaux jeux de données avec plus de véhicules et des configurations différentes. De plus, nous voulons tester notre approche avec des piétons, qui étaient absents sur notre jeu de données original. Nous avons donc fait 3 jeux de données au niveau du rond-point en faisant varier le nombre de véhicules et les configurations de l'infrastructure ainsi qu'un autre jeu de données au niveau d'une infrastructure.

R.4.5.2 *Résultats qualitatifs*

Comme dans notre première implémentation, nous pouvons remarquer que notre approche fonctionne aussi avec l'aspect sémantique comme le montre la figure R.3. On remarque qu'ici, la fusion bayésienne fonctionne quasiment aussi bien que l'approche basée sur la DST. L'étude quantitative nous permet de mieux les distinguer. Pour comparer nos résultats avec les méthodes de l'état de l'art, nous avons utilisé les algorithmes de reconstruction 3D proposés avec COLMAP [95]. Cependant, en donnant les images de tous les points de vue ainsi que la pose des capteurs à COLMAP, nous ne sommes pas parvenus à obtenir de résultats. Cela est dû à la différence de point de vue et à l'apparence des objets qui rend impossible l'association des features.

R.4.5.3 *Résultats quantitatifs*

Pour effectuer notre étude quantitative, nous avons utilisé trois métriques : l'IoU, le F1-Score ainsi que le Correct Ratio (CR). Nous avons



(a) Vérité terrain.

(b) Carte après fusion par la règle de Dempster et après une prise de décision avec la masse maximale.

(c) Carte après fusion bayésienne.

FIGURE R.3 : Cartes sémantiques pour l'étude qualitative. En violet : le terrain, en cyan : les véhicules et en jaune : les piétons.

Fusion	Classe	IoU	F1-Score	CR
Méthode bayésienne	$\mathcal{V}$	26.21	36.46	96.50
	$\mathcal{P}$	22.33	41.52	99.95
	$\mathcal{T}$	96.40	98.17	96.45
	Moyenne	48.31	58.71	N/A
Méthode (DST)	$\mathcal{V}$	50.55	67.07	98.87
	$\mathcal{P}$	28.03	43.49	99.98
	$\mathcal{T}$	98.84	99.41	98.85
	Moyenne	59.14	69.99	N/A

TABLE R.1 : Detail de l'IoU, F1-Score et du CR (en %) sur notre jeu de données autour d'un rond-point et une densité de trafic forte.

comparé l'approche avec la fusion bayésienne et la fusion par la DST. Puis, nous avons comparé les méthodes de prise de décisions, le taux d'agents contributeurs dans la flotte de véhicules et la densité du trafic.

FUSION BAYÉSIENNE VS FUSION PAR LA DST Dans un premier temps, nous voulons comparer pour un même jeu de données l'approche avec la fusion bayésienne et la fusion par la DST. Nous avons donc effectué cette comparaison sur notre jeu de données autour d'un rond-point et une densité de trafic forte. C'est aussi l'occasion pour nous de comparer nos trois indicateurs. Le tableau R.1 montre que la fusion par DST donne des résultats supérieurs à la fusion bayésienne. puisque le F1-score et le CR sont des métriques intrinsèquement liées, nous n'utiliserons que l'IoU dans les prochaines lignes.

PRISE DE DÉCISION Nous avons remarqué un comportement étrange au premier abord puisque tous les résultats étaient identiques pour

toutes les méthodes de prise de décision ainsi que pour les deux règles de combinaison de la DST : la règle conjonctive et la règle disjonctive. Cela est dû au fait que les résultats finaux sont obtenus après une comparaison entre les labels. Ainsi, si le classement entre les labels est le même, même si les valeurs absolues sont différentes, nous obtenons le même résultat final.

TAUX D'AGENTS CONTRIBUTEURS Nous avons remarqué que la méthode bayésienne offre ses meilleurs résultats lorsque le taux de participation est faible pour atteindre un mIoU de 54.86 % avec 3 % de véhicules contributeur et seulement un tiers des points de vue infrastructure en fonctionnement. Toutefois, dans tous les cas, la méthode basée sur la DST est supérieur à la méthode de fusion bayésienne.

La fusion basée sur la DST quant à elle semble atteindre un seuil sur l'IoU moyenne d'environ 57 % pour 50 % de véhicules connectés et un tiers de l'infrastructure ainsi qu'un seuil d'environ 60 % pour 50 % de véhicules connectés et tous les points de vue de l'infrastructure en fonctionnement.

DENSITÉ DU TRAFIC Nous avons remarqué que, aussi bien pour la méthode de fusion bayésienne, et celle basée sur la DST, les performances sont meilleurs pour une densité de trafic moyenne. On relève un IoU moyen de 52.59 % pour la fusion bayésienne et de 62.05 % pour la fusion basée sur la DST. Cependant, l'écart entre les deux méthodes semble augmenter avec la densité du trafic.

R.5 CONCLUSION

Dans ces travaux de thèse, nous avons établi un état de l'art sur la perception coopérative dans le contexte automobile. Nous avons notamment listé les modalités de perception les plus communes, les méthodes de communication ainsi que les problèmes soulevés. Nous avons ensuite discuté des méthodes de localisation, de détection et suivi d'objets et de cartographie coopérative avant de lister les scénarii et expérimentations impliquant de la perception coopérative.

Nous avons ensuite proposé une nouvelle méthode prenant aussi bien en compte les points de vue *dans-la-scène* des véhicules ainsi que les points de vue en hauteur qu'offrent les infrastructures pour générer une carte d'occupation de l'environnement. Nous avons décidé de nous limiter à l'utilisation de boîtes englobantes 2D afin de respecter les limitations imposées par le réseau. Nous avons testé deux approches pour fusionner les points de vue : une approche bayésienne et une approche basée sur la théorie de Dempster-Shafer qui nous a permis d'obtenir une carte cohérente avec la vérité terrain. Finalement, nous avons poursuivi le développement de notre approche pour y inclure l'aspect sémantique et lui apporter des amméliorations. Nous avons aussi effectué une validation plus approfondie avec un ensemble de jeux de données proposant différents scénarii. Nous avons pu montrer lors de notre étude qualitative que notre approche fonctionne

tandis que les méthodes de l'état de l'art proposé dans COLMAP ne parviennent pas à obtenir de résultat. Nos améliorations ont significativement amélioré les résultats suite à une fusion bayésienne, mais les résultats issus d'une fusion basée sur la théorie de Dempster-Shafer restent systématiquement supérieurs.

Les travaux de cette thèse ouvrent de nouvelles perspectives encore inexplorées. Cependant, pour continuer le développement de cette approche, il est nécessaire d'obtenir de véritables jeux de données coopératifs. Une étude devrait aussi être faite pour la gestion du bruit de pose des capteurs et du bruit de détection aussi bien sur la génération de grilles locales que sur l'assignation de valeur de probabilités ou de masses. Pour nous concentrer sur la partie coopérative, nous n'avons pas traité l'obtention des boîtes englobantes 2D dans les images. Il serait intéressant de tester la robustesse de notre système en fonction de la distance, des perturbations météorologiques et du bruit de classe. Il serait intéressant d'inclure des informations optionnelles dépendantes des capacités perceptives des agents comme des boîtes englobantes 3D ou des informations de distance.

ONBOARD/OFFBOARD EXTENDED PERCEPTION FOR AUTONOMOUS NAVIGATION

INTRODUCTION

Terrestrial transport is a fundamental element of our society and has been evolving continuously, adopting a wide variety of technical developments. As soon as steam engines appeared, the idea of creating machines dedicated to transportation was born, giving birth to the automobile. However, with the increase in speed, we have also observed an increase in accidents, mostly caused by human error. In response to this problem, ITT Automotives introduced the first electronically controlled anti-lock braking system (ABS) in 1969 to improve braking performance. Other systems such as electronic stability program (ESP) or traction control have been introduced to help drivers. The common point of these driving assistance systems is that they are based only on proprioceptive sensors and do not interact with the environment.

With the progress of embedded computer technology since the early 2000s, driver assistance systems have become more complex and have become advanced driver assistance systems (ADAS). The latter are equipped with exteroceptive sensors and can interact with their environment. This is notably the case of lane keeping assistance (LKA) based on a camera observing the road and an actuator inflicting a slight correction on the steering wheel. There is also the automatic emergency braking (AEB) using a radar to slow down the vehicle to maintain a safe distance or to brake in case of risk of collision. Finally, we can also cite the blind spot warning (BSW) that warns of the presence of obstacles.

Naturally, with the multiplication of ADAS, the idea of autonomous vehicles and autonomous navigation has been reinforced. However, autonomous navigation requires a perfect understanding of the environment in order to adapt to each situation. One solution to this problem is the multiplication of sensors on the vehicle in order to reduce the blind spots as much as possible and to surpass the perception capacities of humans. Nevertheless, some obstacles remain such as sensor limitations (e.g. angular resolution, dynamic range, etc.), external disturbances (e.g. weather, ambient light, etc.) or limitations inflicted by the environment in which the vehicle is navigating (e.g. occlusion due to buildings or other vehicles).

Since the beginning of the 2010s, a solution seems to emerge: connecting vehicles to each other. Indeed, if vehicles share their state (position, speed, direction), then it becomes possible to anticipate situations that were unpredictable until now. However, this does not take into account other objects like pedestrians and other inanimate obstacles. Moreover, not all vehicles are equipped to communicate. This is how the cooperative perception was imagined where vehicles are equipped with sensors and transmit what they detect to each other. However, today the proportion of connected vehicles and, among them, the proportion of vehicles equipped with sensors is still very

low. It is also to address this issue that lane-side sensors have started to be introduced into this equation. The other major advantage of point-of-views is that they are often positioned high up and offer a new perspective to understand the environment.

Today, the methods of cooperation are still disparate and poorly coordinated despite efforts to standardize them, while raising new technical challenges. Thus, the work carried out during this thesis has two major axes which will be represented by three chapters in this manuscript. The first axis is to establish a state of the art of cooperative perception in the automotive context in order to identify the different projects, methods, approaches and difficulties encountered so far. The second axis consists of two contributions in which we present a new approach of cooperative perception on board / off board. This approach will first address the presence or absence of obstacles in a scene before adding a semantic dimension in a second step.

In the first chapter, we present the state of the art of cooperative perception in the automotive context. We start by discussing the sensors and data most frequently used to perceive the environment and then we present the different communication modes to share the acquired data. We also discuss the approaches and architectures of cooperation as well as the difficulties encountered in this domain. We then present and evaluate the methods and performances of cooperative localization, object detection and tracking before discussing the methods for generating cooperative maps. Finally, we discuss the scenarios to which cooperative perception responds as well as related projects.

In the second chapter, we note that cooperative approaches involving infrastructure do not take into account data from connected vehicles. We therefore propose an approach using all available data to generate a dynamic object occupancy map and the associated cooperative architecture. In order to keep the impact on the network infrastructure as low as possible, we have decided to use only 2-dimensional information, such as that captured by ADAS currently available on the market. We also present two data fusion methods: one based on Bayesian theory and the other based on Dempster-Shafer theory (DST). Our approach is tested on a dataset built from the CARLA simulator. In our qualitative evaluation, we show that our approach works and allows us to find the position of vehicles in the scene with very little information. Finally, our quantitative evaluation highlights the superiority of the DST-based solution over the most widely used state-of-the-art method based on the Bayesian theory.

In the third chapter, we take the idea presented in chapter two and add the semantic dimension. We start by updating the already presented architecture and then we present new methods for local grid generation needed for viewpoint fusion. After adapting the fusion methods to the semantic aspect, we propose several methods for decision making, an aspect that was not treated in the previous chapter. Finally, we perform an extensive study on a set of new and more complete datasets. This study allows us to evaluate the global performances of our approach and to study its response to the change

of the proportions of connected vehicles in the scene or the number of vehicles in the scene.

Finally, we conclude this manuscript with a summary of the key points presented and by discussing the perspectives of this work.

1

COOPERATIVE PERCEPTION IN AN AUTOMOTIVE CONTEXT

ASSOCIATED ARTICLE

[1] Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Rémi Boutteau, and Yohan Dupuis. "Survey on Cooperative Perception in an Automotive Context." In: *IEEE Transactions on Intelligent Transportation Systems* (2022), pp. 1–20. ISSN: 1558-0016. DOI: 10.1109/TITS.2022.3153815. URL: https://hal.archives-ouvertes.fr/hal-03608119/document.

1.1 INTRODUCTION

The concept of driverless cars is one of the landmarks of a futuristic world for generations. Already in 1939, General Motors (GM) initiated the first attempt of making this a reality by showcasing a radio piloted car [13]. Since then, the development of this technology has never stopped and is increasingly getting complicated over a wide range of fields such as perception, decision making, and control. After the pioneer works of GM, during the 1980s, Mercedes-Benz showcased the first autonomous car with a vision-controlled robotic van reaching a speed of 63 km/h on streets without traffic. This led to the creation of international projects and challenges such as the Defense Advanced Research Projects Agency (DARPA) Grand Challenge in 2004 consisting of autonomously navigating through the Mojave desert in 142 miles long course [107]. The next step was navigation in an urban environment through normal traffic conditions. In 2007, the DARPA announced the holding of the Urban Challenge that simulates an urban environment with streets, traffic lights, and human-driven vehicles. [108]. We can also note the VisLab Intercontinental Autonomous Challenge (VIAC) challenge in 2010 consisting of driving autonomously through a 13000 km long way from Parma in Italy to Shanghai in China [18]. Nowadays, several companies sell cars with the ability to offer an autonomous driving experience such as Tesla [31] or the Audi A8 [100]. The idea of cooperative vehicles quickly appeared and in 2011 the grand Cooperative Driving Challenge (GCDC) took place in the Netherlands in which vehicles had to perform the best in a platoon [36, 61]. The GCDC has been reiterated in 2016 to perform lane merging, driving in an intersection as well as emergency vehicle handling in a cooperative context [119]. Cooperation between vehicles can be extended to infrastructure and thus led to the project Providentia in Germany [48] consisting in creating a digital twin of a road section generated from the sensors of an infrastructure.

In our context, the perception task consists in the estimation of the status of the ego-vehicle in the scene as well as the environment elements surrounding it. We distinguish 3 subsections, the localization of the ego-vehicle, the detection and tracking of other users and, finally, the detection and representation of the environment (mapping). Cooperation represents the use of data provided by other agents to perform perception tasks or to refine their results. Cooperation can be performed at three levels of data sharing depending on whether the data is raw (early fusion), preprocessed data (mid fusion), or processed data (late fusion). Fig. 1.1 represents this pipeline with three steps and three blocks (namely: Localization, Object Detection and Tracking and Map Generation) performing the main perceptive tasks to understand the scene. In the early fusion stage, we represent the raw data fusion. In this stage, the data provided by the sensor at a given timestamp is aggregated and associated with a given transformation between sensors. The raw data comes from connected users which perform an early fusion. The raw data from the ego vehicle may also be shared with other users. In the second stage, we note two parallel tasks running. One estimates the vehicle's location in the environment from the sensors and can also benefit from other users' measurements as an aid. The second task performs the detection and tracking of objects in the scene. It can also benefit from the data of connected users to densify the global perception of the environment. Both together perform the heart of the perception outputting feature level data shareable with other users. The last stage aims to build a map, hence giving context to the previously acquired data. It is based on the use of a given prior map and can also be updated cooperatively by connected users. This block diagram tries to briefly showcase the classical scheme of a cooperative Vehicle-to-Everything (V2X) perception pipeline. However, reality offers a broader range of architectures with their specificities and a certain amount of challenges when realizing them, which is exposed later in this survey.

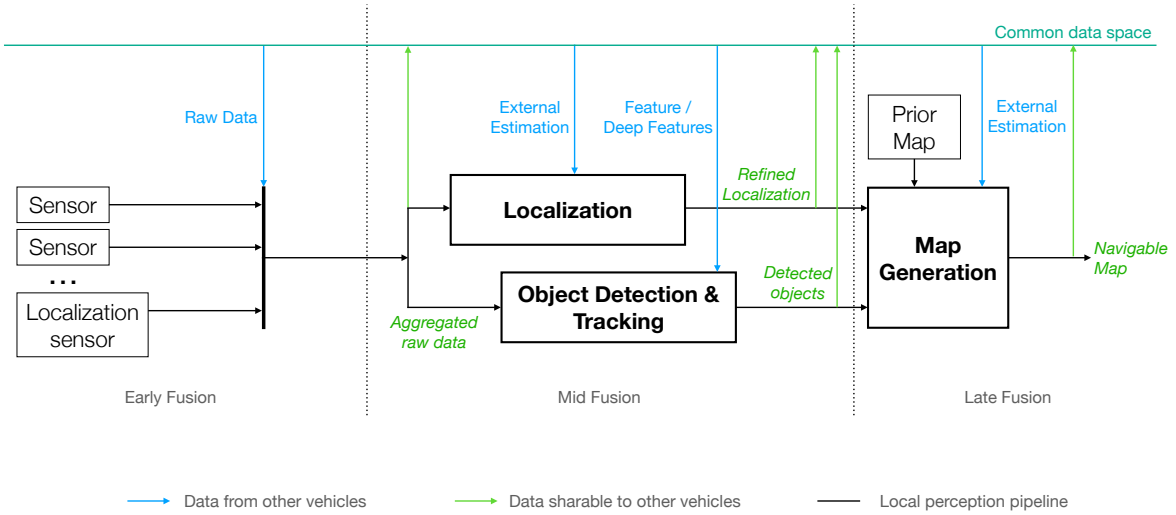


Figure 1.1: Block diagram of the minimal perception pipeline in a vehicle (in black). We can distinguish three main stages able to share the locally produced data (in green). Each of them can receive data (in blue) to perform their task cooperatively.

This chapter aims to provide a state of the art of cooperative perception methods for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). We have organized the chapter in an order that respects the data flow, divided in six sections. Section 1.2 focuses on the creation of cooperative systems from a general point of view. We particularly review the challenges brought by cooperative systems, the possible architectures, and the available communication facilities. We also present a review of frequently used sensors along with their performances in a non-cooperative environment to provide a reference as a basis for comparison. Section 1.3 lists the cooperative methods of locating the ego-vehicle in the scene. Section 1.4, for its part, reviews the methods of detection and tracking of objects in the scene. Section 1.5 reviews the role of maps and their usage in a cooperative context. In Section 1.6, we propose to summarize the cited techniques through a summary table and we propose a Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis. In section 1.7 we review the scenarios in which cooperation brings real advantages illustrated by experimentations. Finally, We list the datasets available to unlock work perspectives before providing our conclusion in section 1.8.

1.2 BASICS OF COOPERATION

The ways of creating cooperative perception systems are multiple and require to assess several types of architecture. Each design has advantages and disadvantages and will deeply affect how the system will react as well as its strengths and its weaknesses. Another unavoidable point of any cooperative system is the communication facilities which define what data can be shared as well as the formats available. These two points will be tackled in this section but we will start by briefly reviewing the results available in the non-cooperative methods based on the same sensors widely used in the cooperative counterpart to get comparison points.

1.2.1 *Sensing Modalities*

Sensors are the basics of any perception system as they allow us to sense ourselves as well as the surrounding environment. Since the sensors we are going to discuss have already been presented in numerous articles, we will rather focus on their performances. In [65], Kuutti et al. brought a survey introducing the sensors and comparing their performances in a positioning context and therefore inspired the following structure. In table 1.1, we provide an overview and a comparison of the most widely used sensors.

1.2.1.1 *Global Navigation Satellite System*

When it comes to knowing our position, the satellite positioning system is the most widely used. Initiated by the United States with the Global Positioning System (GPS), several countries contributed with new satellite constellations. The pure GPS has an error of up to

20 meters [104] but several methods have been used to refine these results. However, since the GPS has an update rate up to 20 Hz, it is often associated with an Inertial Measurement Unit (IMU) bringing a high updating rate [74]. An association of a pure GPS with an IMU showed they could achieve an error of 7.2 meters (Root Mean Square (RMS)) after a path of 408 meters [125].

One of the problems encountered with pure GPS is its first acquisition time. The Assisted GPS (AGPS) brings a solution to this by using the cellular network to download the almanacs and hence reducing the downloading time from the slow satellite connection. However, it does not bring any precision improvement. Unlike the AGPS, the Differential GPS (DGPS) allows a reduction of the error up to 1 to 2 meters in the covered zones [101]. The arrival of the Real-Time Kinematic GPS (GPS RTK) achieved unprecedented performance with an error of a few centimeters range [65]. Similarly to the AGPS, both DGPS and RTK do use a terrestrial infrastructure to download the satellites' almanacs via an internet connection. Although we do not consider Global Navigation Satellite System (GNSS) technology as a cooperative system, it is one as it features several vehicles (terrestrial users and satellites) and infrastructures.

Nowadays, pure GPS had been replaced by the GNSS, currently based on several satellite constellations such as the American GPS, the Chinese BeiDou Navigation System, the Russian Global Navigation Satellite System (GLONASS), the Japanese Quasi-Zenith Satellite System (QZSS) and the European Galileo. Using the Real-time extended (RTX) technology, the Root Mean Square Error (RMSE) achieves a 2.9 cm accuracy [84].

1.2.1.2 *Camera*

Cameras can be used to detect and track obstacles (pedestrians, cars, animals) as described by Arnold et al. [6]. Formerly, these tasks were mostly based on a geometric approach to the problem, but machine learning and deep learning methods have taken over the state-of-the-art. Hence, nowadays, most efforts are based on machine learning solutions.

Another field of application for cameras is trajectory estimation, especially with visual odometry [94]. This technique consists in recognizing key points in a frame and then finding them in the following frames to estimate the displacement of the camera. We can note that this method is sensitive to error accumulation over time. This principle is extended in the Simultaneous Mocalization And Mapping (SLAM) algorithms with the difference that the perceived environment is kept to create a map and estimate its position with an accuracy of 75 cm [70]. Nowadays, new methods featuring Deep Learning bring even better results such as DeepSLAM proposed by Li et al. in [71] which gives a mean translation RMSE drift of 5.58% and a mean rotational RMSE drift of $2.47°/100m$ alongside a 100 m to 800 m path. Since visual odometry and SLAM are based on the notion of optical flow,

the arrival of event-driven cameras with hardware adaptation offers promising results in both localization and classification.

However, monocular systems pose a limitation on the estimation of the position on the depth axis of the images. One solution is to use two or more cameras to create a stereoscopic vision system and to synchronously search in both cameras for corresponding interest points. Another solution to get the depth information from a monocular system is to use a deep learning algorithm [44]. In addition to these techniques, Camera-Light Detection and Ranging (LiDAR) or Camera-Radar coupling has been extensively investigated in the state of the art.

1.2.1.3 *Radio Detection and Ranging (RADAR)*

Compared to cameras, radars have a lower angular resolution. This characteristic makes them less suitable for the classification of perceived objects. However, their accuracy in distance and speed measurements is much better than cameras and they are therefore used in addition to the latter as in the Providentia project [48].

The concept of visual odometry has been adapted to the radar device. A high-speed rotating radar has allowed a position estimation with an error of 12 meters despite the distortions due to the rapid rotation [111]. Another system using Short Range Radar (SRR) allowed an estimation with an RMS error of 7.3 cm on the lateral axis and 37.7 cm on the longitudinal axis in [114]. In the same way with the SLAM, an experiment allowed a localization with a mean error of 9 cm and a standard deviation of 38 cm [111]. Nevertheless, radars can penetrate certain materials, notably those that compose the ground. Thus, a method based on the mapping of underground terrain has allowed a localization with a precision of 4 cm and is presented in [28]. Despite advantages such as insensitivity to weather conditions, the authors specify that further researches are needed to create robust maps to multi-path effect or to characterize reflections induced by vehicle's chassis.

1.2.1.4 *LiDAR*

LiDARs (Light Detection And Ranging) can be considered as an intermediary between radar and camera. They provide a list of points in a three-dimension space. These points are extracted from the angle formed by the laser beam and the distance from the sensor and the impact. To get the distance, there are several techniques. The most common one is based on the Time of Flight (ToF) principle, but we can cite other techniques such as the Frequency Modulated Continuous-Wave (FMCW) or the Amplitude Modulated Continuous-Wave (AMCW) [44]. Since the angular resolution is thinner than the radar, we can classify detected objects besides being able to locate them more accurately [23, 77].

In the same way as what we have seen with previous sensors, the principles of visual odometry and SLAM can be adapted to LiDAR

sensors. In [68], a GPS, IMU and wheel odometry have been combined within a SLAM framework that allowed a localization estimation error between 10 and 30 cm. An improvement of the SLAM and an implementation of dynamic maps allow an error of 9 cm in a dynamic environment [69]. By projecting the ground on a grid invariant to the laser perspective, a position estimation with an RMS error of 3.3 cm on the longitudinal axis and 1.7 cm on the lateral axis has been performed in [21].

Halfway between cameras and LiDARs, ToF cameras, made of a sensor similar to cameras are based on measuring the ToF taken bay the light to return to the sensor. They provide depth images that can be related to point clouds generated by the LiDARs. By using them in a visual odometry algorithm, Chen et al. were able to estimate the trajectory with an absolute trajectory error (ATE) of 78 cm on a 25-meter path [24].

1.2.1.5 *Ultrasonic*

The majority of vehicles sold today carry ultrasonic sensors. The drawback of such sensors is that they have a very low angular resolution that requires a too important calculation cost. Also, they are highly sensitive to weather conditions and the Doppler effect when objects are moving fast and have a short-range [65]. These elements make this sensor unsuitable for applications of obstacle localization and classification.

1.2.1.6 *Radio Frequency (RF) based methods*

Wireless communications are mandatory in a cooperative environment hosting mobile users. However, they can be used as sensors, especially to estimate the position of a receiver. Various sources of radio signals can be used, such as the cellular network or infrastructure made up of anchors, as in the case of Ultra Wide Band (UWB) systems allowing centimeter-scale location [30].

Position estimation methods are generally based on measuring the distance between the transmitter and the receiver. Thus there are four main methods for position estimation :

- Received Signal Strength Indication (RSSI): RSSI based method that consists of measuring the signal strength to measure the distance between the transmitter and the receiver based on the electromagnetic permeability and the diffusion factors of the environment. A distance measurement allows us to position ourselves on a circle surrounding the transmitter base, but, as shown in Fig. 1.2, it is impossible to know where on this circle. To eliminate ambiguity, it is necessary to make at least three measurements to find the common intersection of the three circles.

- Time Of Arrival (TOA) and Time Difference Of Arrival (TDOA): These methods that use the transmission delay of a signal be-

tween its emission and its reception. Since the speed of an electromagnetic wave is known, it is possible to find the distance between the two devices. In the same way as the RSSI-based method, at least three measurements are necessary to estimate the position of the receiver.

- Angle Of Arrival (AOA):, Unlike the other two methods, AOA method, is based on measuring the angle formed by the direction of the received signal. This angle associated with the position of an anchor forms a straight line on which the vehicle is located. With a second measurement on another anchor, a second straight line is obtained which intersects the first one at the position of the vehicle as illustrated in Fig. 1.3.

- Fingerprint: This method is based on the specificity of the environment and in particular on its capacity to alter the strength of a signal and to reflect it (multi-path). The aggregated information is compiled into a map allowing us to match the received signals to a position.

Typical setup

The listed sensors succeed to achieve their tasks but also suffer from shortcomings. Therefore, sensor fusion is mandatory to get over the limitations of each one. We already mentioned the fusion between a GNSS receiver and an IMU to improve the localization performance. Similarly, vehicles or infrastructures embed several types of sensors. A usual setup for autonomous cars is constituted of GNSS - IMU to achieve global localization with cameras, laser scanners or RADARs for detection and tracking of elements in the scene or as another source of localization information. Infrastructure also embeds sensors such as cameras and laser scanners or RADARs to locate users as seen in [48, 119].

Sensor	Given data	Environement's impact	Advantage	Disadvantage	Performances
GNSS	Absolute position	Requires at least 4 satellites in sight of view and is sensitive to the canyoning effect in urban environment.	The system doesn't require an initial position to give a result and can be used in an unknown environment.	The result is outputted once per second and the reliability of the signal depends on the services coverage.	Pure GPS: 20 m Pure GPS + IMU: 7.2 m error GNSS RTX: 2.9 cm
IMU	Relative position	The system is not affected by the environment.	Ability to output a result at a higher frequency than GNSS.	The error accumulate as the time passes and is affected by the precision. The higher the precision is, the higher the price is.	Estimated bellow 7.1 % Relative Error for MPU-9150 [109]

Radar	Distance and relative speed	Affected by weather conditions (mainly rain but also snow, mist).	Long range perception and hardware speed measurement possible	Poor angular resolution making object classification harder	Angular accuracy: 0.5° to 5°; Speed accuracy: $0.2ms^{-1}$; Perception range: up to 250 m; Sampling rate: up to 20 Hz
LiDAR	Point cloud	Affected by weather conditions (mainly fog but also rain).	Compromise between radar and camera allowing a physical measure of the distance but with a lower angular resolution.	The sparseness of the point cloud makes it hard to difficult to sense the texture.	Angular accuracy: 0.03°; ranging accuracy: 10 cm to 2 cm; Perception range: 80 m to 200 m; Sampling rate: up to 100 Hz
Camera	Image	Affected by weather conditions and brightness.	Sense color and textures facilitating segmentation and classifying.	Although it can be estimated, there is no direct depth measurement.	Highly dependent on the sensor and associated optics.
Ultrasonic	Distance from obstacle	Affected by weather conditions	Low cost sensor	Small detection range and high sensitivity to Doppler effect	Maximum range: 6 m

Table 1.1: Sensor comparison based on [6, 10, 84, 85, 87, 104, 123, 125]

1.2.2 *Communication*

In the previous section, we have reviewed the most used sensor in an automotive context. In a cooperative context, we want to share the generated data, raw or processed, with other agents with the aim to densify the image of a covered area. Thus, it is mandatory to discuss the communication facilities available, which is the aim of this section. We will focus on the ways to wrap the data they produce and how to share them. Then, we present some of the most widely used communication facilities. We also consider new approaches.

1.2.2.1 *Wrapping and sharing the data*

To share data, users have to choose a specific network architecture. The most common is the Vehicular Ad-hoc Network (VANET) architecture consisting in connecting every vehicle in the range from each other [38]. In VANET, a channel is common to every vehicle to coordinate the network. The data is shared on different channels and routed by hopping on vehicles between the sender and the receiver. To assess the physical layer's requirement in a VANET network, an amendment of the IEEE 802.11 was added to create Wireless Access in Vehicular Environments (WAVE) (IEEE 802.11p). In Europe, the IEEE 802.11p standard was used to create the ITS-G5 standard [40]. In the same way, two communication protocols are based on these two standards which are respectively the Dedicated Short-Range Communication (DSRC) [60] and the Cooperative-ITS (C-ITS) [40]. Table. 1.2 gives an overview of both of the standards and their components compared to the OSI model as given in [40, 60]. We can note the presence of Basic Transport Protocol (BTP) and GeoNetwork which are defined in [40] as well as WAVE Short Message Protocol (WSMP), defined in [60] as facilities to achieve the network and transport layer tasks. The specificity of the GeoNetwork protocol is that it bases itself on the geographical position of the agents to determine the path to follow for the data.

The information shared with DSRC protocol is wrapped in Basic Safety Messages (BSM) [60] which convey information about the emitting vehicle to avoid collisions. Similarly, C-ITS introduces the Cooperative Awareness Messages (CAM) also conveying vehicle information as the BSM but also introduces the Distributed Environment Notification Messages (DENM) which notify hazards on the road and which has a higher priority than the CAM [38]. CAM and DENM messages proposed with C-ITS are used by [119] but the authors also needed to use another type of message, the i-GAME Cooperative Lane Change Message (iCLCM), to indicate to other vehicles their willing to change lane. Authors in [77] used the Signal Phase and Timing (SPaT) messages to anticipate the traffic light changes and used the DSRC's BSM to notify the presence of detected vehicles by the infrastructure. To respond to these new needs, messages such as SPaT but also the messages for road topology data (MAP), for special vehicles (SRM, SSM), for probe vehicle data (PVD, PDM), and in-vehicle information (IVI) are being standardized [40].

<table>
<tr><td>Application</td><td>Other App. Layer</td><td>Safety App. Layer</td></tr>
<tr><td>Pre-Application</td><td colspan="2">CAM / DENM
BSM / SPaT / MAP / SRM / SSM</td></tr>
<tr><td>Transport</td><td>TCP / UDP</td><td rowspan="2">GeoNetwork / BTP
WSMP</td></tr>
<tr><td>Network</td><td>IPv6</td></tr>
<tr><td>Data Link</td><td colspan="2" rowspan="2">ITS-G5
WAVE</td></tr>
<tr><td>Physical</td></tr>
</table>

Table 1.2: Representation of the two protocols available in a VANET architecture given through the OSI model [40, 60]. The the C-ITS defined standard are given in green while the DSRC defined standard is given in blue. Both of them provide adapted answers for vehicular communication on the physical layer based on IEEE 802.11p as well as dedicated messages to encapsulate the data between the application layer and the transport layer.

Novel network architecture is used by Li et al. in [73]: the Software-Defined Network (SDN). This solution is placed between the VANET and the fully centralized network. The common network is thereby replaced with centralized architecture communicating with a controller which manages the interconnections between the road users dynamically.

Another common architecture used nowadays is based on the publisher / subscriber paradigm, mainly supported by the Robot Operating System (ROS) [88] which is frequently used in recent projects [4, 63, 64, 73, 110]. The structure is based on nodes communicating messages transmitted on topics. Each node can be a publisher or a listener and they can be placed on different devices on the same network. A master program runs and plays the role of a dictionary and is contacted by every node either to inform about the topic they publish on or to know which node to listen to for a specific topic. Messages transiting through topics and are very various and can contain coordinates, images, or point clouds. A new version of ROS (ROS 2) is being developed with some improvements regarding fleets of collaborative robots.

1.2.2.2 *Communication facilities*

A wide range of communication facilities has been proposed for tackling different needs. We have already mentioned WAVE and ITS-G5 which are based on Wireless Fidelity (Wi-Fi) (IEEE 802.11) but with a given frequency of 5.8 GHz in Europe as well as in Japan and 5.9 GHz in USA [12]. Authors of [35] used the IEEE 802.11p to establish a communication between infrastructure and a vehicle and used DENM to transmit the control messages and the position information. Chen et al. [23] similarly used DSRC, and thus WAVE, to share regions of

interest of LiDAR point clouds and indicate sufficient speed. Kim et al. [63] used Wi-Fi IEEE 802.11n and studied the impact of the delay on the position estimation error.

Even if the majority of the current solutions are based on IEEE 802.11 technology and its derivatives, other technologies can be used such as the cellular network. The advantage of it is its wider coverage and the already existing infrastructure [7]. 5G cellular network is particularly promising thanks to its features such as precise localization, high throughput, and low latency. As described in [56], Proviendentia takes the advantage of the 5G network to communicate between the different elements (back-end station, Road Side Unit (RSU), On-Board Unit (OBU)).

Emerging communication technologies are being explored by authors of [73] who used the Millimeter Wave (mmWAVE) [83] band to transmit the point cloud produced by the RSU to the OBU and noted a significant data throughput increase. Another technology studied is the Visible Light Communication (VLC) [5] which consists of using light-emitting diode (LED) arrays (e.g. traffic lights, car lights) to display patterns. VLC allows data rates up to 96 Mb/s but is sensitive to the environment [65]. Finally, UWB which is used for localization is capable of communication [93] with data rates tested up to 250 Mb/s in [52] and up to 1 Gb/s in [62]. However, to our knowledge, UWB is not used for data sharing in the Intelligent Transportation System (ITS) context.

1.2.3 *Designs and challenges*

Until now, we have reviewed the most used sensors used in the automotive context as well as the communication facilities available to share the generated data between agents. However, when several users interact with each other, we have to define the organization of the communication. We distinguish two main approaches: the centralized and the distributed ones. We discuss and compare these approaches in the next lines. Nonetheless, no matter the chosen approach, cooperation brings new challenges. We provide a review of these challenges following the discussion on organization approaches.

1.2.3.1 *Centralized approach*

The cooperative approach makes it possible to overcome the problems of non-cooperative approaches such as extending the horizon line. As an example, multiple points of view can be used to reduce the effects of obstructions while densifying the areas covered.

The centralized aspect of this approach concerns the processing of the acquired data. In this mode of operation, users share their acquisitions to a single point, for example, a road-side processing unit. This server is in charge of processing the data and extracting useful information from it, which are then shared with users. The Providentia project is based on this approach. Data acquired by the sensors placed on gantries on a section of highway are transmitted to

a roadside processing unit, which creates a digital twin of the section of road accessible to all [48]. Similarly, Lv et al. based their work on a centralized approach in which four LiDARs monitor an intersection and transmit their data to an RSU. Users are detected, located and their information is then relayed to other users [77]. The main disadvantage of this solution is that the efficiency of this architecture relies mainly on the processing power of the processing unit. In [99], Shi et al. proposed a solution to the throughput drop of the service model of [82] by introducing the cluster-based VANET which consists of linking sub vehicular network into a larger one. Data collected by each vehicle of a sub-network are filtered and stored on a server to be broadcasted under request which resulted in lower network usage and hence reduced energy consumption.

1.2.3.2 *Distributed approach*

In contrast to the centralized approach, the data acquired by the users are directly transmitted to all vehicles simultaneously. Therefore the processing of these data is done onboard for each vehicle. A typical case of decentralized management is presented in [119] by Xu et al. through their participation in the GCDC of 2016. Each vehicle was broadcasting its state and its maneuver intentions which allowed the event anticipation and improved the car control. However, the system used connected cars which are in range with each other limiting the size of the network. Li et al. proposed the use of the SDN in [73] to optimize the network usage and set up mmWAVE communication to increase the throughput allowing them to share raw LiDAR point clouds. To solve the problem of disconnection in a sparse fleet, Zheng et al. proposed in [126] the use of the cellular infrastructure to create a heterogeneous network. The common point of these applications is that data of every vehicle is processed onboard on each vehicle. However, the coverage quality depends on the size of the user fleet [20].

1.2.3.3 *Centralized vs Distributed approaches*

As stated before, both centralized and distributed approaches have several advantages and weaknesses, as shown in table 1.3. We can observe that the distributed approach is the most common because, nowadays, the majority of cooperative applications are based on V2V approaches. However, applications based on centralized approaches are increasingly present today, especially within projects such as MEC-View [16, 42] and Providentia [48].

	Distributed	Centralised
Advantages	• Reliable in case of failure of an element • Available everywhere	• More data aggregated • Global view of the scene • More computing power
Disadvantages	• Limited computing power • Network less optimised (duplicated data) • Synchronisation	• Synchronisation • Converging network (Possible bottleneck effect) • Latency between the sensor and the received information

Table 1.3: Advantages and disadvantages observed between distributed and centralized architectures.

1.2.3.4 *Challenges of cooperation*

As we have seen, the architecture of a cooperative system dramatically impacts the efficiency of a cooperative system. However, this is not the only challenging part of cooperative systems. As well as for non-cooperative systems, difficulties brought by the type's diversity from the data acquired from the sensors exists as well as the one from the calibration of the acquisition hardware. But, to them, other challenges are added such as the synchronization between the actors, the extreme difference of point of view, or the matching of the receiving data with the locally acquired one.

MULTI-MODALITY Multi-modality is the one we are the most aware of since it appeared in the early time of robotic perception. Indeed, this challenge appears as soon as several types of sensors giving data of different nature are used within a system. Some projects avoided this problem such as Lv et al. [77] who decided to solely use LiDARs as well Chen et al. in [23] and Li et al. in [73].

Another project trend is to use the different sensors for an application to merge the results to improve the reliability or to enrich the properties of a detected item. As explained by Hinz et al. [56], the Providentia project uses cameras and radars to sense the environment. The choice of multi-modality has been made to answer different needs which are the detection and the classification, performed by the cameras, and the distance and speed measurement, performed by the radars. Later in this chapter, we assess the way of merging the streams of data given by the sensors with three different approaches: early fusion, late fusion, and deep fusion.

Another way to solve the multi-modality and calibration challenges is to process the data locally for each sensor and share the output in the form of messages. The above-cited project of Lv et al. [77] uses this principle to share the detected vehicles facilitating the broadcasting with smaller data. However, data association is a challenging topic that must be performed afterward during the aggregation step. The GCDC 2016 offers an answer based on the choice that users broadcast their states and intentions only avoiding duplicate data and assignment tasks. Xu et al. used in [119] a LiDAR to perceive vehicle in front of the ego-vehicle on both lanes by looking for clusters of points. As reported by the authors, these clusters could be associated with the messages sent by other vehicles on the map with their coordinates.

CALIBRATION Calibration is the other most known challenge in the perception pipeline. The calibration in a cooperative environment aims to determine the transformation between the sensors to be able to merge acquired data from several views at least at a given frame. If this task can already be challenging on a single agent, it becomes more laborious in a multi-mobile user environment. In this situation, the transformation matrix between sensors constantly changes as the vehicle moves in the scene, featuring long baselines. Moreover,

synchronization is arduous due to the absence of a physical triggering line.

Fortunately, to calibrate an infrastructure, manual measures can be sufficient and remain simple to conduct. Lv et al. [77] calibrated their infrastructure by measuring the distance between the four sensors placed at each corner of the intersections.

Similarly, it is possible to semi-automate the calibration process in the same way as the vision calibration with a chessboard. Krammer et al. describe in [4] the calibration procedure of the Providentia project: cameras have been intrinsically calibrated using a chessboard and the radar was calibrated by using the built-in tools based on the vanishing point method. We can note that for a cooperative project, the baselines encountered in the scenes are much wider than the ones met locally on a vehicle. Thus, a similar application might bring an answer to calibrate infrastructures with a large baseline and very different point of view: the Motion Capture (Mo-Cap) systems. Yang et al. give in [120] an example of calibration with multiple Microsoft Kinects v2 synchronized through Network Time Protocol (NTP). The system uses a calibration wand to fix the common origin between the cameras similarly to several other commercial systems (e.g. Vicon, OptiTrack). Unfortunately, the use of a calibration wand will encounter laser scanners or radars limits: their low angular resolution. In [117], Xia et al. propose a state of art for global calibration of non-overlapping cameras. Some of the presented methods could apply to cooperative roadside infrastructures such as the methods based on Structure From Motion (SFM) or the visual measuring instruments consisting in locating landmarks with a known position in the sensor data to recover the position of the sensors.

However, none of these methods helps when mobile acquisition platforms appear. Nowadays, the most widely used method, in this case, relies on absolute coordinates and hence relying on the pose estimation performance assessed in the Perception part.

SYNCHRONIZATION Synchronization is another major challenge to consider. In a cooperative context, calibration relies on the synchronization of the elements to determine the transformation between the sensors, especially with the mobile sensors. There are multiple sources of desynchronization such as an offset between the clocks or the communication delays. Although clocks are synchronized, we cannot ensure their acquisition are triggered at the same moment which adds uncertainty at the moment to merge the acquired data. Similarly, different sampling rates require interpolation between acquired or predicted data, also adding uncertainty.

In a local system such as a car or an infrastructure, physical lines can be used to trigger and thus synchronize the sensors together. However, this solution cannot be used in a cooperative context since some users are mobile.

In [63], the authors showed that the delay induced by the communication can significantly affect the position estimation and thus estimate

delays between the users to match the timestamps of acquisition to reduce the delay's impact.

Another solution can be found by using the NTP to synchronize the users. This is the solution given by Yang et al. in [120]. As we mentioned earlier, they use the NTP protocol to synchronize their Kinect to perform their acquisitions. Nevertheless, while adapting the NTP to the automotive network seems to be a reasonable solution, it brings the question of which user provides the clock. A natural answer could be to use the infrastructure's clock but we know by experience that they are not always accurate (e.g. clock provided by the Radio Data System (RDS) data from the local radios). Another answer is to use the GPS timestamps and the triggering signal they provide with a Coordinated Universal Time (UTC) format offering a basic accuracy of $2\mu s$ [112], widely used nowadays.

Movement-based synchronization can also be an answer but highly depends on the calibration stage and requires an overlapping area in the acquired data.

POINT OF VIEWS Point of views can be extremely different in a scene featuring infrastructure and mobile users. Thus ask the question on the fundamentally different looking of a single object which can even be considered as non-overlapping data. An example could be a sensor observing the front left corner of a car and another sensor observing the right back corner of the same car. The Mo-Cap systems can bring an answer to this section here as well by trying to match the perceived object with a skeleton or a bounding box and fitting them together.

Another question comes with the mix of mobile and static users. In [79] Merriaux et al. show that LiDAR scans are affected by the movement and demonstrate that the rectification of the point cloud brings better results at the merging step. To our knowledge, there is no study of a fixed laser scanner with moving objects but we can suppose that some alteration can be caused on the scanned moving structures.

PERCEPTION MATCHING Perception matching between objects sensed by others and shared to the ego vehicle and the object sensed by the local sensor is a typical challenge of a cooperative system and is rarely assessed in the works we have seen. A basic solution is to match the object with their positions as in [119] but the noise induced by the sensors can lead to errors. Similarly, we can use features describing a vehicle. The position can indeed be a feature and we can add to them more features. This is what Kim et al. do in [63] by using the speed of the vehicle as the key feature to match the shared data with the perceived objects.

With a more mathematical approach, Miller et al. propose in [81] a solution based on the bipartite graphs which are based on the graph theory. However, the limitation of the bipartite graphs seems that the data can be associated with only two participants. Thus, it can perfectly fit with a centralized architecture with each participant fitting their observation with the one stored by the infrastructure.

1.3 LOCALIZATION

As we have seen earlier, some non-cooperative methods manage to reach the constraints of $0.3m$ given for in-lane autonomous navigation [47, 65] in optimal conditions. However, non-cooperative approach is limited by sensor capabilities such as the GPS coverage density significantly affecting its performance as well as weather and light conditions affecting optical sensors such as cameras and LiDARs. Indeed, the multiplication of the estimations makes it possible to eliminate the outliers as highlighted in [43]. Moreover, cooperative systems allow the extension of covered areas and fields of view, which again increases the reliability and precision of the estimations [50, 55, 63]. The other interest of cooperation in a localization context lies in the reduction of costs. The improvement of the accuracy and reliability of a sensor is generally proportional to its price. However, they can be improved by multiplying the number of sensors distributed over other users or infrastructures hence reducing the cost of each vehicle [50].

The cooperation can be implemented at several levels of estimation from the lowest level by sharing raw sensor results to a higher level by sharing estimated coordinates. In the first case, the objective is rather to extend the coverage of services either because they are inaccessible (e.g. GPS in a tunnel) or because the vehicle is not sufficiently equipped.

1.3.1 *Low-level cooperative position estimation*

One of the most commonly used examples of cooperative position estimation today is GNSS. GNSS uses the multilateration technique to estimate the position of a point by measuring the distance between it and several anchors as explained for TDOA or TOA algorithms (Fig. 1.2). Here, GPS satellites are used as anchors with their known positions since, in addition to transmitting the time of transmission allowing to estimate the distance between the satellite and the receiver, they also transmit their orbital parameters (almanacs) allowing to recalculate their position depending on the date.

1.3.1.1 *Multilateration*

Because of the effectiveness of multilateration, this method has been adapted to other sensors from other vehicles or infrastructures. In particular, Rohani et al. in [91] made a simulation with a GPS and a measurement of the distance between vehicles, obtaining an error ranging from 3.3 m to 6.75 m depending on the quality of communication with other vehicles. The maximum error corresponds to the error of the GPS alone which shows that, in this case, the cooperation only adds a better accuracy to the GPS but does not degrade it in case of bad conditions.

To reduce the impact of poor communication, it is possible to apply weight on distance measurements. This is notably what Ahammed et al. propose in [2] by applying weight on the measurements depending on the distance between the two entities leading to an average error of

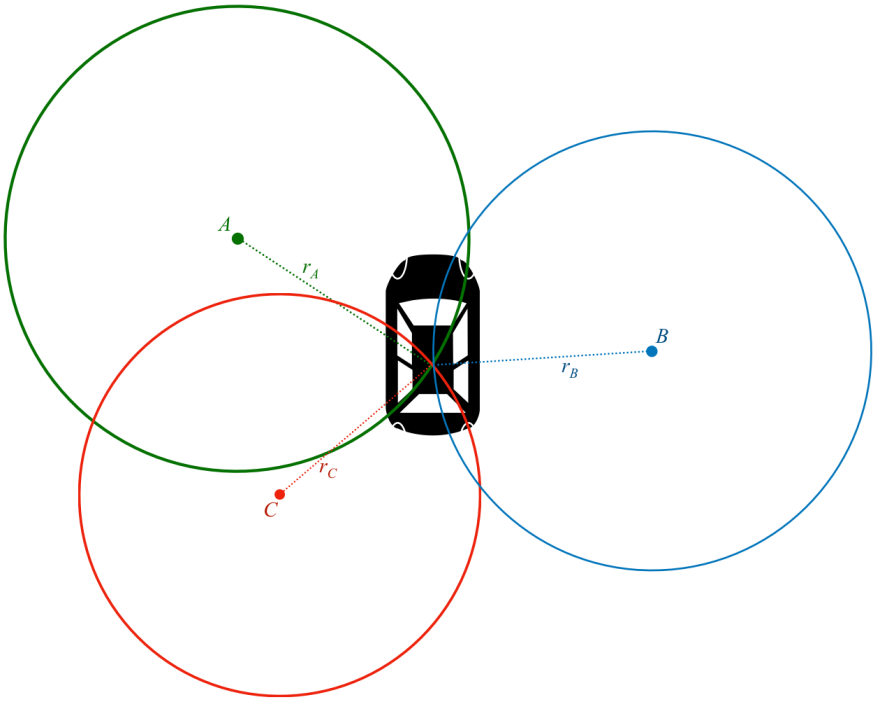


Figure 1.2: Illustration of the multilateration principle. A, B and C represent users or infrastructure points with known locations. The multilateration allow to find the location of the vehicle from the distances r_A, r_B and r_C and the positions of A, B and C.

2.38 m on a fleet of 10 vehicles. Similarly, Altoaimy et al. in [3] apply weight on position estimates using the signal to noise ratio (SNR) on the communication used to estimate the distance between entities. The simulation of this scenario leads to errors of 85 cm with 20 vehicles and 25 cm with 200 vehicles.

Although these results are not accurate enough for stand-alone navigation, it is important to note that they were obtained using GPS only as a base. Therefore, the use of other technologies may lead to better results, such as the work by Del Peral-Rosando in [86] using a TDOA algorithm on 5G cellular network antennas estimating the position of the receiver with an error between 20 cm and 25cm.

1.3.1.2 *Triangulation*

In the same way, as for multilateration, triangulation makes possible the estimation of the position of a receiver in an environment equipped with anchors. However, where multilateration uses the measurement of the distance between the receiver and the anchors, triangulation uses the angle of incidence of the signal emitted by the anchors. Triangulation is therefore the principle on which the AOA approaches are based, as illustrated in Fig. 1.3.

However, the authors of [54] note that the multilateration approach obtained better results at the middle of the network but that the triangulation approach became more efficient at the edges of the network. The authors, therefore, propose the implementation of hybrid TDOA

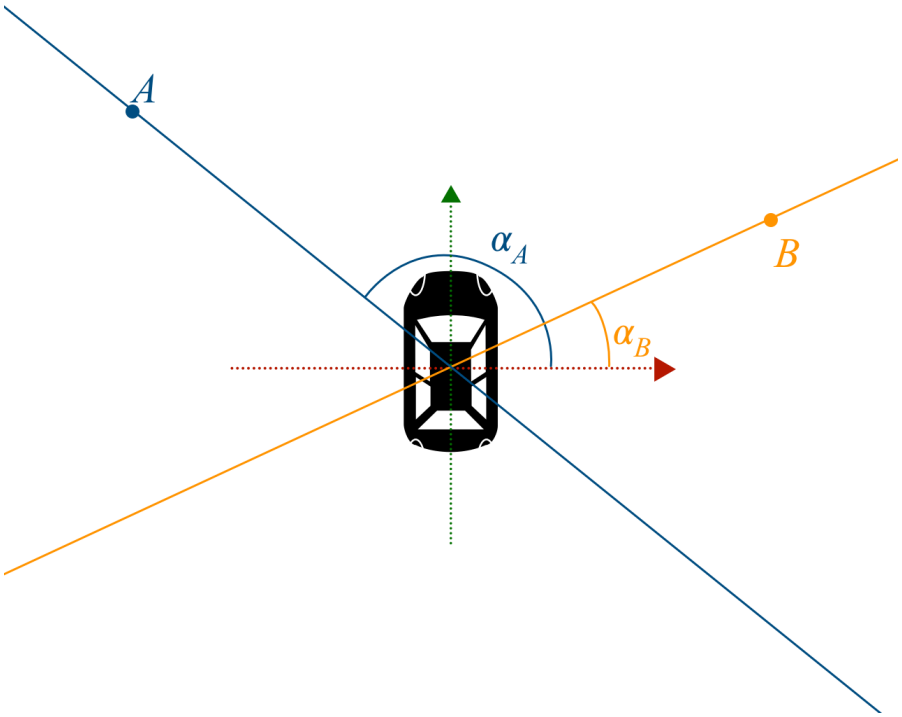


Figure 1.3: Illustration of the triangulation principle. A and B are users with a known location and are detected by the ego-vehicle. The angles of detection α_A and α_B form two lines intersecting at the position of the ego-vehicle.

and AOA systems. Nevertheless, triangulation-based approaches in the context of cooperative vehicle localization are still rare today.

1.3.1.3 *Geometric*

Compared to the two previous approaches, the geometric approach is one of the most direct methods. It consists of positioning the users in the local coordinate system of an observer (vehicle or infrastructure) having its global position known. To locate the user in the local coordinate system, several sensors can be used such as cameras, RADARs or LiDARs.

In particular, Einseider et al. have implemented an alternative positioning system for underground parking lots [35]. The detection and localization of vehicles are done via cameras placed at known positions. To estimate the position of vehicles in the fields of view of the cameras, the images are segmented into zones of 1 meter. The device set up by the authors allows detecting a vehicle at 20 m with a maximum error of 80cm. This methodology takes advantage of the geometric topology and the small distances of the scene but does not apply to larger baselines. To overcome this problem of scenes with large distances of the Providentia project, RADARs have been added to the cameras. This device allows the detection and localization of vehicles over distances up to 200 m with a longitudinal RMSE of 3.27 m and a lateral RMSE of 0.53 m [4].

With a smaller scene, Lv et al. proposed an approach based on LiDARs at the four corners of an intersection. Vehicles are identified

in point clouds by clustering points having a distance between them below a fixed threshold. The position extracted from this point cloud corresponds to the nearest point of the laser scanner. In [55], Héry et al. suggest a solution to extract the position of the vehicle from these point clouds. They distinguish two types of clusters, those shaped like an L common in lateral detection and those shaped like a C for longitudinal detection. In addition to these two types of clusters, two cooperation formulations are presented. The first one corresponds to the one where the ego-vehicle is equipped with sensors estimating the pose of a vehicle with a known position. The second one corresponds to the formulation where the ego vehicle has its position estimated by the other vehicle having its position known and being equipped with the sensors. Héry et al. observe that the second formulation, corresponding to the case where the ego-vehicle position is estimated by the vehicle knowing its position and being equipped with sensors to estimate the pose between the two vehicles, obtains better results than the first formulation. This lies in the case of infrastructures where the positions of the sensors are precisely known. Besides, L-shaped clusters provide, in both formulations, results with better accuracy and consistency, underlining the importance of the structural perception of the vehicle.

1.3.2 *High-level cooperative position estimation*

As we have shown, low-level-oriented approaches are much closer to the hardware. In the case of high-level approaches, they use estimates of already established positions as a basis for refining them. One of the most popular methods for position estimation applications is based on the Extended Kalman Filter (EKF). This approach has been chosen by Miller et al. in [81] to enrich the position estimate obtained by a GNSS system with position estimates from other vehicles and integrating these data through the use of an EKF with a resulting standard deviation of 0.02m.

However, despite their efficiency in terms of calculation cost, EKFs are only applicable to locally linear signals with noise following a Gaussian distribution. In other words, the error of position estimation must be contained in a Gaussian distribution and thus won't allow jumps (which can appear in urban canyoning conditions). Outside these conditions, they are no longer efficient and other methods such as particle filters are preferred. This is what Huang and Wu have chosen in [58] by proposing a cooperative framework based on this approach and the Interacting Multiple Model (IMM) adapted to cooperation. The authors simulate the use of the simple Particle Filter (PF) and obtain an RMS error of 0.2146 m/m traveled on the x-axis and 0.2135 m/m on the y-axis whereas with the IMM-PF filter they obtain 0.1249 m/m and 0.1193 m/m on the x-axis and y-axis respectively.

While Miller et al. [81] use an approach based on graph theory and in particular bipartite graphs to associate perceived vehicles with the one from the real world, Gulati et al. use bipartite graphs in the

form of factor graphs for localization [50, 51]. In [50], the authors present a formulation of the cooperative localization problem by setting constraints between vehicles according to their distance to correct measurement errors and obtain better results than those obtained using an EKF. The authors reiterate in [51] by integrating data from infrastructures and exceed the previous results.

The use of the high-level approach based on optimization and filtering methods brings several advantages. The first one is that this approach is compatible with low-level approaches. Indeed, low-level approaches give as output a position estimation whereas high-level approaches take as input position estimates to refine them. Consequently, the high-level approaches operate as a brick placed to improve those used for position estimation. However, this advantage of easy integration into an existing system underlines a major drawback: high-level approaches require basic components to obtain a first position estimation and therefore cannot be used alone. Another advantage of using this approach is that the processing and size of the data required are reduced significantly facilitating the communication between users. This is indeed the observation of Gulati et al. in [50, 51] via the use of factor graphs.

We could distinguish 3 methods mainly used: EKF, Particle Filter, and Graph-based methods. Thus we introduced an example of each to understand the available methods with their advantages as well as their limitations. However, Gao et al. gather a lot more of these methods in a cooperative context in their book [43] diving into mathematical details as well as the diversity of variations of each method which is beyond the scope of this thesis.

1.3.3 *Conclusion*

In Table 1.4, we note that several methods solely offer a better precision compared to the non-cooperative methods. Generally, cooperative localization optimizes the output of the standalone position estimation methods, refining the estimation through extra data usage. However, a poor quality localization ability of an agent might dramatically affect the overall results. It also offers an alternative source of localization for GNSS denied environments, especially from well-located measure points such as infrastructures.

Paper	Category	Methodology	Metrics	Results	Experiment	Style	Notes
[91]	Multilateration (Low-Level)	GPS Multilateration	Error	3.3 m to 6.75 m	Simulation	V2V	
[2]	Multilateration (Low-Level)	GPS weighted Multilateration (based on distance)	Average error	2.38 m	Simulation	V2V	With a fleet of 10 vehicles.
[3]	Multilateration (Low-Level)	GPS weighted Multilateration (based on SNR)	Error	85 cm to 25 cm	Simulation	V2V	With a fleet of 20 vehicles and another of 200.
[86]	Multilateration (Low-Level)	TDOA with 5G antennas	Error	20 cm to 25 cm	Simulation	V2I	
[35]	Geometric (Low-Level)	Image segmentation	Error	80 cm	Experimental	V2I	At 20 m
[55]	Geometric (Low-Level)	Sensor fusion known $\rightarrow$ unknown	Mean error	x: 11 cm, y: 36 cm, h: 39 cm	Experimental	V2V	

[55]	Geometric (Low-Level)	Sensor fusion unknown → known	Mean error	x: 27 cm, y: 116 cm, h: 124 cm	Experimental	V2V	
[58]	Optimisation (High-Level)	Particle Filter	RMSE	x: 0.2146, y: 0.2135 m/m traveled	Simulation	V2V	
[58]	Optimisation (High-Level)	IMM-PF	RMSE	x: 0.1249, y: 0.1193 m/m traveled	Simulation	V2V	
[81]	Optimisation (High-Level)	EKF based optimisation	Standard deviation	0.02 m	Both	V2V	
[50]	Optimisation (High-Level)	Factor Graph	combined RMSE	See original publication for graph	Simulation	V2I	Improvement compared to EKF

[51]	Optimisation (High-Level)	Factor Graph	decrease RMSE	10.54 %	Simulation	V2I	Compared to EKF with 4 vehicles for 1000 iterations

Table 1.4: Recapitulative table of the reviewed localization works.

1.4 OBJECT DETECTION AND TRACKING

To navigate in an environment, it is necessary to be able to detect obstacles in the scene and to track them. Today, most approaches are based on non-cooperative detection algorithms. This is mainly due to the limitations of communication methods. In this section, we will review the different approaches to perform detection in a cooperative context and the available tracking methods.

1.4.1 *Detection and classification*

The first step before classifying objects is the extraction of areas of interest from the data produced by the sensors. Here, we want to isolate the mobile objects from the background. This is what Lv et al. do in [77] where after subtracting the background, group the remaining points into clusters. These clusters are delimited by batches of points having a distance to each other below a threshold set beforehand. Another strategy was adopted by Chen et al. in [23] where the shared data correspond to areas of interest depending on the position of the vehicles such as the part of the scene scanned by two vehicles. The more precise extraction of objects is performed during the detection phase.

The trend of point cloud raw data sharing is very recent. This is because communication between users has been limited for a long time. For instance, the majority of cooperative systems perform the detection and classification of objects in a scene locally. The extracted data is often enriched before being shared. A typical example is the Providentia project [4] where cameras provide a video stream sent into a neural network based on the You Only Look Once Version 3 (YOLOv3) architecture to detect vehicles in the images. The data of the vehicles thus classified are enriched thanks to RADAR sensors allowing a better estimation of their position in the scene. Similarly, Lv et al. [77] based their solution on the same concept where the user's characteristics are locally extracted and where the classification, using the random forest algorithm, is done locally. The corresponding data are then centralized to facilitate user tracking.

Nowadays, the majority of detection and classification methods are based on algorithms based on a neural network-oriented architecture. Grigorescu et al. in [49] and Arnold et al. in [6] provides an overview of methods used to detect and classify other users in a non-cooperative manner. Although the details of these methods are beyond the scope of this thesis, Arnold et al. offer a review of data fusion methods, thus providing insight into the problem of multi-modality and the management of several streams. Based on the work of Chen et al. [27], the authors raise 3 fusion schemes :

EARLY FUSION:
: The data streams are merged and formatted before passing through the neural network. As an example, color data can be

added to point clouds from cameras. The disadvantage of this solution is that it is not robust to stream failure.

LATE FUSION:
This is the classic scheme we have seen: the data are processed locally and separately for each modality and then the results are merged only at the end. Although it does not benefit as much from the cooperation in terms of classification, it offers the best performance.

MID FUSION:
Also named as deep fusion, the raw data are sent to the neural network, which will handle the association of the data by itself. Although it is sensitive to the absence of modality, it takes full advantage of cooperation and offers better results than the previous methods. It is with this scheme that the work of Chen et al. [23] can be associated.

These three approaches were initially formulated for the local processing on the vehicle but can easily be extrapolated into a cooperative context. Therefore, we can associate these three strategies to the notion of a stream that can contain point clouds, images, or the characteristics of the detected users from any source. However, this extrapolation has a cost in terms of complexity because the sensors have to be calibrated dynamically from each other. Since the systems are independent of each other, the extrinsic parameters between the sensors are composed of translation, rotation, and time-shift parameters.

The authors of [23] however proposed an extrapolation of the deep fusion scheme in [22] in which the raw data from a laser scanner start being processed in a neural network. The authors tried using the feature at a different level: the voxel feature level and the spatial feature level. The first one shares a 3D grid containing the result of the VoxelNET neural network while the other one shares a higher-level feature from the fusion of spatial features maps. While the first one generates a large amount of data, the spatial feature level is sparser, thus lighter, facilitating the exchanges in a bandwidth-limited environment. Similarly, Marvasti et al. in [78] propose a method to share deep features from an intermediary layer of a neural network. However, such an approach brings the question of standardization of the perception pipeline among every user especially on the evolution of the neural network in charge of detection as well as the diversity of models from the different suppliers.

1.4.2 *Tracking*

The aim of tracking users is to follow them as long as possible in the scene. Several methods are available to tackle tracking tasks, enumerated in [29] by Datondji et al., such as region-based, contour-based, feature-based, or model-based methods. Datondji et al. also list two types of tracking algorithms: matching-based and Bayesian-based algorithms. However, this can be a challenging task because of several

parameters such as occlusions, change of perception (e.g. appearance, distortion, etc.), or environment changes (e.g. lighting, color change, weather change, etc.). Cooperation brings an answer to these difficulties by bringing various points of view.

In [113], authors underline that localization tasks and tracking tasks can be bounded in Simultaneous Localization And Tracking (SLAT). Authors propose to use a localization method based on the footprints of radio transceivers based on the Omnipresent Signals of Opportunity (SOOP) method. Connected vehicles seek targets by exterminating the radio reflection of illuminated targets. They propose a SLAT method based on the derivation of Fisher Information Matrix (FIM) to locate users and use a hybrid distributed algorithm based on Belief Propagation (BP) to track them and obtain better results than EKF based methods. The tracking method used is based on the region matching method. Similarly, Miller et al. used in [81] a region matching method based on bipartite graphs to track vehicles in a V2V context.

In Providentia project [4, 56], authors based their tracking methods using Gaussian Mixture Probability Hypothesis Density (GMPHD). Similarly, Chen et al. in [25] used a GMPHD based method to extract the tracks of multiple vehicles. The authors perform a SLAT using a Bayes inference-based algorithm optimizing relative pose estimation and fusing the matched tracks using fast covariance intersection based on information theory (IT-FCI). These methods are based on region methods alongside Bayesian-based algorithms. An answer to the resolution of complex scenes is provided by Huang and Wu in [58]. The authors rely on cooperation and on a method using particle filters to locate vehicles more precisely, thus reducing ambiguities when the vehicles are very close to each other.

Kim et al. in [63, 64] uses the speed of the vehicles as a feature to identify the user and to track them, thus performing a feature-based tracking with a matching algorithm. Lv et al. in [77] used the corner of the detected car the closest to the sensor and applied the Global Nearest Neighbor (GNN) [15] method to track the vehicles. This approach lies in the use of a contour-based method with a matching algorithm.

In [9], authors propose a set of metrics available for tracking tasks performance evaluation which is nowadays frequently used. However, in a cooperative context, we have not found works that bring a quantitative evaluation of their tracking methods. This is mainly caused by the fact that, in cooperative works, a tracking task is just a tool but not at the center of the research efforts.

1.4.3 *Conclusion*

The multiplication of the points of view offers a significant advantage to overcome the limitations of the sensors or to reduce the effects of the changes of the scene condition. In Table 1.5, we provide a summary of the solutions given for user detection. Tracking on the other hand seems to be put aside since the cooperative tracking methods used

are often only a means to obtain other results on other parts of the perception pipeline. We also observe that the field of detection and tracking in a cooperative domain benefits from very little research effort. We believe that this lack of experimentation in a cooperative context is due to the bandwidth requirements in communication as well as the sensitivity to desynchronization and pose estimation errors.

Paper	Category	Methodology	Metrics	Results	Experiment	Style	Notes
[77]	Raw data based detection	Random Forest	Detection Rate	95.5 %	Experimental	V2I	No data given for the tracking performances
[22, 23]	Raw data fusion based detection	CNN based network	Average Precision	Near detection: 77.46 %, far detection 71.42 %	Experimental (Datasets)	V2V	With KITTI
[22]	Voxel feature fusion detection	CNN based network	Average Precision	Near detection: 77.46 %, far detection 58.27 %	Experimental (Datasets)	V2V	With KITTI
[22]	Spatial feature fusion detection	CNN based network	Average Precision	Near detection: 50 %, far detection 57.14 %	Experimental (Datasets)	V2V	With KITTI

[78]	Deep feature fusion detection	CNN based network	Detection of undetected vehicle	Up to 30 %	Simulation (CARLA)	V2V	Detection of undetected vehicle by non cooperative algorithm
[35]	Geometric Tracking	Image segmentation	Error	80 cm	Experimental	V2I	At 20 m
[4]	Geometric Tracking	Sensor fusion	RMSE	lat: 3.27 m lon: 0.53 m	Experimental	V2I	At up to 200 m

Table 1.5: Recapitulative table of the reviewed cooperative detection and tracking works.

1.5 MAP GENERATION

In the previous sections, we have reviewed the ego-localization methods as well as the detection and tracking methods of agents in a given scene. Ego-localization and detected and tracked objects can be merged in a map. Thus, the map can be built cooperatively by aggregating information from multiple agents. Nowadays, commercial solutions are available to bring a cooperative aspect to the maps available in navigation aids. This is notably the case for crowdsourcing-based solutions such as TomTom, HERE, Waze, etc.

Hence, it is clear that the goal of cooperative mapping used today is to optimize routes and adapt vehicle navigation by anticipating the different events on the user's route. These objectives can be taken further, in particular, to predict trajectories in real-time thanks to lower latency and better accuracy of shared data.

In this section, we review the use of maps in a cooperative context and the different formats available.

1.5.1 *Geometric maps*

Geometric maps are made up of vector elements describing the environment. This method is used in applications such as OpenStreetMaps. However, in a cooperative context, data from services like the one mentioned above are not precise enough, which has led to the creation of maps with better accuracy. In [11], the authors present Enhanced Maps (Emap) that provide lane level accuracy maps. To achieve this goal, Bétaille et al. propose to add a set of circles and clothoids to the traditional vertices. Also in view to improve map accuracy, Bender et al. present in [8] the lanelets. The lanelets take the form of vertices representing the left and right sides of a traffic lane. These vertices also have an enhanced topological role by representing the links between places and the distance between them.

The use of geometrical maps in a cooperative application has a supporting role in which the information shared between users is integrated. Xu et al., in their review of their participation in the 2016 GCDC [119], had to recreate a high-definition map to enrich the OpenStreetMap plots before using it. Thanks to these high-definition maps, it has become possible to precisely place elements in real-time such as other users or danger zones to be avoided and thus to navigate in a context of cooperative driving in several scenarios that we will present later. Similarly, in the Providentia project, the autobahn section has been modeled beforehand with great precision, creating a digital twin of the scene [4]. Here, the infrastructure shares the position of each of the detected vehicles to generate a dynamic map. Finally, the team of Lv et al. [77] didn't use maps but has rather relied on sharing information in real-time that can be used to enrich geometric maps such as the position of vehicles, pedestrians or even information on the status of traffic lights.

Through these applications, a global pattern emerges: the shared information is used to enrich the map rather than to modify it in depth. Cooperative geometric maps are therefore made up of a succession of layers. The base layer represents the terrain and is almost invariable. It can be created from national institutes or directly extracted from sensors. Then, the higher the layer level is, the shorter the lifespan of the elements of this layer is. This layer organization has been formalized under the name Local Dynamic Maps (LDM) by European Telecommunications Standards Institute (ETSI) [39] and takes the format of layers with varying validity periods and offers an implementation framework. The LDM is thus defined as 4 layers :

- Type 1: Static data (Roads, applied speeds, infrastructuresetc.)
- Type 2: Long term transient data (Work zone, temporary speed change)
- Type 3: Medium-term transient data (weather situation, parked vehicles, traffic jams, etc.)
- Type 4: short term transient data (vehicles on the road, traffic lights, etc.)

Each layer is updated with a frequency depending on the duration of validity of the information. Typically, the Type 4 Layer is updated in real-time.

1.5.2 *Volumetric maps*

Volumetric maps are, unlike geometric maps, atomic elements representing the presence or absence of an obstacle that form a grid with squares contiguous to each other or scattered arbitrarily. The advantage of volumetric maps lies in the fact that they can be easily created from sensor data and therefore represent the immediate environment at the time of data acquisition. Occupancy grids fall into volumetric maps category forming a 2-dimensional grid, or matrix, similar to an image [96]. Indeed, a greyscale image can be taken where each pixel corresponds to an area of the environment and where the greyscale represents the probability that the area corresponding to the pixel contains an obstacle as illustrated in Fig. 1.4.

These maps have the advantage that they can be combined very easily. The authors of [14] have thus shown that they were able to associate the maps of several robots to obtain a complete map of the environment. The goal of associating them is to find the transformation matrix between perception systems. In the case of 2D occupation grids, the transformation matrix $T_{x,y,\theta}$ contains three parameters: translation on the x-axis, translation on the y-axis and rotation by an angle θ. Hence the authors seek a matrix $T_{x,y,\theta}$ that maximizes the similarity between two overlapping maps also called a point registration algorithm.

Kim et al. propose in [63, 64] to enrich their map by taking pictures with cameras positioned on several vehicles. The images captured in

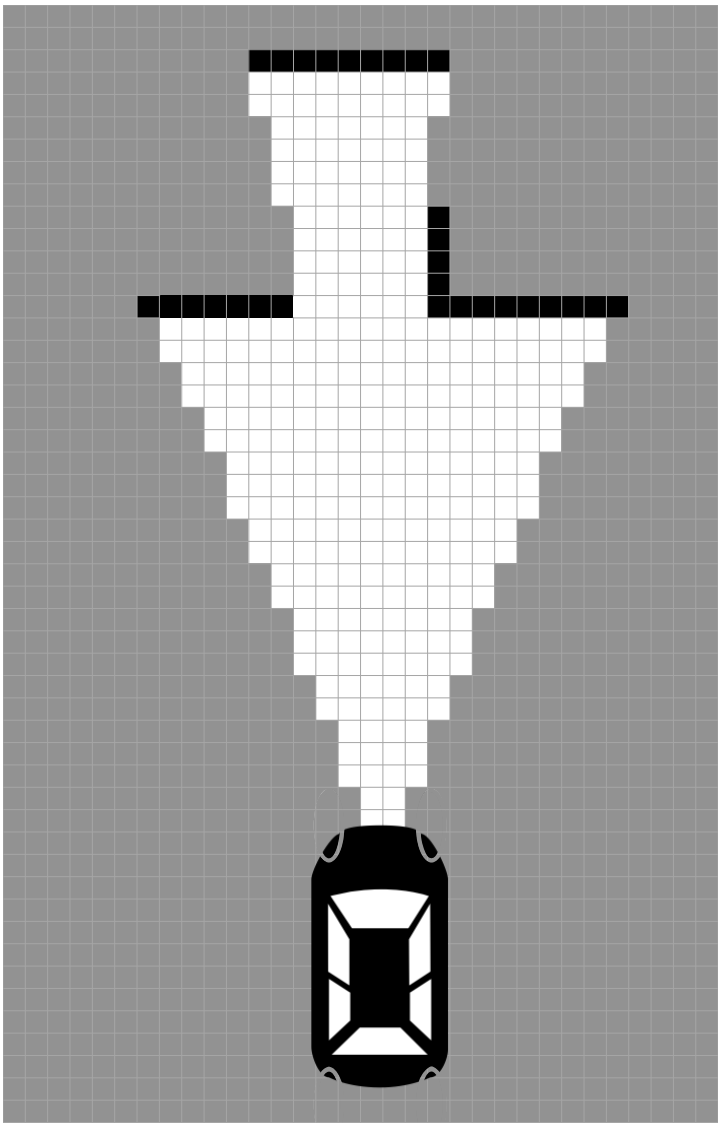


Figure 1.4: Occupancy grid example. Grey boxes represent a 50 % probability of occupancy if the area is unknown. The white boxes correspond to the zones identified as free and the black boxes correspond to the zones occupied by an obstacle.

this way are distorted to be laid on the ground, providing a satellite view of the scene. To obtain this result, they applied the Inverse Perspective Mapping (IPM) method. When the camera acquires an image, the scene is projected onto the sensor plane. TheIPM is based on the inverse principle: the 2D points of the sensor plane (stored in an image) are projected back into a 3D space, assuming that each of the points is on a flat surface (e.g. the road). The authors of [63, 64] set the plane to $Z = 0$ and used other sensors to remove the points that do not belong to this plane. Thus, by knowing the position of the different vehicles and, by extension, the position of the cameras, it is possible to obtain a map enriched with a satellite view cooperatively.

Although this type of map has the advantage of being simple to use and share, it has the disadvantage of becoming heavier with the size of the environment being explored invariably, whether the areas are interesting or not. To overcome this problem, the notion of quadtree can be introduced. The quadtree divides the map into coarse blocks which, if they contain useful details, can be subdivided into sub-blocks which, in the same way, can be divided into sub-sub-blocks.

The authors of [59] used the quadtree-based method to store a grid of occupancy generated by LiDAR type sensors. Although they note that the method is more computationally intensive, it shows its advantage by dividing up to 10.9 the storage required for an equal area and accuracy. However, to the best of our knowledge, there are no methods for merging maps in quadtree format.

Until now, we have mainly been talking about two-dimensional maps, both geometric and volumetric maps. However, three-dimensional maps are becoming more and more popular thanks to sensors that provide information in three dimensions rather than just on one plane. 3D maps play an active role in navigation, especially in complex environments [80], and provide additional elements that make it easier to combine several maps.

Similarly, the 2D occupancy grids are also available in a 3D version consisting of voxels (volumetric elements). However, just as 2D maps tend to be too large, 3D maps are even more affected by this problem due to the additional axis. The answer to this problem is similar to that of two-axis maps: the octree. Hornung et al. present in [57] the OctoMap framework allowing the management of maps and their updates based on a probabilistic approach. Unlike the quadtree maps, the octree maps have benefited from a better interest in the context of cooperation. We can notably mention Drwiega's work in [34] proposing a method for associating several maps in the Octree format. To estimate the transformation matrix between the respective coordinate reference of the two maps, the author translates the Octree map into a point cloud and then applies the Iterative Closest Point (ICP) algorithm to it.

This brings us to maps based on point clouds. The volumetric maps we have seen so far represent the first steps in navigation in the context of mobile robotics that can be cooperative. However, in the context of the autonomous vehicle, point cloud-based maps are more widespread. Point cloud-based maps have the advantage of representing each im-

pact (and therefore obstacle) in Cartesian coordinates as well as the raw data from laser scanners like sensors [102]. These maps contain both fixed elements (the background, Type 1 in the LDM reference frame) and highly dynamic elements (Type 4 in the LDM reference frame). As a result, the static part of the map is occluded by the dynamic elements of the scene. One solution to reduce the impact of occlusion is cooperation, where the map can be generated by several sensors offering several points of view. This is notably what Bosch's teams chose in the MEC-view project [16, 42] where a set of cameras and LiDARs were placed on lampposts to generate an High Definition (HD) map and offer a view free of blind spots to autonomous vehicles updating in real-time. In [77], Lv et al. proposed a solution to extract the background from the raw scans by aggregating several frames and then applying thresholding to the voxels resulting from the rasterization of the accumulated point clouds.

In the same way, as for occupation grids (2D or 3D), the key point allowing the cooperation and thus the association of point cloud maps is the estimation of the transformation matrix between the respective referential of each point cloud. As explained by Yang et al. in [121, 122], the point set registration algorithms are particularly suitable for this task. Indeed, their objective is to find the transformation matrix minimizing the distances between a set of points located on overlapping acquisition parts. Note that sensors, and thus point clouds, by convention, are measured in metric systems which implies that scaling is generally not necessary (if it is required, it would be specified by the manufacturer). Thus, the desired transformation is then qualified as rigid in which the transformation matrix is composed only of the translation matrix and the rotation matrix. The most popular algorithm in mobile robotics is the ICP algorithm that looks for the minimum distance between corresponding points in the two-point clouds by using the method of least squares. However, this method is particularly sensitive to outliers. Another challenge appears with the lengthening of the baseline which is the increase of the disparity of the points. To overcome this problem, Wu et al. propose in [116] a semi-automatic solution to merge sparse point clouds called PA-ICP. PA-ICP is based on the recognition of corners which must be paired with their corresponding corners in every point cloud. Finally, in the context of the autonomous vehicle, it is vital to know the confidence index of the generated map and thus the quality of the point cloud association. Yang et al. propose in [122] TEASER, a point set registration algorithm capable of indicating its confidence index and being robust to outliers.

As we have written, maps based on point clouds contain both static and dynamic elements. The dynamic elements can therefore be extracted from the latter to be processed to recognize their role in the scene and track them if necessary.

1.5.3 *Conclusion*

To conclude this section about mapping, we can observe that cooperative mapping serves the enrichment of the context in which vehicles moves. Thanks to the larger memory available on the infrastructure it is possible to store and thus share heavy HD maps. As we will see in the next section, the multiplication of the point of view reduces the occlusions and improve the reliability of the detection and tracking. These detected objects can be placed on the map following the LDM model and then shared with the connected vehicles to help the trajectory planning stage.

Strenghts:	Weaknesses:
• More precise localisation in environment with GNSS • Localisation possible in GNSS denied environment • No drift in Localisation • V2X Mapping • Better reliability • Cost reduction • Less occlusion • Real-Time update (solely depending on the transfer latency and computation time) • Larger field of view • Detection of unconnected User	• Similar precision of pose estimation with non cooperative system • Dependent to the number of users • Computation expensive • High throughput required • Latency

Opportunities:	Threats:
• Raw sensor data fusion • Various point of view of the scene • V2I Map generation • V2I Object management • Infrastructure always available and calibrated • Further Trajectory planning • Anticipation of dangers • Infrastructure offers more storage and can delete duplicate parts allowing storing HD maps • Better Bird Eyes View map creation • Existing matching methods	• Higher cost for the infrastructure • Lack of normalisation between constructors • Consistency of the accuracy of the pose estimation between the sensors • Detection and classification accuracy of each participant • Synchronisation between participants • Data association of a single object with a very different point of view. • Missing stream or data management • Data management between mobile and fixed users

Table 1.6: Cooperative Perception - SWOT

1.6 REVIEW AND SUMMARY

In the previous sections, we have reviewed the three main blocks of the perception pipeline in a cooperative context: localization, mapping, and object detection and tracking. In addition to this, we reviewed the architectures available for cooperative systems along with their advantages and drawbacks. We also observed the challenges brought by cooperative solutions as well as the available network facilities. This information allows us to establish a SWOT and thus obtain a clear view of the state of the art of cooperative perception and more particularly of those using an infrastructure. This SWOT is available in Table 1.6.

Through these sections, we have also reviewed several solutions that make use of cooperation for certain blocks of perception. For the sake of clarity, Table 1.7 provides a review of them.

1.7 COOPERATIVE PERCEPTION IN REAL LIFE

So far, we have reviewed the available data to perform perception, the methods to share them as well as the different approaches and challenges related to cooperation. We also reviewed three main tasks of perception using cooperation namely, ego-localization, detection and tracking, and, finally, map generation. This section aims to assess the scenarios in which cooperative perception proposes a significant impact as well as the related experimentations. We will close this section with a presentation of datasets.

1.7.1 *scenarios & Experiments*

The cooperative perception responds to safety issues and more specifically those related to the lack of visibility in blind spots. This lack of visibility can be caused by the structure of the scene or by other users. We can take the example of pedestrians wanting to cross the road but being hidden by parked vehicles or even vehicles appearing in an intersection and being hidden by buildings. It is on this last example that the point cloud sharing project of Li et al. is based [73]. The authors' work focuses on the SDN network structure for connected vehicles as well as the use of mmWAVE wireless communication links offering higher data rates than networks in the 2.4 GHz frequency bands. In this network, there are several infrastructures equipped with laser scanners that allow the visualization of areas hidden by buildings thanks to the fusion of LiDAR point clouds covering the trajectory of the connected vehicle. In this way, they can reduce the effects of blind spots and detect other users that were previously undetectable.

Li et al. also addressed the overtaking scenario in which it can sometimes be difficult to know whether a vehicle is coming into the opposite lane since the view is occluded by the vehicle we wish to overtake. This is also one of the scenarios that motivated the work of Kim et al. in [64]. In this paper, the authors use cameras placed on

several vehicles to create a see-through visualization system. To merge the images, the authors project these pixels onto the ground to create a birds-eye view map. This map can then be back-projected according to a camera model to visualize what is behind the vehicle.

The 2016 edition of the GCDC was an opportunity to explore several other scenarios as well as challenging several teams. In this case, Xu et al. [119] presented these scenarios and their comments about their experience. Three scenarios are presented:

ZIPPER MERGE:
: This case corresponds more generally to the insertion of a vehicle into a lane and is encountered in several situations such as when a traffic lane becomes inaccessible (e.g. for maintenance).

CROSSING AT INTERSECTION:
: Here, a vehicle wants to cross an intersection with as little disturbance to the traffic situation as possible.

EMERGENCY VEHICLE YIELDING:
: This situation corresponds to the arrival of an emergency vehicle and which therefore has the priority. The vehicles on the scene must leave a passageway between the traffic lanes to allow them to circulate.

During the experimentation phase, the vehicles transmitted their status (position, speed, wheel angle, etc.) and could make requests involving a change in vehicle behavior. For example, when inserting into a lane, the vehicle behind changes its speed to leave sufficient space for the requesting vehicle to change lanes. During this challenge, the vehicles do not cooperate on the perception axis but rather on the vehicle control axis. However, the authors note in their remarks the weaknesses of the perception implemented caused by the lack of a multi-sensor based perception methods.

The project Proviendentia [4] aims to bring cooperative perception to the motorways. This project is composed of cameras and radars placed on gantry bridges on a section of the motorway. Vehicles are detected and classified using machine learning algorithms, and their positions are estimated using the data provided by the radars. A digital twin of the road section is created from this data and is accessible in real-time.

MEC-View is a similar project implemented by Bosch [16, 42] where LiDARs and cameras are placed on lampposts at an intersection to cover blind spots caused by other vehicles. Similarly, Lv et al. in [77] equipped an intersection with 4 LiDARs sensors to track vehicles and detect obstacles to inform users. The problem of intersections is particularly extrapolated to roundabouts, which are more frequent on the European continent.

Another issue, raised by Kim et al. [64] as a limitation to their system resides in the roads forming parabola peaks. Under these conditions, the topology of the terrain reduces the driver's field of view to the sensors.

Figure 1.5: Synchronous video frames from each camera of our multi-agent dataset made with CARLA.

1.7.2 *Datasets*

The increasing interest in the cooperative vehicle initiated the sharing of some datasets. However, they tackle specific contexts such as communication or infrastructure perception.

KO-PER [103] This dataset proposes a context of a cooperative infrastructure. It is made of sequences monitoring an intersection with 14 laser scanners (4 for the road, 2 for the sidewalk, and 8 for the egresses) and 8 monochromatic cameras (only two are available in the dataset due to personal data protection purposes). Laser scanners are synchronized and operate at 12.5Hz while the cameras operate at 25Hz in phase with the laser scanners. Raw data from the scanners and undistorted images from the cameras are available alongside reference data of selected vehicles and object labels.

WARRINGAL [115] The authors propose a dataset gathering communication interaction between vehicles of a fleet of 13 elements for 3 years. The data proposed are the state of the vehicle, the list of each communication and their length, the signal strength of each communication (e.g. RSSI or antenna used by each vehicle), and the map.

T&J [23] This dataset has been created to complement KITTI's dataset [46] by adding a cooperative dimension. For the learning and evaluation phase of their Sparse Point-cloud Object Detection (SPOD) algorithm, the authors needed a dataset offering overlapping acquisitions from several points of view. The latter is composed of images from multiple cameras, radar data as well as point clouds from LiDARs. As with the KITTI dataset, this data is linked to a GPS and an IMU but offers simultaneous views from different positions.

A LACK OF COOPERATION We could have presented other datasets such as KITTI's or more recently the Waymo Open Dataset or INTERACTION by Zhan et al. [124]. However, we can note the absence of cooperation and dataset representing the scenarios we presented despite the interest and the projects responding to its problems. We can also note that the datasets presented deal either only with infrastructure or V2V cooperation. However, simulators can bring an answer to this lack by allowing the acquisition of data from several points of view synchronously. Moreover, they solve the problem of the ground truth definition as well as the calibration challenges. CAR Learning to Act (CARLA) [33] is one of them providing several sensors such as cameras, depth cameras, LiDAR (simulated ray cast), IMU and RADAR. In July 2018, version 0.9.0 introduced the multi-client multi-agent support offering cooperative vehicles perspective. Fig. 1.5 showcases the possibilities offered by CARLA with synchronized image acquisition from vehicles and infrastructure at a round-about. Other simulators are available such as Deepdrive [89], LGSVL Simulator [92] or AirSim [98]. However, CARLA remains the most popular nowadays.

Ref	Type	Tasks	Sensors	Communic.	Architect.	Method	Comment
[77]	V2I	Localization, Mapping, Classification & Tracking	LiDAR	DSRC, SPaT, BSM	Centralized	Geometric Relative Localization, Background filtered, PC cluster, Random Forest	The lanes are detected by aggregating vehicles paths and vehicles are tracked with the closest point of their corresponding cluster. The infrastructure does not merge the Point Clouds and transmit the information to the users via Bluetooth. The pose of the vehicle is determined in relative coordinates and converted to absolute coordinates.
[23]	V2V	Detection & classification	LiDAR	DSRC (ROI)	Distributed	CNN	SPOD is based on CNN. A dataset has been created.
[119]	V2X	Localization	LiDAR, GNSS-RTK	ETSI C-ITS (CAM, DENM, iCLCM)	Distributed	Control	Absolute coordinates are transmitted by messages by each vehicle.
[73]	V2I	Mapping	LiDAR	mmWave, ROS	Mixed	LDM	Share Point clouds through mmWave links.

Ref	Type	Tasks	Sensors	Communic.	Architect.	Method	Comment
[4, 48]	V2I	Detection, classification & tracking, Localisation	Camera, radar	4G, 5G, Optic Fibre	Centralised	YOLOv3, Tracking via radar, GM-PHD	The radar help to determine the position of the users in the absolutes coordinates.
[63]	V2V	Mapping, vehicle matching	Odometry, LiDAR, camera, DGPS	IEEE802,11n (WiFi)	Distributed	IPM	RAW data are shared between vehicles for mapping. Feature-based object matching (speed of the vehicles). Maps are merged using the coordinates given in the messages.
[64]	V2V	Tracking, Mapping	Odometry, LiDAR, camera, DGPS	IEEE 802,11gn (WiFi), 3G, 4G, ROS	Distributed	Mapping: IPM, ICP, CSM	The position of tracked users are given into relative to the ego-vehicle coordinates.
[110]	V2X	Tracking	Camera, LiDAR	IEEE 802.11bgn (WiFi), ROS	Distributed	GM-PHD Filter, EKF, Sequential Monte Carlo	The relative poses are estimated

Ref	Type	Tasks	Sensors	Communic.	Architect.	Method	Comment
[50, 51]	V2I	Localisation	Range detector, Odometry, GPS (all simulated)	Simulated	Distributed	Factor graph (High Level)	The absolute positions are directly processed.
[35]	V2I	Tracking	Camera	IEEE 802.11g (WiFi), IEEE 802.11p (WAVE) with DENM messages	Centralised	Geometric (Low level)	The map and position of the user are transmitted from the infrastructure. The position is given in absolute coordinates of the car park space.
[55]	V2V	Localisation	LiDAR, GNSS RTK	Not given	Distributed	Geometric (low level)	The relative pose is extracted from the LiDAR's data and is used to compute the absolute pose.
[91]	V2V	Localisation	GPS, Range sensor	Not given	Distributed	Bayesian (High level)	The estimated position is given in absolute coordinates.

Ref	Type	Tasks	Sensors	Communic.	Architect.	Method	Comment
[81]	V2V	Localisation, Tracking	GPS RTK, camera, radar, LiDAR	DSRC	Distributed	EKF, Bipartite graphs (High level)	The localization is given in absolute coordinates. The bipartite graphs are used to match users to the detected ones.
[16, 42]	V2I	Localization, detection and tracking	Camera, LiDAR	4G, 5G	Centralised	Not given	
[26]	V2V	Localization and tracking	Radar	DSRC	Distributed	GMPHD	The estimated position is given in absolute coordinates.

Table 1.7: Summary of the experimentation and methods reviewed along the chapter underlining their conditions of realization, the methods used and their results.

1.8 CONCLUSION AND PERSPECTIVES

This chapter was an opportunity for us to review the different stages of a perception pipeline under a cooperative context and its associated challenges.

A large amount of work tackling the localization problem has been accomplished. We reviewed a wide range of solutions introducing different paradigms, improving the pose estimation, or offering an alternative reference point in a GNSS denied environment. We also noted that the cooperative localization topic is very active as witnesses the amount of recent literature. However, we perceived a strong contrast concerning the available literature amount on cooperative detection and tracking. Even if some projects employ local perception systems merging the detected user's information, the topic remains sparse in raw data sharing.

We also reviewed the usage of maps in a cooperative context which, similarly to localization, is currently an active topic. In this field, cooperation also makes it possible to overcome the limits of the distance of sensors, allowing better anticipation of trajectories and possible adjustments.

More generally, we witnessed a difference in the cooperative scheme between V2V and V2I architecture. In V2V, vehicles communicate evenly with each other whereas, in V2I, the privileged approach is unidirectional from the infrastructure to the connected agents. We believe that bidirectional cooperation could be beneficial in bringing an "in the scene" point of view, thus adding details helping to understand the scene. This bidirectional scheme may provide new opportunities for dynamic calibration, reinforcement learning [75] or as an arbitrator in case some agents share erroneous data.

Finally, although cooperative perception is currently an active topic, we noted the absence of datasets featuring multiple points of view, from different actors, in a scene. These datasets are a real key point in cooperative perception since they are mandatory to bring novel cooperative solutions. However, their creation requires solving the abovementioned challenges such as calibration.

In the next chapter, we present a bidirectional approach exploiting the "in the scene" Point of View (PoV) of the vehicles as well as the elevated PoV of the infrastructure. We also provide an answer to the lack of dataset that we observed.

2

MULTI-AGENT COOPERATIVE CAMERA-BASED EVIDENTIAL OCCUPANCY GRID GENERATION

ASSOCIATED ARTICLE

[1] Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Yohan Dupuis, and Rémi Boutteau. "Multi-Agent Cooperative Camera-Based Evidential Occupancy Grid Generation." In: (2022), pp. 203–209. DOI: 10.1109/ITSC55140.2022.9921855.

2.1 INTRODUCTION

In the previous chapter, we have noticed that two approaches for cooperative perception benefit from the majority of the research effort. The first approach corresponds to Vehicle-to-Vehicle (V2V) cooperation where we take advantage of the sensors embedded in the vehicles to perceive the scene in which the users evolve. However, this approach is highly dependent on the number of users with onboard sensors but requires almost no additional cost to establish cooperation. The second approach corresponds to the Vehicle-to-Infrastructure (V2I) cooperation where an infrastructure provides sensors to increase the range of sensing of the users in the monitored scene. This approach has the advantage of not depending on the number of users equipped with sensors and of having more omniscient points of view than those of the vehicles, but requires a non-negligible cost for the maintenance of expensive sensors that are prone to bad weather.

In this chapter, we explore a new approach: providing user's resources to use their in-scene PoV alongside with the elevated PoV of the infrastructure. The objective is to build a map, shared to all users (contributor or not) while reducing the impact of sensor limitations and occlusions due to terrain or other vehicles.

GENERAL IDEA Today, many vehicles already have cameras on board, usually accompanied by a processing unit. It is the same for the infrastructures which can be based on a mesh of cameras of surveillance already in place or having solutions on the shelf. Indeed, cameras are nowadays cheap sensors and offer a wide range of possibilities. It is because they are so common today that the work of this thesis uses exclusively cameras to generate our map.

The behavior of cameras can be modeled by a simplified mathematical model called the pinhole camera. This model allows to project objects from the 3-Dimensional (3D) world space into the 2-Dimensional (2D) space of the image plane. However, when we look-up from 3D points to 2D points, we lose a dimension: the distance of the 3D point from the camera. The consequence is that, when we want to do the

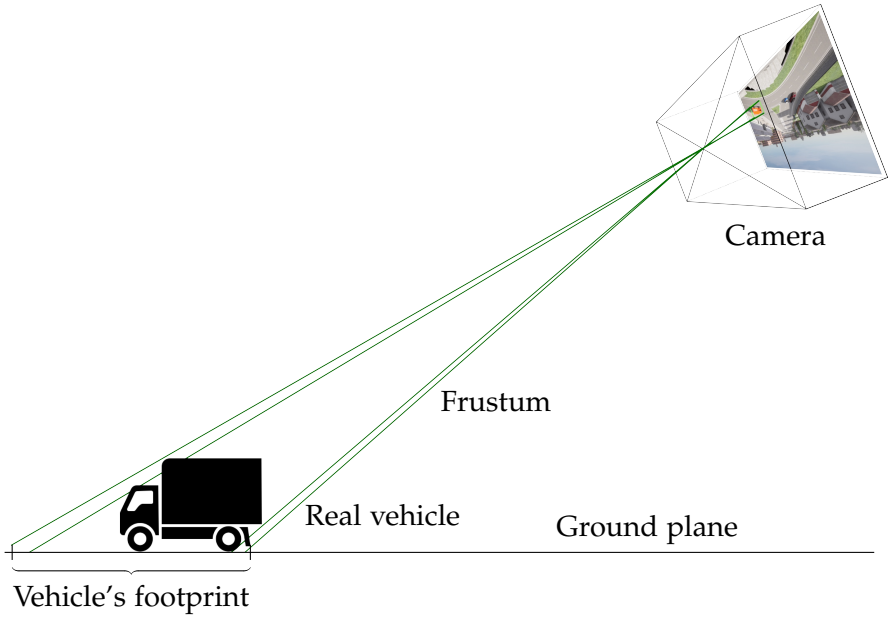


Figure 2.1: Inverse projection of a 2D bounding box in the space. The vehicle is inside the frustum built from the bounding box.

reverse path, for each 2D point on the image plane, an infinite number of 3D points exist in the world, all belonging to a half-line starting from the camera. This half line is frequently called a ray in the computer graphics field, and we will also use this terminology. Thus, for four 2D points positioned on the image plane, we obtain four rays crossing at the optical point of the camera before diverging forming a pyramid called frustum. By knowing the position of the camera in the world, it is possible to know the position of the frustum in 3D space. If we define our four points from a bounding box corresponding to a detected vehicle, then we know that this vehicle is somewhere in this frustum.

In the case where we take several PoVs overlapping each other, if a vehicle is present in the scene and is detected by all sensors: several frustums will be created (one per PoV). Although we do not know where the vehicle is in each frustum, we are sure that it is in all frustums. Therefore, the vehicle will be at their intersection. It is on the basis of this reasoning that we can generate a map of the dynamic elements detected in a scene.

CHALLENGES As dealt with in the previous chapter, several challenges are encountered when using cooperative systems. We have already discussed a solution to the problems of the network infrastructure limitation and data synchronization: the use of 2D bounding boxes. Indeed, these bounding boxes have the advantage of being able to be represented with very little data and can therefore be transmitted in a very short time, no matter the quality of the network. This also addresses the synchronization issue: with the GPS clock time that we can easily obtain associated with a small transmission delay, we can choose to neglect the synchronization issue. Moreover, the

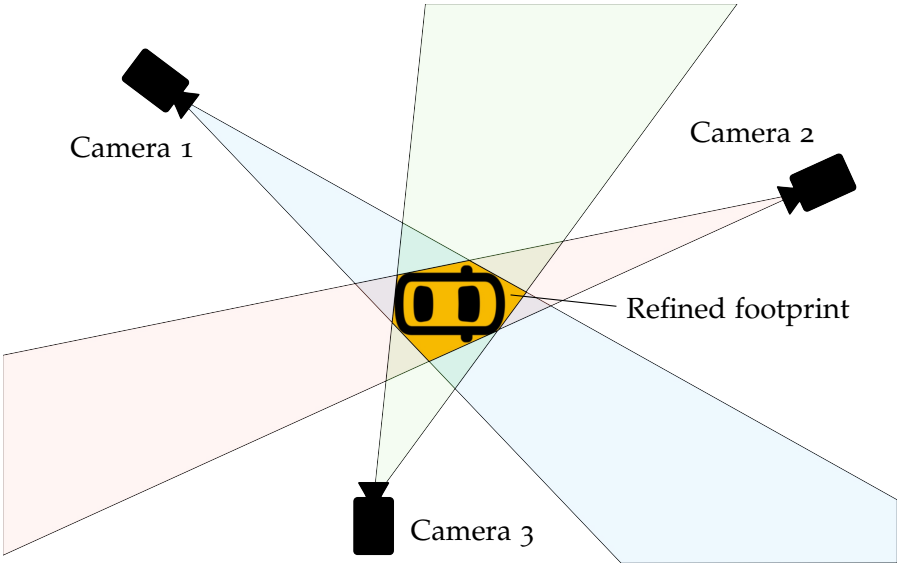


Figure 2.2: With multiple frustum, a finer zone corresponding to the real vehicle footprint can be found.

approach based on the intersection of frustums solves the problem of the appearance and the matching of the detected vehicles.

The calibration of the sensors is another challenge that we have to deal with in the following lines. Although the sensors on the infrastructure can be calibrated with respect to each other, it is impossible to manually estimate the placement of the sensors on the vehicles moving in the scene. In the following sections, we will present a method to take into account the positional noise of the sensors with respect to each other.

Finally, the question of the common container in which the perceived data will be represented and merged arises. Since volumetric maps are frequently used for information fusion, the format of the occupancy grid will be the focus of our attention in the rest of this manuscript. Another advantage of occupancy grids is that it is possible to represent several types of data. Thus, in future developments, we could imagine adding data from depth sensors like laser scanners without worrying about the multimodal aspect.

IN THE REMAINDER OF THIS CHAPTER We will first examine the architecture of our approach in section 2.2 before detailing the methods used in section 2.3. Finally we will detail our results in section 2.4 before concluding this chapter.

2.2 SYSTEM ARCHITECTURE

The architecture of our approach is based on two elements: the perceiving agents and the Road Side Unit (RSU). The perceiving agents, which we will call agents in the following lines, perceive the scene and transmit them to the RSU. The RSU is in charge of merging the information to create a global map of the scene that is transmitted to

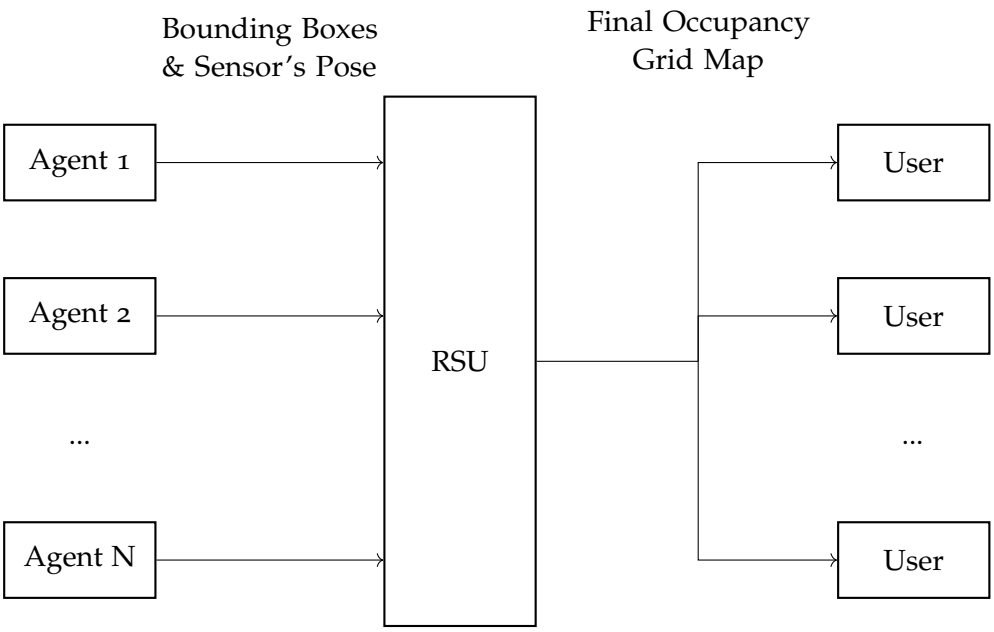


Figure 2.3: Macro organization of the agents and the RSU. The agents (vehicles or infrastructure) perceive the environment, the RSU processes the data to build a global semantic map shared to every CV

all the Connected Vehicle (CV)s. To implement this framework, we have resorted to using ROS [88]. In fact, ROS provides a framework for the communication between the agents and the RSU as well as a set of standards that will allow us to replicate our work or to use it in other projects.

2.2.1 *Agents*

The agents can be intelligent roadside sensors or connected vehicles and can be of an arbitrary amount in the scene. They are equipped with an image sensor and a system to identify vehicles in their field of view that extracts bounding boxes. Every agent publishes its messages on a global topic that will threfore contain every bounding box of every agent and will be read by the RSU as illustrated in Fig. 2.3.

In our work, we consider that the extraction of bounding boxes is derived from off-the-shelf solutions and is therefore not a topic covered here. We consider that timestamps are generated at the time of shooting from a GPS clock and thus the sensors are roughly synchronized. Therefore, we used the synthetic data from the ground truth to which we added random Gaussian noise in order to simulate synchronization, sensors' pose noise and detection noise.

2.2.2 *Road Side Unit*

The RSU is the central element of our framework. It aggregates all the data from the agents to form a map to be transmitted to all the CVs. The processing of the data to obtain the final occupancy grid is

divided into several tasks. Figure 2.4 represents the path of the data transmitted by the agents through these tasks. Note that we consider that the data transmitted by the agents takes the form of a continuous flow. It is the stacking task that discretizes the data temporally in order to be merged. The discretization interval is given by the agent that transmits data the fastest. In fact, the map is computed and transmitted as soon as the data from an agent that contributed in the previous interval is received. The different tasks, for which the mathematical details will be given in section 2.3, are described as follows:

MONTE CARLO UNCERTAINTIES SAMPLER:
: This block takes the bounding boxes and the sensor pose from which they are extracted and models the uncertainties by applying noise to the parameters on N samples created from each original measurement, with N the larger possible.

BACK PROJECTOR:
: This block uses the bounding box parameters for each of the N samples, finds the 4 corners of the bounding box, and projects them on the ground by ray tracing.

RASTERIZER:
: This block takes the 4 projected points on the ground of each bounding box and N samples and rasterizes them on the N occupation grid.

SAMPLE MERGER:
: This block merges the N occupancy grids forming a Local Occupancy Grid (LOG) for a sensor.

STACK:
: This block keeps the LOGs until the next block empties it.

Basic Belief Assigment (BBA):
: This block assigns, from the observations, the masses to the different classes used with the Dempster-Shafer Theory (DST) method to each cell of the occupation grids. This block appears only for the DST fusion. In other cases, it is bypassed.

COMBINER:
: This block merges the occupancy grids of the stack either based on the DST and the Basic Belief Assigment (BBA) values or directly with the probabilities contained in the occupancy grids.

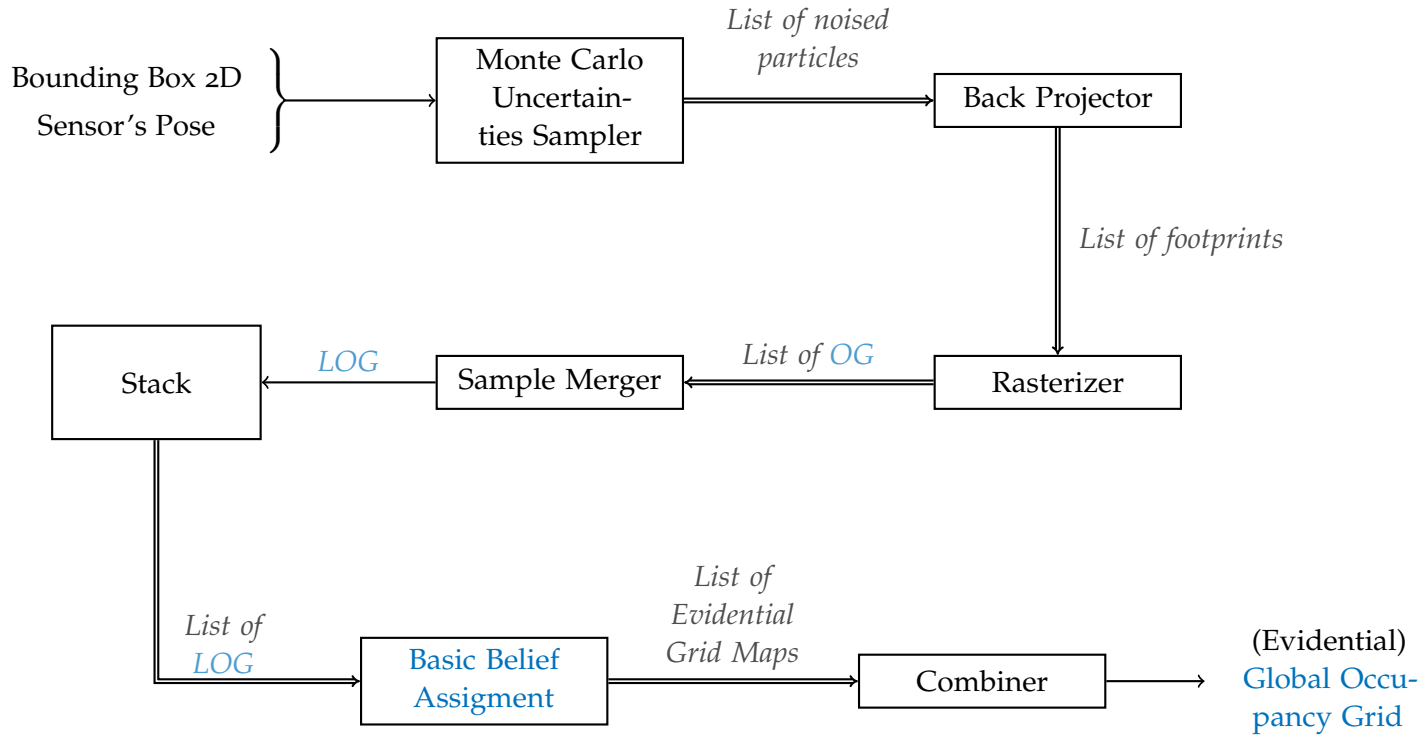


Figure 2.4: Illustration of bounding box processing and fusion framework in the RSU.

2.3 METHODS

In this section, we provide more details about the functioning of the blocks composing the RSU. We start with the basic principle of our system: the Back Projection, which allows us to obtain the silhouettes of the bounding boxes. We also present the methods allowing us to generate the LOG. Finally, we present the details of the methods for merging the LOG. As we go along, we also give details on the use of Monte Carlo methods.

2.3.1 *Back Projection*

In order to be merged together, the 2D bounding boxes provided by the agents must be placed on a map. Since, today, vehicles are in physical contact with the ground, we can consider that the position of the vehicle on the map is at the intersection with the frustum from the bounding box with the ground. In other words, we want to make a projection of the bounding box from the camera plane to the ground plane under the vehicle's coordinate. This procedure is also called IPM [63]. The intrinsic parameters of the camera and the extrinsic transformation from the camera to the vehicle's center are previously calibrated. The technique used in our approach to perform this IPM is based first on the calculation of the frustum from the bounding box's corners and then the intersection of this latter with the ground plane.

To compute the frustum related to a bounding box, we will use the Plücker coordinate system as detailed below. Then, we will present the persepective transform using the pinhole camera model followed by the inverse perspective transform.

2.3.1.1 *Plücker Coordinate System*

The Plücker coordinate system is frequently used in computer graphics and computer vision. This coordinate system is described by Hartley and Zisserman in [53]. Indeed, this coordinate system allows to perform 3D geometry operations in homogeneous coordinates, offering concise solutions that we will detail. In this case, we will look for intersection points between the ground plane and rays constituting the edges of the frustum formed from each bounding box. The ground plane can be represented in Plücker coordinates as welle as the rays which are symbolized by Plücker's line. Therefore, we benefit from the concise solution offered by using the Plücker coordianates.

PLANE We can define a $\boldsymbol{\pi}$ plane in homogeneous coordinates by Equation (2.1).

$$\begin{aligned} \boldsymbol{\pi} &= (\pi_1, \pi_2, \pi_3, \pi_4)^\intercal \\ \pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 &= 0 \end{aligned} \tag{2.1}$$

Where, π_1, π_2, π_3 are the coordinates of the normal vector of the plane, and π_4 is the distance between the origin $\boldsymbol{O}$ and the plane $\boldsymbol{\pi}$. Therefore, we can construct the vector $\boldsymbol{\pi}$ using the normal vector of

the sought plane and its distance from the origin such that $\boldsymbol{\pi} = [\boldsymbol{N}|d]$ where $\boldsymbol{N} \in \mathbb{R}^3$ is the normal vector of the plane and d is the distance between the origin $\boldsymbol{O}$ and the plane $\boldsymbol{\pi}$, as defined in Equations (2.2).

$$\begin{aligned} \boldsymbol{N} &= (\pi_1, \pi_2, \pi_3)^\intercal \\ ||\boldsymbol{O\pi}|| &= \pi_4 \end{aligned} \tag{2.2}$$

LINE A ray, or line, can be defined by two points in homogeneous coordinates $\boldsymbol{A} = [x_1, y_1, z_1, 1]^\intercal$ and $\boldsymbol{B} = [x_2, y_2, z_2, 1]^\intercal$. The definition of the line $\boldsymbol{L}$ from these two points is defined by Equation (2.3).

$$\boldsymbol{L} = \boldsymbol{AB}^\intercal - \boldsymbol{BA}^\intercal \tag{2.3}$$

INTERSECTION Given the ground plane and the set of four rays corresponding to the four corners of each bounding box, the intersection between each ray $\boldsymbol{L}$ and the ground plane $\boldsymbol{\pi}$ can be found according to Equation (2.4) in non-normalized homogeneous coordinates.

$$\boldsymbol{P}_{intersection} = \boldsymbol{L\pi} \tag{2.4}$$

where $\boldsymbol{P}_{intersection}$ is a four-dimensional vector in homogeneous coordinates. Thus, we obtain for each bounding box a set of four projected points on the ground plane forming a polygon that represents a silhouette on the ground of the detected object at a given glspov. To obtain the final 3D point, it is required to normalize $\boldsymbol{P}_{intersection}$.

2.3.1.2 *Inverse Projection*

To obtain the projections of the bounding boxes on the ground in order to find the position of the vehicles, we want to use the frutum that these bounding boxes form with the optical center of the camera. The four edges constituting the frustums are considered as rays passing through the optical center of the camera and the corners of the bounding box (for each ray, a corner of the bounding box). The pinhole camera model allows to find rays passing through a 3D object and the optical center of the camera to obtain a 2D point on the image plane. We will present this model in the next paragraph before doing the reverse path to find the 3D rays from points on the image plane and the optical center of the camera. From these rays, we will explain how we get the four corners projected on the ground belonging to each 2D bounding box.

PINHOLE CAMERA To make an inverse projection, we have to see first the projection model of the objects in the plane 2D of the image. For that, we can use the pinhole model as defined in [53]. This model, defined in the equation (2.5), allows to project a 3D point in the camera frame of coordinates $\boldsymbol{P}_{cam} = (X, Y, Z)^\intercal$ on the image plane with coordinates $\boldsymbol{p}_{img} = (u, v, w)^\intercal$ after normalization by the value of w.

$$\boldsymbol{p}_{img} = K\boldsymbol{P}_{cam} \tag{2.5}$$

where K is the intrinsic camera matrix defined in the equation (2.6) and constructed from f_x and f_y which both here equals to f, the focal

length, c_x and c_y the coordinates of the camera optical center on the image.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.6}$$

However, it should be noted that this model ignores the optical deformations that the lenses can bring. In this thesis, we consider these optical deformations as neglectable.

INVERSE PINHOLE CAMERA To go from a point 2D to a point 3D in the camera frame, we can use the inverse principle of what we saw before. Thus, for a point $\boldsymbol{p_{img}} = (u, v, w)^\intercal$ on the plane, there exists a point $\boldsymbol{P_{cam}} = (X, Y, Z)^\intercal$ in the camera frame, as given in the equation (2.7).

$$\boldsymbol{P_{cam}} = K^{-1}\boldsymbol{p_{img}} \tag{2.7}$$

However, since we only have the coordinates u and v of the point in the image and the value of w is lost, the values of $\boldsymbol{P_{cam}}$ will depend on w.

Therefore, instead of having a fixed point, we have a line defined by all values of w passing through the center of the camera and the real point in the world $\boldsymbol{P_{real}}$.

We can construct a ray $\boldsymbol{R_p}$ from the point corresponding to the center of the camera, which we will name $\boldsymbol{C_{world}}$, expressed in the world frame, as well as a reprojected point of $\boldsymbol{p_{img}}$, called $\boldsymbol{P_{cam}}$. $\boldsymbol{P_{cam}}$ is reprojected using an arbitrary value of $w \neq 0$ in the world frame using ${}^W T_C$ and named $\boldsymbol{P_{world}}$. Where ${}^W T_C$ is the transformation matrix from the camera frame to the world. Equation (2.8) expresses these steps.

$$\begin{aligned} \boldsymbol{R_{real}} &= (\boldsymbol{C_{world}}\boldsymbol{P_{real}}) \\ \forall w \neq 0, \exists \boldsymbol{P_{cam}}, \boldsymbol{P_{world}} &= {}^W T_C \boldsymbol{P_{cam}} \in \boldsymbol{R_{real}} \\ \Rightarrow R_p &= \boldsymbol{C_{world}}\boldsymbol{P_{world}}^\intercal - \boldsymbol{P_{world}}\boldsymbol{C_{world}}^\intercal \end{aligned} \tag{2.8}$$

2.3.1.3 *Silhouette's Estimation*

The silhouettes are obtained from the intersection of the 3D frustum, formed by the four rays passing through the optical center of the camera and the four corners of the bounding boxes on the image plane, and the ground plane. Since we now have rays $\boldsymbol{R}$ created from the corners of the 2D bounding boxes and the center of the camera. The silhouettes are thus formed by these rays coming from the four corners of each of the bounding boxes and the ground plane $\boldsymbol{\pi_{sol}}$ according to the equation (2.9).

$$\boldsymbol{P_{sol}} = \boldsymbol{R}\boldsymbol{\pi_{sol}} \tag{2.9}$$

If a corner of a bounding box is above the horizon, then it will be projected to infinity of the map. In this case, we will not try to use the intersection with the ray pointing to the sky and the ground plane since this intersection will be at the back of the camera, which is absurd. We therefore recalculate the point $\boldsymbol{P}_{world}$ but this time with a value of w greater than the size of the map ($\sqrt{2} \cdot mapsize$ for instance). We will thus obtain $\boldsymbol{P}_{world}$ outside the map which we can consider it as at the infinite. To reproject it on the ground plane, we will just place its altitude at that of the ground plane on the same $< x, y >$ coordinates.

2.3.2 *Local Occupancy Grids generation*

We consider a Local Occupancy Grid (LOG) as an occupancy grid containing the information provided by a single sensor. Let $\mathcal{M}$ be the occupancy grid map over a region of interest divided into square cells $\mathcal{M}_{x,y}$ where $\langle x, y \rangle$ correspond to the position of the cell $\mathcal{M}_{x,y}$ within $\mathcal{M}$ as defined in [106]. Thus, the problem addressed is the determination of the probability of occupancy of each grid cell given the measurements. The assigned values to $\mathcal{M}_{x,y}$ are $\{m_{x,y} \in \mathbb{Z} | -1 \leq m_{x,y} \leq 100\}$ where -1 denotes a cell of unknown occupation, 0 denotes a free cell and 100 an occupied cell as given in ROS documentation.

2.3.2.1 *From Ground coordinates to Occupancy Grid coordinates*

Let δ be the length of the square cell in meters, (O_x, O_y) correspond to the position of the origin in the occupancy grid. The position of a silhouette is obtained in cell coordinates from metric coordinates according to (2.10),

$$\mathbf{x}_{grid} = \left\lceil \begin{pmatrix} 1/\delta & 0 & 0 & O_x \\ 0 & 1/\delta & 0 & O_y \end{pmatrix} \boldsymbol{X}_{gnd} \right\rceil \tag{2.10}$$

where $\mathbf{x}_{grid}$ is a 2-vector and $\boldsymbol{X}_{gnd}$ a 4-vector.

2.3.2.2 *Rasterization*

Since the silhouettes are in topological format, it will be difficult to merge them together. Therefore, we convert the topological information describing the silhouette into volumetric information, which is a set on cells on a grid, by resterization. This step consists in defining for each cell if it belongs to a silhouette (and thus is occupied), to the terrain (defined as free) or if it has not been observed. We will use either occupancy grids or evidential grids, depending on the desired fusion method.

First, all our cells are considered as unobserved (-1). Then, the whole area covered by the camera is considered as free (0). To define this area, we use the principle of inverse projection explained previously but with the 4 corners of the image. Finally, the cells belonging to a silhouette are considered as occupied (100).

The map resulting from this operation thus takes the format of a grid where each cell contains a label (unknown, free or occupied). This grid can be denoted $M_{<x,y>}$ where $< x, y >$ are the cell coordinates. Since the latter has a similar structure to the images, we used the tools offered by image processing library to perform the rasterization task. To plot the silhouettes on the occupancy grid, we used the function `fillPoly` of the OpenCV API [17]. We use this function in 8-connected lines mode, also called Moore's neighborhood to draw the polygons constituting the silhouettes. This mode takes into account the 8 cells bordering around a cell to draw a line, contrary to the 4-connected.

2.3.2.3 *Modeling uncertainties*

The position estimation of the camera in the scene is subject to noise as well as the bounding box position and dimension determination on the image. To model these uncertainties, we created N samples from each original measurement, with N the larger possible. Then, we applied noise to the pose estimation and bounding box estimation parameters for each of the sample. The noise follows a Gaussian distribution with parameters μ the original measurement and σ the standard deviation presented in [66] and in [1]. Each of the N samples is projected on N sample grids and then merged by averaging the cells.

2.3.3 *Local Occupancy Grids Merging*

Since each sensor provides a LOG, these latter have to be merged in order to create a global one. The LOG is already created with respect to a global frame reference and can therefore be directly merged without frame transformations. In fact, two main paradigms have been investigated in the state of the art to perform the merging namely the probabilistic approach and the Evidential approach.

Let $\mathcal{M}$ be a Global Occupancy Grid (GOG). Let's consider a LOG $\mathcal{M}_l$ and $\mathcal{M}^i_{x,y}$ a given cell of $\mathcal{M}_l$ where $\langle x, y \rangle$ refer to the location of the cell and i to the index of the agent $1 \leq i \leq N_A$ with N_A the number of the available agents.

2.3.3.1 *Probabilistic merging method*

The first method we implemented is the Bayesian fusion method, as proposed by the authors of [14, 41]. This method consists in using probability theory to estimate the probability that two images are similar.

Agents perform perception independently, *i.e.*, they do not take into account the observations of other agents to define the bounding boxes to be detected and they do not take into account past observations. Hence, we can make the assumption that there is no dependency between different observations of a given cell. The joint probability of a cell that is being observed by two agents where agent 1 performs an

observation denoted as o_1 and agent 2 as o_2 is expressed in Equation (2.11).

$$P(o_1 \cap o_2) = P(o_1) \times P(o_2) \tag{2.11}$$

Since this operation is associative, for N agents, we can compute the probability associated to a cell of the global map $\mathcal{M}_{\mathcal{P}<x,y>}$,where the indice $\mathcal{P}$ indicate the map is issued from a probabilistic fusion, from the maps issued from the agents $\mathcal{M}_{<x,y>i}$ according to Equation (2.12), i being the index of the agent.

$$\begin{aligned} &\forall x \in [0,m],\ y \in [0,n] \\ &\mathcal{M}_{\mathcal{P}<x,y,c>} = \prod_{i}^{N} \mathcal{M}_{<x,y>i} \end{aligned} \tag{2.12}$$

We propose two methods that perform a product between cells to determine its occupancy probability as given in Equation (2.12) [41]. The former, named **inter1**, considers the cells having an unknown state (-1) as having a probability of 0.5 before performing the product of the cells. The latter, named **inter2**, ignores the cells having an unknown value (-1) in the product. For both of them, values between 0 and 100 are divided by 100.

2.3.3.2 *Evidential merging method*

Another possible method of merging LOGs is to use the evidential theory, also called Dempster-Shafer Theory (DST) [97] as the authors of [19] did. This theory is based on a set of classes with associated masses. It is these masses that we will be able to merge using a merge rule. We will detail these different concepts in the following paragraphs.

2.3.3.3 *Classes*

In our work, we used a set of classes given in Equation (2.13), where $\mathcal{O}$ describes the status of an occupied cell and $\mathcal{F}$ that of a free cell.

$$\Omega = \{\mathcal{O}, \mathcal{F}\} \tag{2.13}$$

Ω represents the available universe of classes and we will use it later. In addition, there is an internal state that we use to define whether a cell has been observed or not. This will be treated differently depending on the merge mode.

2.3.3.4 *Evidential Grids*

To perform a merge in the framework of the DST, it is necessary to create evidential maps.

GRID DEFINITION The map takes a format very similar to the occupancy grids we presented in 2.3.3.1 but have more sub-cells than $|\Omega|$. In fact, they are made of $|2^{\Omega}|$ sub-cells, 2^{Ω} being the power set of Ω defined in the equation (2.14). This evidential grid format was notably used by Richter et al. in [90]. We will note this map $\mathcal{M}_{\mathcal{E}<x,y,c>}$ with $< x, y >$ the coordinates of the cell and the indice $\mathcal{E}$ indication we this map is an evidential map, c the index of the sub-cell (one per element of the power set).

$$2^{\Omega} = \{\varnothing, \{\mathcal{O}\}, \{\mathcal{F}\}, \Omega\} \tag{2.14}$$

The advantage of using a power set is that we can take into account states of doubt or unknowns. For instance, in the case where a vehicle is in the scene but occluded by the terrain from the PoV, it would be classified as a terrain while seen from other PoV it would be easily classified as a vehicle. Thus, we would want to apply to Ω a value to reflect the unknownness of the observation. In another case, if a cell has not been observed, we can consider that the confusion between all classes is maximal. Therefore, we will consider only the set Ω.

MASSES In the previous paragraph, we mentioned masses. They are similar to the probability values used in the Bayesian theory, but they are applied to the sets of a power set and are defined according to the equation (2.15).

$$\begin{aligned} m &: 2^{\Omega} \rightarrow [0, 1] \\ m(\varnothing) &= 0 \end{aligned} \tag{2.15}$$

When associating values with masses, it is necessary to follow the property of the equation (2.16).

$$\sum_{A \in 2^{\Omega}} m(A) = 1 \tag{2.16}$$

These are the masses that are stored in the sub-cells of the evidential grids.

BASIC BELIEF ASSIGNMENT FUNCTION The association of a mass with a 2^{Ω} status is performed by a function named BBA. Our basic belief assignment function is given by the Algorithm 1. When a cell is not observed, the uncertainty is maximal which leads to a value of $m(\Omega) = 1$. We then distinguish two cases. The first is the case where the observation is made by a vehicle. The vehicles are more affected by occlusions than the infrastructure. Therefore, the non-detection of obstacles is more uncertain. On the other hand, since vehicles have a PoV in the scene, when an object is detected, its bounding box is more accurate than the infrastructure PoV. Thus, the cells observed by the vehicles apply a value to the obstacle mass and uncertainty. On the contrary, the infrastructure PoV will apply values on the masses of the free case and on the uncertainty. Indeed, I make the hypothesis that the infrastructure suffers much less from occlusions.

Algorithm 1 Basic Belief Assignment

Require: $\mathcal{C} \in \mathcal{M}$
  $m(\varnothing), m(O), m(F), m(\Omega) \leftarrow 0$
  if C = -1 **then**
    $m(\Omega) \leftarrow 1$
  else
    if C is from an infrastructure sensor **then**
      $m(F) \leftarrow 1.0 - \frac{\mathcal{C}}{100}$
      $m(\Omega) \leftarrow \frac{\mathcal{C}}{100}$
    else
      $m(O) \leftarrow \frac{\mathcal{C}}{100}$
      $m(\Omega) \leftarrow 1.0 - \frac{\mathcal{C}}{100}$
    end if
  end if

Once each cell of each LOG has had its masses assigned, it is possible to merge them one by one with Dempster's rule of combination given in the equation (2.17) to merge two cells where $X \in 2^{\Omega}$ is defined by equation (2.18) with $K = \sum_{Y \cap Z = \varnothing} m_1(Y) m_2(Z)$.

$$m_f(X) = m_1(X) \oplus m_2(X) \tag{2.17}$$

$$m_f(X) = \frac{1}{1-K} \sum_{Y \cap Z = X \neq \varnothing} m_1(Y) m_2(Z) \tag{2.18}$$

$$m_{out}(X) = \bigoplus_{i=0}^{N} m_i(X) \tag{2.19}$$

Dempster's rule of combination being commutative and associative, it is, therefore, possible to combine N masses as expressed in equation (2.19). Thus, we combine the masses associated with cells of the same coordinate in each layer.

OCCUPANCY GRID FORMATING We propose two methods to associate the values to the cells from the masses of the final grid. The former, named **dst1**, directly assigns to the cell the mass of the set O while the latter one, named **dst2**, follow the rule given in Algorithm 2.

Algorithm 2 Evidential grid map to occupancy grid map rule for **dst2**

Require: $m(O), m(F), m(\Omega)$
  $\mathcal{C} \in \mathcal{M}_{out}$
  if $m(F) > m(O)$ **and** $m(F) > m(\Omega)$ **then**
    $\mathcal{C} \leftarrow m(O) \times 100$
  else if $m(\Omega) > m(O)$ **and** $m(\Omega) > m(F)$ **then**
    $\mathcal{C} \leftarrow -1$
  else
    $\mathcal{C} \leftarrow 100$
  end if

2.4 RESULTS

Since we have detailed the methodological points used to implement our approach in the previous section, we will now discuss the results given by our method. We will start by discussing the dataset we used to evaluate the method. We will then discuss the qualitative evaluation of the results. Finally, we will make a quantitative study of the results in which we will also give details of the metrics used for the evaluation.

2.4.1 *Carla dataset*

To the best of our knowledge, we have not identified any dataset that delivers a vehicle-infrastructure cooperative experimental framework at the moment we were working on our approach. Therefore, we created a dataset generator based on the CARLA simulator [32] allowing the generation of datasets with one or more viewpoints from infrastructure and vehicles. For the works of this chapter, we generated a dataset with 4 agents: 3 connected vehicles and an infrastructure. The vehicles pass through the roundabout and are in the field of view of the infrastructure. Also, some vehicles will enter the field of view of one or more other vehicles and have their fields of view overlapping as shown in Fig. 2.5. Each agent can have different sensors:

- $1\times$ RGB camera (90° fov, 1384×1032 pixels)
- $1\times$ Depth camera (90°, 1384×1032)
- $1\times$ Semantic segmentation camera (90°, 1384×1032)
- $1\times$ LiDAR (32 layers, 40° vertical fov)

For the different agents, the rigid transformation between each onboard sensor and the attached reference frame is the same. For the infrastructure, the sensors are positioned at $13m$ altitude at the center of the roundabout (located at the scene's center) and with a pitch of -20°. For the vehicles, the sensors are located at $1.9m$ above the chassis. In addition to the raw data, the vehicle's state is stored in a JSON file for each frame. This latter contains the sensor's transformation matrix with respect to the world's reference frame as well as the vehicle's transformation matrix, linear velocity, angular velocity, acceleration, forward vector, and 3D bounding box.

In order to generate the ground truth, we used the JSON files. We retrieve the 4 points forming the bottom plan of the bounding box and place them in the scene with the given transformation matrix to express their coordinates in the world reference frame. We get a perfect polygon forming the footprint of the vehicle which is then rasterized on the grid. Fig. 2.6e illustrates at the frame 155 of the dataset an example of the map saved where black color corresponds to a value of 100 and white color corresponds to a value of 0. Alongside, in Fig. 2.6, the outputs of each above-mentioned algorithm are featured.

Figure 2.5: Synchronous video frames from each camera of our multi-agent dataset made with CARLA.

2.4.2 *Qualitative Evaluation*

In the first instance, we carried out a qualitative study to ensure that our solution gave us usable results. It is notably thanks to this study that we validated our BBA function, explained in the previous section. This study is based on the dataset that we created which contains 280 frames. These 280 frames are divided into 6 sub-scenarios of cooperation defined as follows:

Seq 0: This sequence corresponds to the best coverage from the cars. Each car sees at least one other vehicle.

Seq 1: The infrastructure coverage is maximal: each vehicle is visible from the infrastructure's point of view.

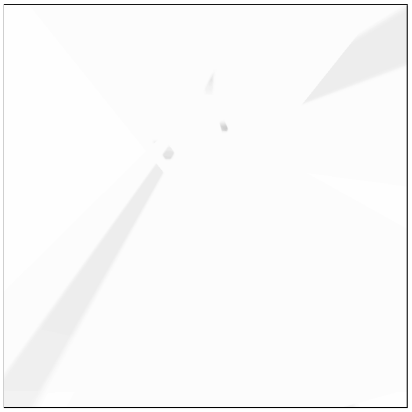
Seq 2: The coverage is maximal from every agent. Each car is seen by at least one vehicle and by the infrastructure.

Seq 3: This sequence gives an example of partial coverage where both vehicle and infrastructure operate but not every car is seen by the infrastructure.

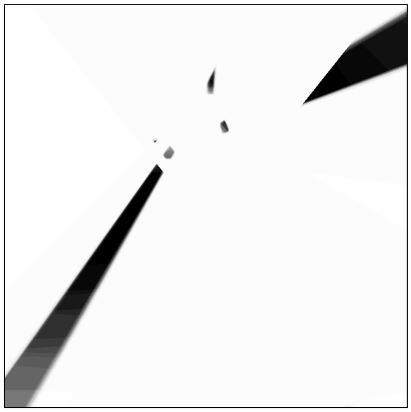
Seq 4: This sequence features monomodal detection. This means that cars are detected either by the infrastructure or by other vehicles but not both.

Seq 5: This sequence features single detection. The detected cars are detected by only one agent and thus is not a cooperative situation.

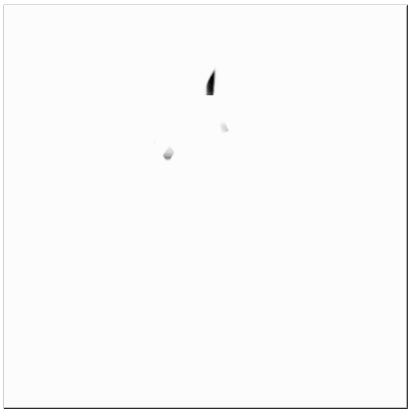
Through these sub-scenarios, we observed that the performances vary significantly. Therefore, these variations will be studied in the next section.



(a) Using inter1.

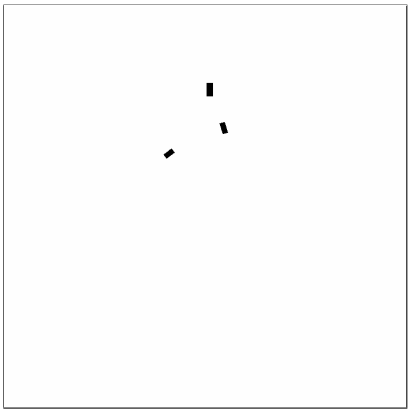


(b) Using inter2.



(c) Using dst1.

(d) Using dst2.



(e) Ground truth.

Figure 2.6: Occupancy grid map for different methods (frame 155).

2.4.3 *Quantitative Evaluation*

We based our quantitative evaluation on Intersection over Union (IoU) and F1-score which are two common metrics for occupancy grid evaluation. Both of them are based on the number of True Positive (TP), False Positive (FP), and False Negative (FN). To define if a cell is positive, we compare its value to a threshold. IoU is defined as in equation (2.20) and F1-score is defined in (2.21).

$$IoU = \frac{TP}{TP + FP + FN} \tag{2.20}$$

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{2.21}$$

Table 2.1 gives an overview of the IoU and the F1-score for each sequence and each algorithm. These results were given with a noise applied following a normal distribution with a standard deviation of $\sigma = 0.0243m$ on the lateral and longitudinal position and of $\sigma = 0.0518m$ on the altitude as we can find in [1]. For the rotations, the noise follows a normal distribution with a standard deviation of $\sigma = 0.1°$ on all axes as we can find in [66]. The bounding boxes have a normal distribution noise with a standard deviation of 5 pixels applied on each edge of the bounding box. The threshold was set at 0.5 but requires in-depth research to determine its impact on the results.

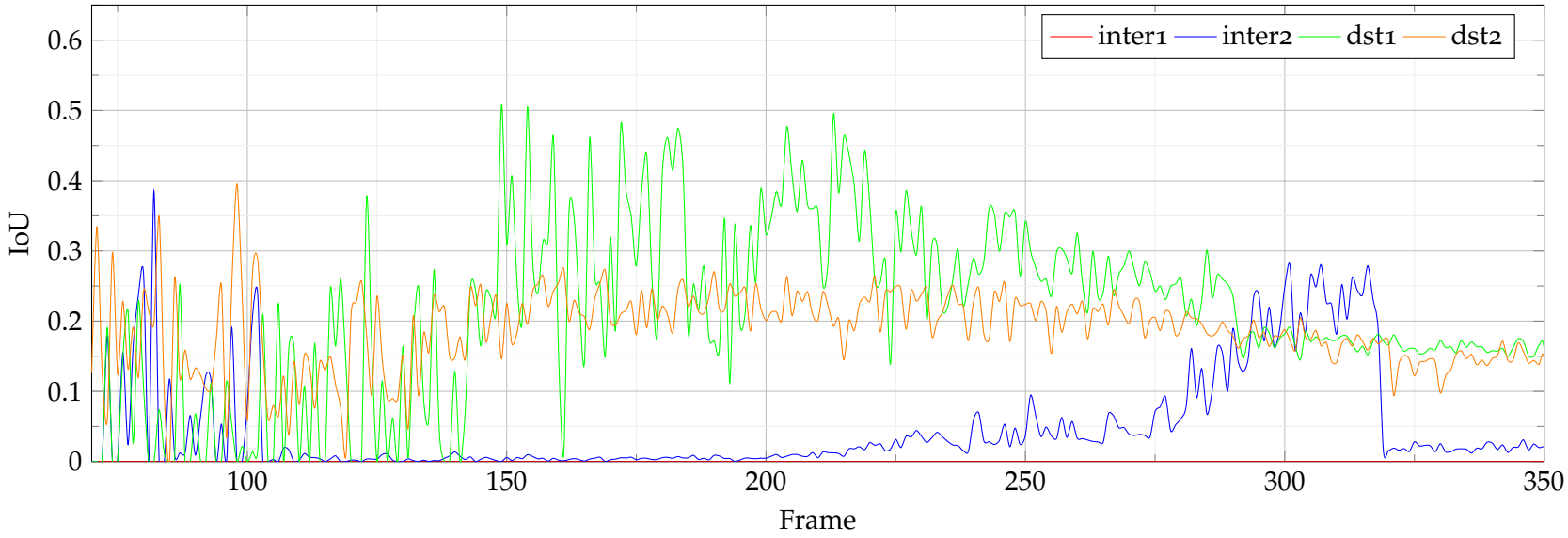


Figure 2.7: Example of the evolution of the IoU with a threshold of detection of 0.50 (normalized) with 3 vehicles transiting in a roundabout.

Table 2.1: Example of the evolution of the IoU and F1 scores with a threshold of detection of 0.50 (normalized) with 3 vehicles transiting in a roundabout.

Algorithm	**Metric**	**Seq 0**	**Seq 1**	**Seq 2**	**Seq 3**	**Seq 4**	**Seq 5**	**Total**
inter1	IoU	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
inter1	F1 Score	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
inter2	IoU	0.052524	0.005595	0.004103	0.022008	0.214580	0.019923	0.013490
inter2	F1 Score	0.099806	0.011128	0.008172	0.043068	0.353341	0.039067	0.026620
dst1	IoU	0.055993	0.305245	0.287551	0.233977	0.174009	0.162757	0.223513
dst1	F1 Score	0.106048	0.467721	0.446663	0.379224	0.296435	0.279951	0.365362
dst2	IoU	0.175572	0.217005	0.213857	0.175982	0.172789	0.142080	0.188038
dst2	F1 Score	0.298700	0.356621	0.352359	0.299293	0.294663	0.248810	0.316551

We note that **inter1** obtains a global result 0%, either on IoU and F1-score, showing that the basic probability-based occupancy grid fusion method as presented in [41, 105] is not usable in this situation. However, **inter2** offers slightly better results with a maximum IoU of 21.45% in sequence 4 as shown in Fig. 2.7 but a global IoU of 1.35%.

This behavior is explainable by the fact that **inter2** is a modified version of **inter1**. Indeed, although these two algorithms follow the same merging rule, **inter2** excludes the unobserved cells during the merging. This avoids that individual detections be removed from the final map because of the successive multiplications by 0.5 that **inter1** would perform in individual detections. However, this approach shows its limits since beams appear when projecting the frustums as shown in Fig. 2.6b, thus increasing the number of false negatives. In sequence 4, the position of the vehicles offers viewpoints to the agents allowing them to reduce the beams due to frustum and thus to reduce the number of false positives. Nevertheless, this situation disappears when moving to sequence 5, and the false-positive rate increases dramatically.

Regarding the results obtained with our method, the algorithms based on the Dempster-Shafer theory (DST) offers much better results than the standard method cited in the previous paragraph. The fusion algorithm **dst1** offers a maximum IoU of 30.52% in the sequence 1 and a global IoU of 22.35% while **dst2** offers a global IoU of 18.8%.

The fusion algorithms **dst1** and **dst2**, based on the DST, show much better results since the DST allows the management of cells with an unknown state. We can consider that the distribution of masses can give a hint on the confidence of a measurement. Thus, when a cell is not observed, the confidence associated with this measurement is null. The consequence of this behavior is the elimination of the beams as observed in the methods **inter1** and **inter2** and thus the reduction of the false positives. We notice a more erratic behavior on Fig. 2.7 until frame 140. This is due to the fact that a vehicle is too far away to be detected which corresponds to a false negative. Moreover, the vehicles are distant from each other, which has the consequence of amplifying the observation errors. As for the last sequences, the vehicles move away from each other and leave the field of view of the infrastructure, thus increasing the measurement errors. Therefore, we can conclude that the results are given at the beginning and the end of the traffic circle transit as given in Fig. 2.7 are due to measurement errors. To conclude, we note that **dst1** and **dst2** do not seem to be affected by the arrangement of the vehicles as is **inter2** and therefore **dst1** and **dst2** are more robust than the state of the art methods while providing better results.

2.5 CONCLUSION

In this chapter, we presented a new approach for cooperative perception in order to create an evidential occupancy grid map. We used the

vehicle PoV in addition to the infrastructure PoV in order to build confidence at low cost.

In addition, we propose a method for cooperative generation of evidential occupancy grid using only the two-dimensional bounding boxes given by an image sensor as well as the position of that sensor with the aim of keeping the system's cost low as well as reducing the load on the communication system.

Finally, we propose a study on different data fusion methods based either on a Bayesian approach or on a Dempster-Shafer based approach on which we observe much better results. We have validated our results on a cooperative dataset that we have created from the CARLA simulator that we provide and to which we have added measurement noise.

We were able to validate our approach and the general idea of creating maps from sparse data. However, this map only shows the obstacles without giving the nature of the obstacle. Moreover, several axes remain to be explored, notably on the decision making and the assignment of masses. Finally, the dataset on which we validated our results lacks scenarios. We explore these elements in the next chapter.

3

MULTI-AGENT COOPERATIVE CAMERA-BASED SEMANTIC GRID GENERATION

ASSOCIATED ARTICLE

[1] Antoine Caillot, Safa Ouerghi, Yohan Dupuis, Pascal Vasseur, and Rémi Boutteau. "Multi-Agent Cooperative Camera-Based Semantic Grid Generation." In: *UNDER REVIEW - IEEE Robotics and Automation Letters* (2022).

3.1 INTRODUCTION

In the previous chapter, we have introduced a new approach using in-scene vehicle PoVs in addition to infrastructure PoVs to monitor a road section. We have used the 2D bounding boxes of the detected objects instead of the image stream to reduce the pressure on the network and have proposed a dynamic object occupancy grid of the scene. To merge the data from all PoVs, we have implemented two methods: the first one based on the Bayesian theory and the second one on the DST. Our results showed a large benefit with the DST-based method but must be put in perpective because of the rudimentary aspect of the implemented decision making method. In addition, the dataset we used to test the viability of our approach is very limited.

Semantic information brings a new level of understanding of the scene and allows to estimate parameters that are not measured. Thus, a user can assume that a vehicle is moving faster than a pedestrian and can better prioritize his actions. This paradigm is also true within our system. Indeed, through the implicit estimation of parameters of an object, we can estimate the maximum size of an object according to its associated class. This is why, in this chapter, we will develop our approach to integrate a semantic notion. To achieve this, we will have to adapt our architecture as well as the creation of local grids. We will test several decision making methods and evaluate our approach on several datasets.

3.2 ROAD SIDE UNIT ARCHITECTURE

We will not present the general architecture of the approach since it has been done in Chapter 2. However, although the general idea is the same, taking into account the semantic aspect requires some modifications in the data processing. Fig. 3.1 shows an updated version of the path of the data passing through the RSU and its different processing blocks and up to the creation of the final map.

The RSU is constituted of two sets of blocks. The first, made up of the back projector, rasterizer and BxA (Basic Probability Assigment

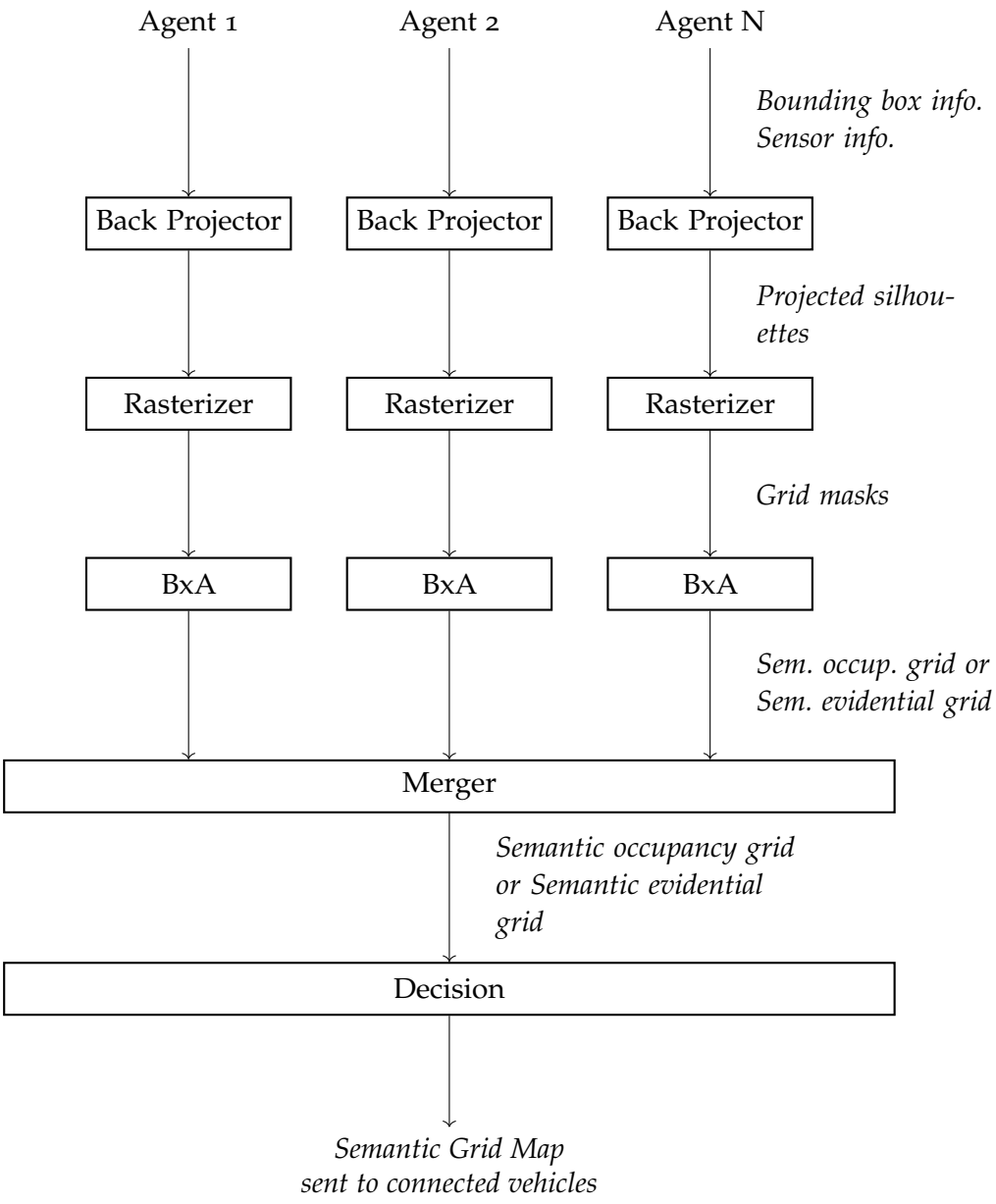


Figure 3.1: Pipeline of the data from the agent to create a semantic grid map. The illustration shows the example with 3 agents, and thus, 3 parallel processings before the merge of the grids.

(BPA) or BBA) blocks, which is intended to perform a first processing on the data sent by each agent. In fact, for each agent, an instance of this first set is created and several instances may, therefore, be created in parallel. Since the output of this set has, not yet, benefited from the cooperative aspect, it is considered as local processing. This latter takes the form of a grid and we will, therefore, refer to it as local grid. The second set of blocks is intended to merge the local grids into a global semantic grid and thus, performs global treatments.

3.2.1 *Local Processing Blocks*

The local treatment consists of 3 blocks.

BACK PROJECTOR:
: This block uses the bounding box and sensor information to make an inverse projection of the bounding boxes onto the ground in the world frame. The produced silhouettes are associated with the label given with the bounding box.

RASTERIZER:
: It allows to create masks in the form of grids from the topological information of the previous block. Instead of storing the probability of occupancy as in chapter 2, the cells store a number corresponding to the label given by the silhouette.

BXA:
: This interchangeable block takes the format of BPA to convert the masks into a probabilistic occupancy grid or the format of BBA to convert the masks into an evidential occupancy grid.

3.2.2 *Global Processing Blocks*

The set performing the global treatment consists of 2 blocks depending on the type of input grid.

MERGER:
: This block merges the local grids of each user using either a Bayesian or a DST based method. This block is very similar to the one presented in chapter 2. However, we presume the data synchronised between the different agents.

DECISION:
: Finally, the global occupancy or evidential grid are converted into a semantic grid. This block must therefore make a decision about each cell belongs to which semantic class among a finite number of available semantic classes.

At the output of this set, we obtain a semantic grid map indicating where the objects are located. In the scope of this chapter, only semantic classes of "pedestrians" and "vehicles" are considered. The other cells are considered as terrain, the default class. However, the number of classes can be extended to any number.

Figure 3.2: Bounding boxes for cars and pedestrians with their two lower points on the ground as given from our Dataset built from CARLA. Green boxes represent the 2D bounding boxes extracted from the 3D bounding boxes given by the simulator.

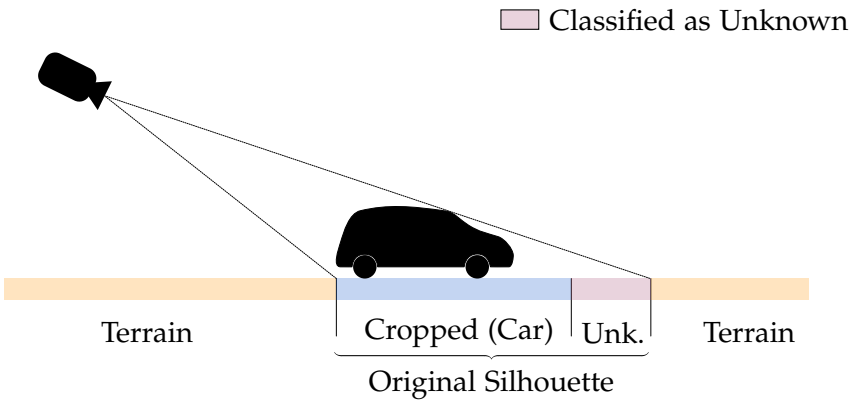
3.3 LOCAL GRID MAPS

In this section, we give details about the methods used in the three blocks of the local processing set.

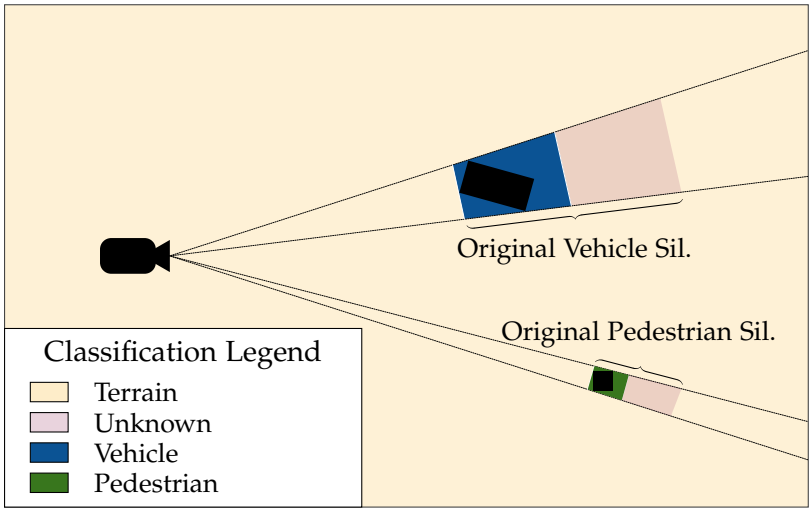
3.3.1 *Inverse Projection*

As detailed in chapter 2, to find the position of the users in the scene from the 2D bounding boxes, we can do an inverse projection of the bounding boxes on the ground. Indeed, the two bottom points of the 2D bounding box correspond approximately to the two closest points on the ground of the 3D bounding box, as shown in Fig. 3.2. The top two points of the bounding box, when they can be projected to the ground, give an upper limit to the span occupied by a user.

However, we observe that the silhouettes projected on the ground are much larger than the span of the vehicle, especially on the axis of the depth relative to the cameras. We can therefore reduce this effect by assigning a maximum length on the depth axis according to the class. In this case, we chose a length of 6 m for vehicles and 1 m for pedestrians. The part of the original silhouette being trimmed will be considered as hidden and therefore in an unknown state. This strategy is notably illustrated in the Fig. 3.3.



(a) Side View displaying the case where the originally projected silhouette is cropped to reduce its size to an acceptable dimension. The removed part of the silhouette is considered hidden by the object observed and thus classified as unknown.



(b) Top view displays the original silhouettes for a vehicle and a pedestrian and the cropped ones. In the case of a pedestrian, the dimensions to crop a silhouette are smaller than for the vehicles.

Figure 3.3: The rays of the bounding boxes are projected onto the ground. If the silhouette is too large, it is reduced along its length. The areas resulting from the reduction are considered as unknown since they are occluded.

3.3.2 *Basic Assignment*

The task of putting the label grids into a compatible format for merging belongs to the BxA block. If this block creates a semantic occupancy for Bayesian-based merging method, it is then called BPA. However, if it uses a mass system used in DST to create an evidential grid, it is then called BBA.

3.3.2.1 *Classes*

In our work, we used three semantic classes namely: pedestrian, vehicle and terrain as given in Equation (3.1),

$$\Omega = \{\mathcal{V}, \mathcal{P}, \mathcal{T}\} \tag{3.1}$$

where $\mathcal{V}$ is the vehicle class, $\mathcal{P}$ the pedestrian class, and $\mathcal{T}$ the terrain class. Ω represents the available universe of classes. In addition, there is an internal state that we use to define whether a cell has been observed or not. This will be treated differently depending on the merge mode, Bayesian or evidential.

3.3.2.2 *Occupancy Grids*

In the case where we want to merge the different PoV by a Bayesian method, we can then transform the previously generated grid into a semantic occupancy grid.

GRID DEFINITION This type of grid has already been defined, as in [76] where the authors propose an augmentation of the classical occupancy grid by appending the presumed class to the occupancy value. Nevertheless, this format is not suitable for grid fusion. Therefore, we chose the format presented in [37] which, for each position, proposes $|\Omega|$ sub-cells, containing the probability of each class. We will note this map $\mathcal{B}_{<x,y,c>}$ with $< x, y >$ the coordinates of the cell, c the index of the sub-cell (one per class).

BASIC PROBABILITY ASSIGNMENT FUNCTION The probability value assignment is done based on the observed cell label and the detection confusion estimate. This task is here called the function BPA and can be defined according to Equation (3.2).

$$BPA : M_{<x,y>} \rightarrow \mathcal{M}_{<x,y,z>},\ z \in \Omega \tag{3.2}$$

To perform this task, we use a lookup table that allows us to know the probability value of each class for each observed label. Table 3.1 shows the Look-Up Table (LUT) used for observations from vehicles. In this example, when a vehicle has been detected, we estimate the fact that it is really a vehicle at 85 %, that it is finally a pedestrian at 10 % and that it is a land at 5 %. Table 3.2 shows the LUT used for observation from the infrastructure.

Obs.	$\mathcal{V}$	$\mathcal{P}$	$\mathcal{T}$
X	0.33	0.33	0.33
$\mathcal{V}$	1.00	0.00	0.00
$\mathcal{P}$	0.00	1.00	0.00
$\mathcal{T}$	0.20	0.20	0.60

Table 3.1: LUT to assign probability values to each sub-cell based on the observed class of the original cell when observed from a vehicle. X stands for unobserved cases.

Obs.	$\mathcal{V}$	$\mathcal{P}$	$\mathcal{T}$
X	0.33	0.33	0.33
$\mathcal{V}$	1.00	0.00	0.00
$\mathcal{P}$	0.00	1.00	0.00
$\mathcal{T}$	0.00	0.00	1.00

Table 3.2: LUT to assign probability values to each sub-cell based on the observed class of the original cell when observed from the infrastructure. X stands for unobserved cases.

3.3.2.3 *Evidential Grids*

To perform a merge in the framework of the DST, it is necessary to create evidential maps.

GRID DEFINITION The definition of the semantic evidential grid is almost identical to that presented in section 2.3.3.4. However, in chapter 2, we considered only two classes, $\mathcal{O}$ and $\mathcal{F}$, whereas we now have three: $\mathcal{V}$, $\mathcal{P}$ and $\mathcal{T}$. Thus, each sub-cell takes the value of the mass associated to each element of the power-set 2^Ω which is defined in Equation (3.3). We also notice that we have chosen only three distinct classes but that more classes could be used in future works. Similarly, we will note this map $\mathcal{E}_{<x,y,c>}$ with $< x, y >$ the coordinates of the cell, c the index of the sub-cell (one per element of the power set).

$$\begin{aligned} 2^\Omega = \{&\varnothing, \{\mathcal{V}\}, \{\mathcal{P}\}, \{\mathcal{T}\}, \\ &\{\{\mathcal{V}\}, \{\mathcal{P}\}\}, \{\{\mathcal{V}\}, \{\mathcal{T}\}\}, \{\{\mathcal{P}\}, \{\mathcal{T}\}\}, \Omega\} \end{aligned} \tag{3.3}$$

Once again, the advantage of using a power set is that we can take into account states of doubt, this time: between classes. For instance, in the case where a motorcycle is seen from the front, it could be classified as a pedestrian while seen from the side it would be more easily classified as a vehicle. It is thus possible to compute specifically the confusion factor between these two classes to assign a mass value to the set $\{\{\mathcal{V}\}, \{\mathcal{P}\}\}$. Similarly, if a cell has not been observed, we will only consider the set Ω.

Obs.	$\varnothing$	$\{\mathcal{V}\}$	$\{\mathcal{P}\}$	$\{\mathcal{T}\}$	$\Omega\backslash$ $\{T\}$	$\Omega\backslash$ $\{P\}$	$\Omega\backslash$ $\{V\}$	Ω
X	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
$\mathcal{V}$	0.00	0.30	0.00	0.00	0.10	0.10	0.00	0.50
$\mathcal{P}$	0.00	0.00	0.30	0.00	0.10	0.10	0.00	0.50
$\mathcal{T}$	0.00	0.10	0.10	0.30	0.00	0.00	0.00	0.50

Table 3.3: LUT to assign mass values to each sub-cell from the observed class of the original cell when observed from a vehicle. X stands for unobserved cases.

Obs.	$\varnothing$	$\{\mathcal{V}\}$	$\{\mathcal{P}\}$	$\{\mathcal{T}\}$	$\Omega\backslash$ $\{T\}$	$\Omega\backslash$ $\{P\}$	$\Omega\backslash$ $\{V\}$	Ω
X	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
$\mathcal{V}$	0.00	0.40	0.00	0.00	0.00	0.00	0.00	0.60
$\mathcal{P}$	0.00	0.00	0.40	0.00	0.00	0.00	0.00	0.60
$\mathcal{T}$	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.60

Table 3.4: LUT to assign mass values to each sub-cell from the observed class of the original cell when observed from the infrastructure. X stands for unobserved cases.

BASIC BELIEF ASSIGNMENT FUNCTION In the same way as for the function of BPA, the function of BBA allows to determine values for each of the masses of the power set and can be formalized in the form of Equation (3.4):

$$BBA : M_{<x,y>} \rightarrow \mathcal{M}_{<x,y,z>},\ z \in 2^{\Omega} \tag{3.4}$$

Similarly to the occupancy grids, we also use a LUT similar to that of Table 3.3 to assign values to the masses of the power set depending on the observation of the cell when observed from a vehicle. We observe, however, that when a cell has not been observed, the mass of Ω is assigned to 1 to account for this state of unknown, unlike the BPA function. Table 3.4 shows the LUT used for observations from the infrastructure.

Today, the functions of BBA still form contributions since no method is yet agreed upon. Thus, we have defined the values of our LUT with a heuristic method using qualitative and quantitative studies. Another good indicator is the conflict value used in the Dempster fusion rule given in Equation (3.9) which should be minimal. However, since these values are influenced by the performances of the agents' classifiers, but also by their pose, their number or by the layout of the terrain in the scene, it is necessary to frequently reevaluate the LUT's values.

3.4 MERGING METHODS

In this section, we come back to the merger block by detailing its functioning and the different approaches evaluated. We firstly describe the method based on the Bayesian theory before the method based on the DST one.

3.4.1 *Bayes-Based Merging*

The Bayesian merge rule is very similar to the one presented in section 2.3.3.1 except that this rule is to be applied for each of the sub-cells $\mathcal{B}_{<x,y,c>i}$ with $c \in \Omega$. Thus, the merge rule is updated to match the one in Equation (3.5), i being the index of the agent, and knowing that the cell contains the probability of the presence of its associated label.

$$\begin{aligned} &\forall x \in [0,m],\ y \in [0,n], c \in \Omega \\ &\mathcal{M}_{\mathcal{B}<x,y,c>} = \prod_{i}^{N} \mathcal{B}_{<x,y,c>i} \end{aligned} \tag{3.5}$$

Therefore, for each subcell at a given $< x, y, c >$ coordinates, we can finally merge the observations by successive multiplications. Nevertheless, this method does not handle observation conflicts.

3.4.2 *Evidential Merging*

A method based on DST as used in [19, 90] provides a better understanding of conflicting observations. Several combination rules are available.

3.4.2.1 *Conjonctive's Combination Rule*

The first combination rule, called the conjunctive combination rule, is defined by Equation (3.6),

$$m_1(A) \textcircled{\cap} m_2(A) = \sum_{B \cap C = A \in 2^{\Omega}} m_1(B) m_2(C) \tag{3.6}$$

where m_1 and m_2 are mass functions defined over the universe Ω. Since the combination rule is associative, we can apply it to the $\mathcal{E}_{<x,y,c>}$ maps of each of the N agents to form a global evidential grid $\mathcal{M}_{\mathcal{E}<x,y,c>}$ according to Equation (3.7):

$$\begin{aligned} &\forall x \in [0,m],\ y \in [0,n], c \in \Omega \\ &\mathcal{M}_{\mathcal{E}<x,y,c>} = \bigcirc\!\!\!\!\!\bigcap_{i=0}^{N} \mathcal{E}_{<x,y,c>i} \end{aligned} \tag{3.7}$$

Following the association of the local grids, a global grid is obtained with the particularity of having $m(\varnothing) \neq 0$ in some cells. This value is generated by conflicts between the different agents observing the same cell. Several interpretations of the conflict are possible [67] such as the

non-exhaustiveness of the discernment framework (lack of available classes), lack of reliability in the observations or bad modeling of the perception capacities (BBA). Therefore it can be a good indication of the weaknesses of our modeling of the scene and of the perception that we will try to correct in order to reduce the conflict. Nevertheless, it is sometimes impossible to reduce this conflict and it will have to be managed either by coefficients of collapse in the BBA or in the phase of combination. However, a possible solution to reduce the conflict is the use of a slump coefficient on the BBA. This coefficient allows, depending on certain parameters, to assign a higher value to the Ω mass. For the moment, in the current state of our work, the closest thing to the use of such a coefficient is the reduction of the silhouettes according to their label. Another approach would be to implement this coefficient according to the distance of the objects from the PoV and the pose noise of the PoV.

3.4.2.2 *Dempster's Combination Rule*

To handle the conflict in the combination phase, we can add a normalization factor to the conjunctive combination rule to form the Dempster combination rule which we used in section 2.3.3.4. This is formalized in Equation (3.8),

$$m_1(A) \oplus m_2(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \varnothing} m_1(B) m_2(C) \tag{3.8}$$

where K, defined in (3.9) gives the conflict value.

$$K = \sum_{B \cap C = \varnothing} m_1(B) m_2(C) \tag{3.9}$$

Thus, using Dempster's combination rule, the conflict is distributed among all masses, but respects $m(\varnothing) = 0$, a property that must be respected in the closed world proposed by Shafer.

As stated in section 2.3.3.4, this rule is associative. Hence, it is possible to create a map from N observing agents providing local evidential grids $\mathcal{E}_{<x,y,c>}$ in order to obtain a global evidential one $\mathcal{M}_{\mathcal{E}<x,y,c>}$ according to Equation (3.10).

$$\begin{aligned} &\forall x \in [0,m],\ y \in [0,n], c \in \Omega \\ &\mathcal{M}_{\mathcal{E}<x,y,c>} = \bigoplus_{i=0}^{N} \mathcal{E}_{<x,y,c>i} \end{aligned} \tag{3.10}$$

At this point, we have either a semantic occupancy grid $\mathcal{M}_{\mathcal{B}<x,y,c>}$ or an evidential semantic grid $\mathcal{M}_{\mathcal{E}<x,y,c>}$. These maps contain the information for each class, but it is necessary to interpret them to obtain a semantic grid representing the scene.

3.5 DECISION METHODS

In this section we discuss the decision making block. We formalize the method to transform an occupancy grid or an evidential grid into a semantic grid.

3.5.1 *From Occupancy Grids To Semantic Grids*

As defined in Section 3.3.2.2, the occupancy grid $\mathcal{M}_{\mathcal{B}<x,y,c>}$ consists of subcells containing the probability associated with each label. Thus, it is possible to transform the occupancy grid into the semantic grid $\mathcal{S}_{<x,y>}$ by selecting the label with the highest probability as formalized in equation (3.11).

$$\begin{aligned} &\forall x, y \in [0, m], [0, n] \\ &\mathcal{S}_{<x,y>} = \underset{c \in \Omega}{\operatorname{argmax}} \, \mathcal{M}_{\mathcal{B}<x,y,c>} \end{aligned} \tag{3.11}$$

$\mathcal{S}_{<x,y>}$ thus consists, for each location cell $< x, y >$, of the label with the maximum estimated probability.

3.5.2 *From Evidential Grids To Semantic Grids*

To make a decision, several approaches are possible.

3.5.2.1 *Using masses*

Similar to the approach presented in the section 3.5.1, one solution is to choose the label with the largest mass as formalized in equation (3.12).

$$\begin{aligned} &\forall x, y \in [0, m], [0, n] \\ &\mathcal{S}_{<x,y>} = \underset{c \in \Omega}{\operatorname{argmax}} \, \mathcal{M}_{\mathcal{E}<x,y,c>} \end{aligned} \tag{3.12}$$

Nevertheless, this solution does not take into account situations where observations was contradictory since the labels of $2^\Omega \setminus A, \, \forall A \in \Omega$ and thus, may bring less good performances than if all the elements of 2^Ω were taken into account.

3.5.2.2 *Belief and Plausibility*

In order to take into account all the elements of 2^Ω, we can use an approaches based on belief (*bel*) or plausibility (*pl*) functions. They are defined in equation (3.13).

$$\begin{aligned} &\forall A \in \Omega \\ &bel(A) = \sum_{B|B \subseteq A} m(B) \\ &pl(A) = \sum_{B|B \cap A \neq \varnothing} m(B) \end{aligned} \tag{3.13}$$

These belief and plausibility functions use the values of the masses to give an interval, equation (3.14), in which lies the estimated probability value for a label $A \in \Omega$.

$$bel(A) \leq P(A) \leq pl(A) \tag{3.14}$$

It is therefore possible to generate semantic maps from these functions. We will then have a believed semantic grid $\mathcal{S}_{bel<x,y>}$ or a plausible semantic grid $\mathcal{S}_{pl<x,y>}$ as defined in equation (3.15).

$$\begin{aligned} &\forall x, y \in [0, m], [0, n] \\ &\forall c \in 2^{\Omega},\ m(c) = \mathcal{M}_{\mathcal{E}<x,y,c>} \\ &\mathcal{S}_{bel<x,y>} = \underset{C \in \Omega}{\operatorname{argmax}}\ bel(C) \\ &\mathcal{S}_{pl<x,y>} = \underset{C \in \Omega}{\operatorname{argmax}}\ pl(C) \end{aligned} \tag{3.15}$$

3.5.2.3 *Probability Estimation*

It is also possible to create a map from an estimated probability, either from the belief and plausibility functions or from the masses.

BELIEF INTERVAL Since the probability of a label $A \in \Omega$ is framed by belief and plausibility, equation (3.14), we can set the probability as being in the center of these two bounds, as given in equation (3.16).

$$P_{est}(A) \sim \frac{pl(A) - bel(A)}{2} + bel(A) \tag{3.16}$$

It is therefore possible, in the same way as in the section 3.5.1, to create a semantic map from this estimated probability, as in equation (3.17).

$$\begin{aligned} &\forall x, y \in [0, m], [0, n] \\ &\forall c \in 2^{\Omega},\ m(c) = \mathcal{M}_{\mathcal{E}<x,y,c>} \\ &\mathcal{S}_{P_{est}<x,y>} = \underset{C \in \Omega}{\operatorname{argmax}}\ P_{est}(C) \end{aligned} \tag{3.17}$$

PIGNISTIC PROBABILITY Finally, it is possible to determine the pignistic probability noted $BetP$ of a label $A \in \Omega$ using equation (3.18).

$$BetP(A) = \sum_{\varnothing \neq B \subseteq \Omega} \frac{m(B)}{1 - m(\varnothing)} \frac{|A \cap B|}{|B|},\ \forall A \subseteq \Omega \tag{3.18}$$

The advantage of calculating the pignistic probability resides in its consideration of the conflict estimation, defined in the section 3.4.2.1, in the decision-making.

The method to define the map is based on the maximum pignistic probability among the elements of Ω, such as (3.19).

$$\begin{aligned} &\forall x, y \in [0, m], [0, n] \\ &\forall c \in 2^{\Omega},\ m(c) = \mathcal{M}_{\mathcal{E}<x,y,c>} \\ &\mathcal{S}_{BetP<x,y>} = \underset{C \in \Omega}{\operatorname{argmax}}\ BetP(C) \end{aligned} \tag{3.19}$$

3.6 RESULTS

In this section, we evaluate our approach. We first present the data we used for our evaluation as well as the metrics allowing a quantitative evaluation. Finally, we discuss the performance of our cooperative semantic map creation approach.

3.6.1 *Datasets*

In order to evaluate our algorithm, it is necessary to put it in situation which is possible via the use of datasets. In [90], the authors based their evaluation on the KITTI dataset [45]. Nevertheless, the KITTI dataset is not a cooperative dataset and, to the best of our knowledge, no cooperative dataset was available.

3.6.1.1 *CARLA*

For the same reasons as those discussed in Section 2.4.1, namely synchronization and pose estimation challenges, and to address the limitations of the dataset we created previously, we created a set of new datasets using CARLA [32]. However, we noticed in parallel to our work that other teams have also realized cooperative datasets based on CARLA. This is notably the case of the authors of [118] who propose OPV2V, a cooperative dataset to test V2V approaches. Nevertheless, our approach also requires views from infrastructures. This latter doesn't, though, include cooperation with an infrastructure that we do use in our approach. Even more recently, authors of [72] propose the V2X-Sim dataset with viewpoints coming from vehicles and infrastructure. Nevertheless, we need more scenarios than the ones proposed in V2X-Sim and especially a scenario in roundabouts where occlusion occurs. That is why we have created our own cooperative dataset.

3.6.1.2 *Our Datasets*

Since the previous dataset created in chapter 2 was featuring a small amount of agents and, most importantly, was only featuring vehicles, we made new datasets with PoV from vehicles and trackside infrastructure in order to provide an extensive validation of our approach. The actors are vehicles and pedestrians using the CARLA autopilot. The figure 3.4 shows a trackside infrastructure PoV with several actors in the scene.

To observe different behaviors, we generated several datasets with different traffic density at a roundabout and another dataset at a crossroads. Our goal is to test the performance of our approach in several situations where there may be occlusions or confusion among agents. We can augment our dataset by enabling or disabling agents. By default, all vehicles are considered agents and provide a stream of images. It is therefore possible to ignore image streams to simulate unconnected vehicles.

Figure 3.4: Image from an infrastructure PoV of our dataset with a dense traffic in a roundabout generated with CARLA [32].

SENSORS AND RECORDED DATA Similar to what we used in chapter 2 to generate our dataset, each PoV consists of a set of cameras. This set of sensors consists of two cameras:

- RGB camera of 1384 × 1032 pixels and a Field of View (FOV) of 90°
- Semantic camera of 1384 × 1032 pixels and a FOV of 90°

These two cameras share the same pose, the same optical parameters and are perfectly synchronized with their images, the position of the sensors and the position of the bounding boxes of the actors are also recorded for each simulation step. This allow a perfect correspondence pixel by pixel in a PoV to filter the final 2D bounding boxes.

Indeed, the 2D bounding boxes are computed on the agent images from the 3D bounding boxes and the camera position of the ego-agent. Some of the 2D bounding boxes should not appear because occluded by other objects. We use the semantic segmented images associated with the RGB images to figure out the ratio of correct label within a bounding box in order to define if the object is occluded and the bounding box erased. A shortcoming of this solution is that objects occluded by a same-label object are not erased as visible in Fig. 3.5. Finally, an adjustable noise can be added to the retained bounding boxes.

ROUNDABOUT ORIGINAL, MEDIUM & DENSE The first set of datasets is located at the same roundabout as the dataset we did in Chapter 2. In fact, the original dataset is almost the same as the one used in Chapter 2 but updated to match the method of generating bounding boxes taking into account occlusions. The other two datasets are then described in Table 3.5.

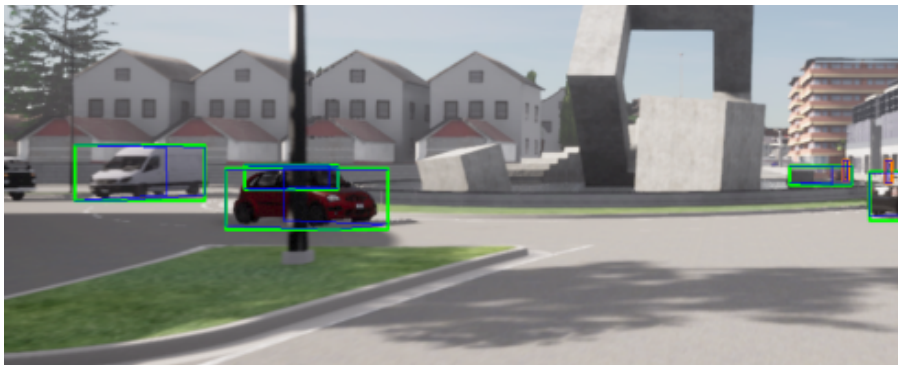


Figure 3.5: Example of a bounding box that should be deleted but is notbecause it belongs to the same class as the occluding object.

Table 3.5: Original dataset with different traffic density in the roundabout.

Dataset	N# Infrastructure	N# Vehicle	N# Pedestrian
Original	1	3	0
Medium	6	6	12
Dense	6	30	6

The vehicles are then placed on the branches of the roundabout as well as in the ring in the initial state and the pedestrians are positioned on the sidewalks. Then, the autopilot provided with CARLA takes the control of the vehicles and pedestrians to make them evolve in the scene. Each dataset has a duration of 450 frames at a rate of 30 frames per second, except for the Medium dataset with 1800 frames.

CROSSROADS Although the framework of this thesis is primarily aimed at roundabout navigation, we also generated an intersection dataset to evaluate other types of occlusions such as buildings at the corner of two streets like the one in Figure 3.6. This dataset also provides a large number of occlusions due to the terrain, notably on the pedestrians, and allows to counterbalance the limits evoked in earlier.

Moreover, while the only roundabout provided in the CARLA maps is positioned in $< 0, 0 >$ coordinate, this dataset allowed us to validate the functioning of our approach outside the particular case of the roundabout. This has notably highlighted the problem of the center of the map. Thus, the center of the map is defined by the barycenter of the position of the PoVs of the infrastructure as guessable in Figure 3.7.

This dataset consists of the elements represented in Table 3.6. The four PoVs of the infrastructure are placed in such a way that their overlapping zone is limited to the intersection of the roads but that, for each of the branches, we count on the presence of vehicles to complete the map. Thus, we wish to observe if this limited number of PoVs is sufficient to monitor an intersection.

Figure 3.6: Image from an infrastructure PoV of our dataset at a crossroad generated with CARLA [32].

Table 3.6: Original dataset with traffic density at a roundabout.

Dataset	N# Infrastructure	N# Vehicle	N# Pedestrian
Crossroads	4	18	20

3.6.1.3 *Ground Truth Generation*

In order to evaluate the performance of our solution, it is necessary to have a reference. In our case, the ground truth takes the form of a semantic map. Since CARLA does not provide a map, it is necessary to generate this map from the available information. Thus, we use the bounding boxes of the different agents that we place on a grid with the same format as the map that we want to generate with our approach as illustrated in Fig. 3.7.

This semantic grid map can be transformed into an occupancy grid by considering the cells corresponding to the terrain as free cells and the others as occupied cells.

3.6.2 *Qualitative Study*

To ensure that our system could generate coherent and usable map, we conducted a qualitative study. We also used this qualitative assessment to roughly adjust the parameters used in the BBA and BPA. Fig. 3.8 illustrates a visual comparison between the ground truth Fig. 3.8b, the map generated using the DST Fig. 3.8a, and the map generated using a Bayesian fusion-based approach Fig. 3.8c. The Bayesian theory-based method succeeds in placing all vehicles on the map, as does the DST based method. However, the method based on the DST seems to have

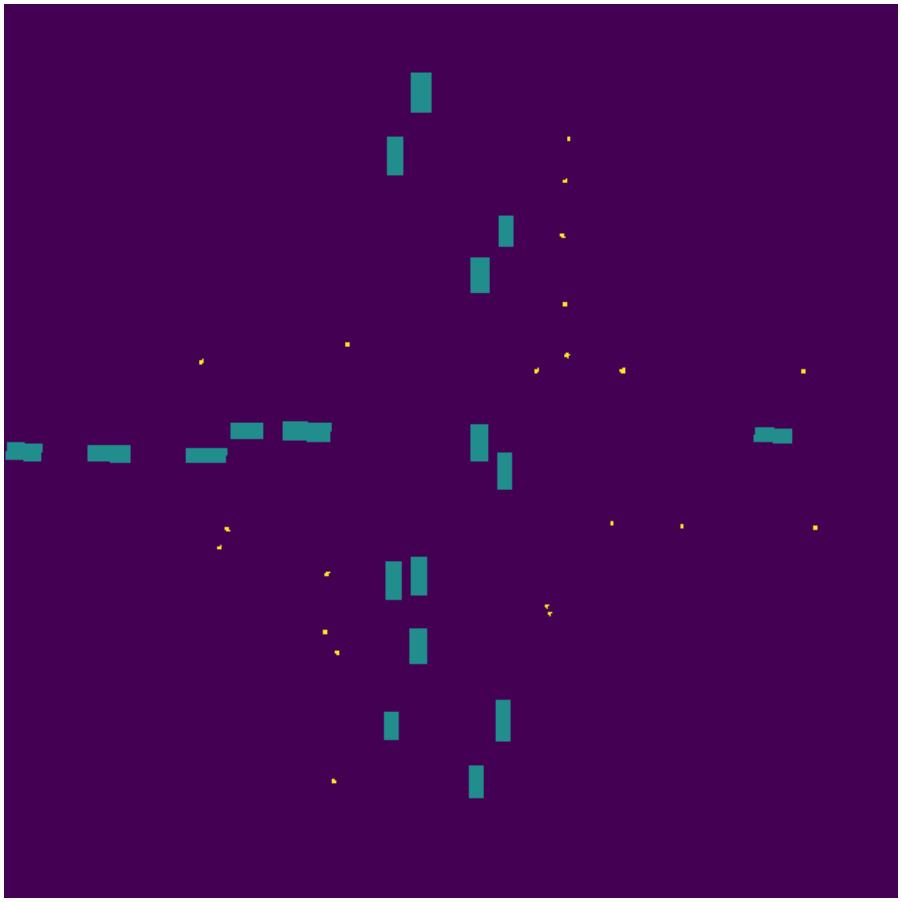
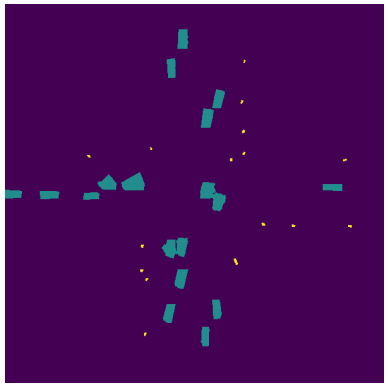


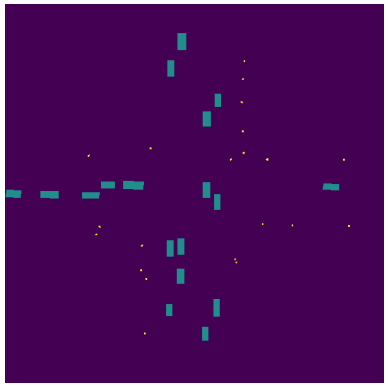
Figure 3.7: Ground truth map generated from bounding box information provided by CARLA. In purple: terrain cells, in yellow: pedestrians and in turquoise: vehicles.

less false positives. As for the pedestrian, they are mostly correctly placed on the map.

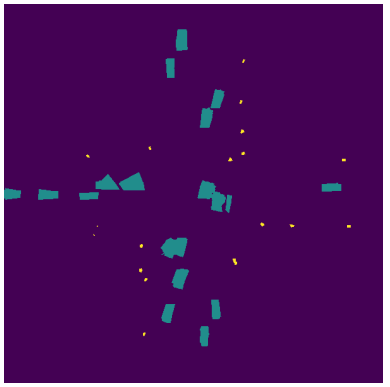
In order to compare our results, we tried to reconstruct a scene using a multiview approach as proposed in COLMAP [95]. However, despite the fact that we gave the true positions of the cameras in the a priori, more margin of error to the algorithm or pairs of initial images: we could not get any results. The reason is the lack of common features between the elements. We believe that the baselines between the images are also too large to perform an association and reconstruction. To the best of our knowledge, we do not know of any other method to create an occupancy grid from images and without depth information from multiple viewpoints with freely available code. Our approach, pragmatic and efficient, does not require object matching among the different PoV and thus functions regardless of the appearance of the objects in the scene.



(a) Semantic map generated by our approach using DST merging.



(b) Ground thruth map.



(c) Semantic map generated by our approach using Bayes-based merging.

Figure 3.8: Comparison of the ground truth maps and the semantic map generated by our solution. In purple: ground cells, in yellow: pedestrians and in turquoise vehicles.

3.6.3 *Metrics*

To provide a quantitative study, we used several metrics commonly found in the literature which we used in Chapter 2, namely Intersection over Union (IoU) and F1-score. IoU and F1-score measure the performance on the size and the detection of objects which we adapted to fit the semantic aspect of our new maps. We also used the Correct Ratio (CR) to measure the semantic performance.

3.6.3.1 *Intersection over Union*

The IoU is based on the number of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). In this chapter, they are generated by comparing the cells of the ground truth map with the obtained semantic map. In this case, for a label $\omega \in \Omega$, if a cell of the semantic map obtained is equal to ω and that it is the same on the cell of the same position on the ground truth map, then this cell is regarded as a TP. If a cell of the obtained semantic map is equal to ω but it is not equal to ω on the cell of same position on the ground truth map, then this cell is considered as an FP. If a cell of the obtained semantic map is not equal to ω but is equal to ω on the cell of the same position on the ground truth map, then this cell is considered as an FN. Finally, if a cell of the obtained semantic map is not equal to ω and it is not equal to ω on the cell of same position on the ground truth map, then this cell is considered as an TN. Thus, the IoU for a chosen ω label is given by equation (3.20).

$$IoU_\omega = \frac{TP_\omega}{TP_\omega + FP_\omega + FN_\omega} \tag{3.20}$$

To estimate the overall performance, it is possible to calculate the average between all labels, as given in equation (3.21).

$$mIoU = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} IoU_\omega \tag{3.21}$$

However, we have noted a limitation of the average score. When the detection gives a failure rate of 100 % and, therefore, the whole map is considered as terrain, the default label, then the average IoU is about 30 %. Another solution is to transform the semantic grids into an occupancy grid and to compute the IoU on the occupancy rather than on the labels.

3.6.3.2 *F1-Score*

The F1-Score is very similar to the IoU since it is also based on the number of TP, TN, FP and FN. It can be calculated as shown in equation (3.22).

$$F1_\omega = \frac{TP_\omega}{TP_\omega + \frac{FP_\omega + FN_\omega}{2}} \tag{3.22}$$

In the same way as for the average F1-score, it is possible to obtain an overall value by calculating the average F1-score, as in equation (3.23). It should be noted that the average F1-score shares the same shortcomings as the average IoU.

$$mF1 = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} F1_\omega \tag{3.23}$$

3.6.3.3 *Correct Ratio*

In order to measure the performance on assigning correct labels to cells, we used the CR which we calculated as shown in equation (3.24).

$$CR_\omega = \frac{TP_\omega + TN_\omega}{TP_\omega + TN_\omega + FN_\omega + FP_\omega} \tag{3.24}$$

Usually, the CR is calculated by comparing corresponding label cells on the ground truth map and the final map devided by the total number of cell in the map. However, the cells having a correct label are constituted by the sum of TP and TN. The limitation of this metric in our use case is that the majority of the cells are considered as terrain in the map to be evaluated and in the ground truth map. Therefore, the results are always very high and it is difficult to distinguish the variations.

3.6.4 *Quantitative Study*

In this section, we observe our approach in terms of several parameters using the metrics designated above.

3.6.4.1 *Bayes-based Method vs. DST-based Method*

Since we have tried two approaches, one based on Bayesian theory and the other based on the DST, we want to highlight which approach is the most efficient. The Table 3.7 aims precisely at showing the performance differences between the two approaches on a scene of the dataset we created while showing the result for each of the metrics stated earlier and can be used as a reference point for the rest of this chapter.

The results highlighted in the Table 3.7 express an average improvement of 22.42 % on the average IoU or 19.21 % on the mean F1-Score in favor of the DST based approach. Vehicles benefit the most from this approach with a gain on the IoU of 92.87 %. Pedestrians also benefit from a better representation on the map when using DST. However, we notice that the pedestrian IoU is low compared to the other classes. This is due to the fact that the areas of the cells are significant compared to the areas occupied by pedestrians. Thus, the number of cells occupied by pedestrians is low and artificially increases the impact of errors in the metrics. Conversely, for the terrain class which occupies the majority of the cells of the map and on which the impact of errors is particularly low in the metrics.

Fusion	Class	IoU	F1-Score	CR
Ours (Bayes)	$\mathcal{V}$	26.21	36.46	96.50
	$\mathcal{P}$	22.33	41.52	99.95
	$\mathcal{T}$	96.40	98.17	96.45
	Mean	48.31	58.71	N/A
Ours (DST)	$\mathcal{V}$	50.55	67.07	98.87
	$\mathcal{P}$	28.03	43.49	99.98
	$\mathcal{T}$	98.84	99.41	98.85
	Mean	59.14	69.99	N/A

Table 3.7: Detail of the IoU, F1-Score and CR (in %) for the heavy traffic scene (roundabout) in our dataset.

3.6.4.2 *Decision taking methods*

In the previous paragraph, we compared our fusion methods based on the Pignistic probability decision making *BetP* defined in Section 3.5.2.3. In fact, it is this decision making method that is used in the remainder of this section to compare different items. However, in Section 3.5, we presented several decision making methods associated with DST. However, as Table 3.8 shows, when we vary the decision making method, we get strictly identical results. To observe differences between these methods for estimating probability, we need to look at the maps generated before each cell is assigned a class. In particular, Figure 3.9 depicts some maps showing that the probabilities vary as a whole, but that their ranking between labels remains the same regardless of the method. Therefore, when a cell is given a label due to the fact that its associated probability is maximal, the differences between the decision making methods are lost.

This observation is valid on all the datasets we have created and we could not obtain a situation in which an observation remains ambiguous until the decision is made. Perhaps the increase of more confusing classes or semantic noise in the observations could have an impact on the decision making. Similarly, another method to associate a class with a cell could offer other results.

Finally, we observe in Figure 3.9e and in Figure 3.9f that the decision making method *BetP* generates the same values regardless of whether the conjunctive or disjunctive merge rule is used. Indeed, this behavior is expected since this decision making method normalizes its result from the conflict value. This conflict is non-zero after merging via the conjunctive rule, unlike Dempster's combination rule.

Combining Rule	Method	Classes $\mathcal{V}$	$\mathcal{P}$	$\mathcal{T}$	Mean
Conjunctive	Max. m	57.33	35.62	99.13	64.03
	Max. bel	57.33	35.62	99.13	64.03
	Max. pl	57.33	35.62	99.13	64.03
	Max. $BetP$	57.33	35.62	99.13	64.03
	Max. P_{est}	57.33	35.62	99.13	64.03
	Max. wP_{est}	57.33	35.62	99.13	64.03
Dempster	Max. m	57.33	35.62	99.13	64.03
	Max. bel	57.33	35.62	99.13	64.03
	Max. pl	57.33	35.62	99.13	64.03
	Max. $BetP$	57.33	35.62	99.13	64.03
	Max. P_{est}	57.33	35.62	99.13	64.03
	Max. wP_{est}	57.33	35.62	99.13	64.03

Table 3.8: Evolution of the mIoU for both conjunctive and Dempster combing rule and for each decision taking methods. Every value are identical because of the assignment method of a class for each cell based on the maximum of probability.

3.6.4.3 *Connected Vehicles Ratio Evolution in a Scene*

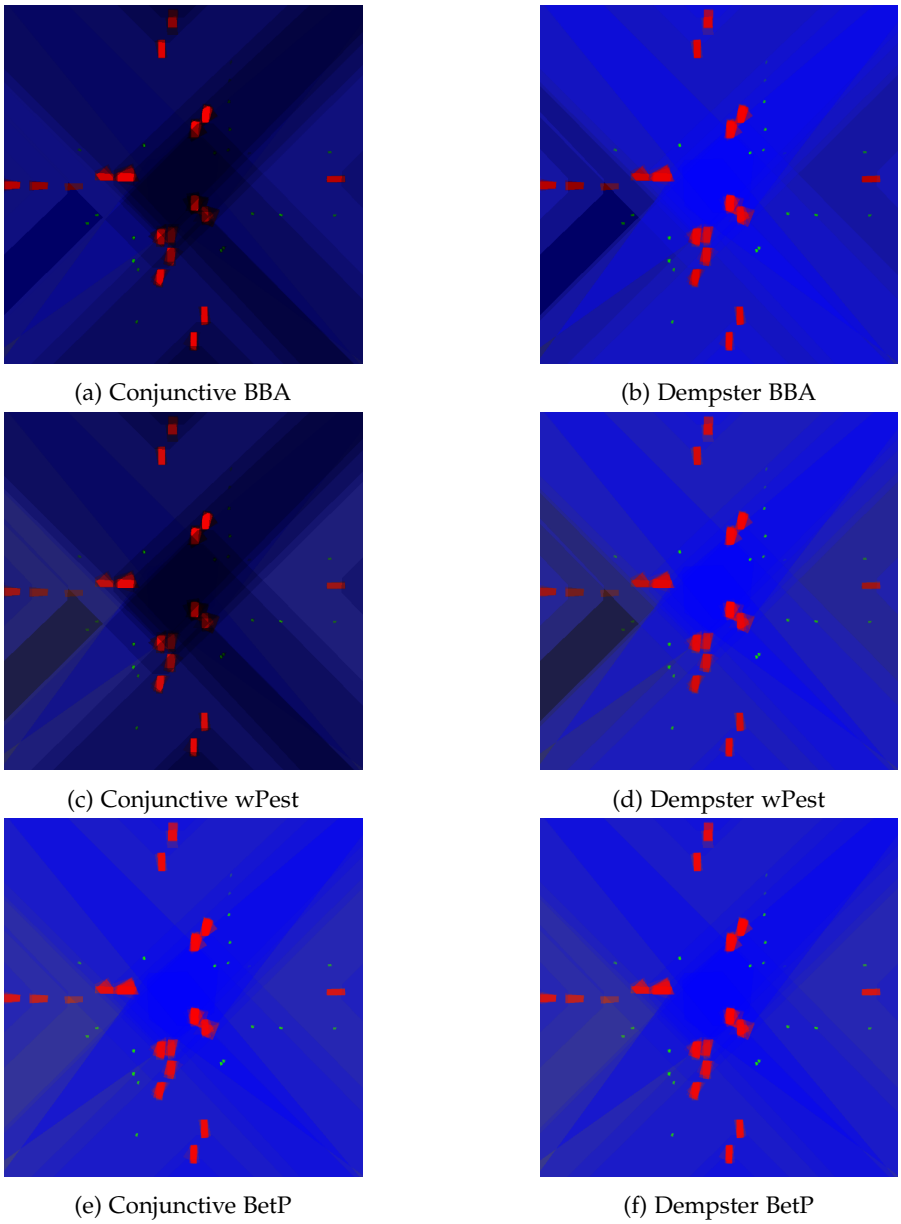
Now that we have seen the performance between the two approaches of our solution, we can test, on the same scene of our dataset, to vary the proportion of CV and PoV of the infrastructure. We therefore performed several sub-scenarios. The first one consists of a single vehicle observing the scene, as an instrumented vehicle. The second scenario consists of an infrastructure alone in the manner of projects like [4]. A third scenario is to have the infrastructure with only 1 CV corresponding to the approach of MEC-View[1]. Finally, other scenarios are created by changing the proportion of CV up to the all connected.

The Table 3.9 shows that the approach based on the Bayesian theory maintains a IoU of 50 % and seems to suffer from the multiplication of the points of view whereas the approach based on the DST benefits from the multiplication of the points of views. Indeed, in the scene of dense traffic in a roundabout, occlusions are frequent and can produce conflicting observations between the agents. However, the approach based on the DST manages the conflicting observations and thus shows its advantage in such scenarios, contrary to the approach based on the Bayesian theory. Thus, as the number of PoV increases,

1 http://www.mec-view.de/

Infrastructure: N# PoV	Connected Vehicles	Ours (Bayes) mIoU (%)	Ours (DST) mIoU (%)	Gain (%)
0 (0 %)	1 (3 %)	53.22	53.28	0.11
2 (33 %)	0 (0 %)	54.43	55.69	2.31
	1 (3 %)	**54.85**	56.00	2.10
	8 (27 %)	51.97	56.77	9.24
	15 (50 %)	50.54	**57.64**	14.05
	23 (77 %)	50.30	57.58	14.47
	30 (100 %)	49.87	57.29	**14.88**
6 (100 %)	0 (0 %)	50.67	58.41	15.28
	1 (3 %)	**51.24**	58.53	14.23
	8 (27 %)	49.91	59.23	18.67
	15 (50 %)	48.74	**60.12**	**23.35**
	23 (77 %)	48.58	59.87	23.24
Full dataset (6 Infra. PoV + 30 CVs)		48.31	59.14	22.42

Table 3.9: Evolution of the mIoU (in %) for the dense traffic scene (roundabout) of our dataset, varying the proportion of agents in the users fleet.



(a) Conjunctive BBA

(b) Dempster BBA

(c) Conjunctive wPest

(d) Dempster wPest

(e) Conjunctive BetP

(f) Dempster BetP

Figure 3.9: Probability maps before assigning a label to each cell. The lighter, the greater the probability. We can note that with the conjunctive combination rule, the images are darker because of the absence of normalization with conflictual observations except with the *BetP* which performs such normalization. Blue stands for the terrain, red for the vehicles, and green for the pedestrians.

the gap between the DST approach and the Bayesian approach widens, up to a maximum of 23.35 % of mIoU gain.

We also observe that with an infrastructure reduced to the strict minimum and a fleet with a proportion of about 50 % of mCV, it is possible to generate a map with good results. This observation is therefore encouraging in the transition that we will see until we have 100 % of CV instrumented on the roads.

Number of agents		4	12	36
Ours (Bayes)	$\mathcal{V}$	36.82	32.10	26.21
	$\mathcal{P}$	N/A	26.70	22.33
	$\mathcal{T}$	99.39	98.98	96.40
	Mean	45.40	52.59	48.31
Ours(DST)	$\mathcal{V}$	42.21	53.99	50.55
	$\mathcal{P}$	N/A	32.57	28.03
	$\mathcal{T}$	99.55	99.59	98.84
	Mean	47.25	**62.05**	59.14
Gain (%)	Mean	4.07	18.00	**22.42**

Table 3.10: Comparison of the IoU varying the number of vehicles in the roundabout (in %).

3.6.4.4 *Traffic Density Evolution*

Finally, another important variable at intersections is traffic density. Indeed, the denser the traffic is, the more the phenomenon of occlusions is accentuated and the more difficult the scene is to understand and map. We have three scenarios with varying the number of agents at the same roundabout as shown in the Table. 3.10.

As showed in Table 3.9, Table 3.10 points out that the more observers there are, the larger the gap between the DST based approach and the Bayesian theory based method. However, we also observe that even the DST approach is affected by the complexity of the scene due to occlusions and conflicting observations.

Nonetheless, we observe that the values of mIoU are fair and that our solution provides usable maps regardless of the traffic density in the scene.

3.7 CONCLUSION

In this chapter, we presented a new method to generate semantic grids from sparse and light information coming from both vehicle's embedded sensors and roadside infrastructure sensors. This approach is designed to be highly cooperative and exploits the in-scene PoV of the vehicles to refine the generated map.

Our approach succeeded to generate maps regardless of the appearance of the objects from the multiple PoV and overtook other state-of-the-art tools such as COLMAP [95] which was unable to bring results due to the limitations of its algorithms based on depth and 3D reconstruction.

The method we have presented is based on two approaches: one based on Bayesian theory and the other on the DST. We have tested the

performance of our approach on a dataset composed of several scenes, generated with CARLA [32] and provided an extensive validation with various amount of CV and traffic density. These results also highlighted the resilience of the approach based on DST in case of conflicting observations.

CONCLUSION & PERSPECTIVES

In this thesis, we have addressed onboard/offboard cooperative perception for autonomous navigation.

In Chapter 1, we have reviewed the state of the art of cooperative perception methods. In particular, we have discussed cooperative perception approaches by detailing what constitutes them as well as the challenges that these approaches raise. We also noticed that cooperative localization is a very active topic as well as map generation, contrasting with efforts in cooperative object detection and tracking. We also noticed a pattern absent in cooperative approaches: full vehicle-infrastructure cooperation. Finally, we noted that cooperative perception is currently a very active topic with projects in this area, but we also noted a lack of cooperative datasets featuring instrumented vehicles and infrastructure.

In Chapter 2, we implemented the missing cooperative scheme. We also developed a method to generate occupancy grids only from camera data and a fundamentally cooperative but pragmatic and efficient approach. We tested two methods to perform information fusion: one based on Bayesian theory, the one closest to the state-of-the-art methods, and the other based on Dempster-Shafer theory. We also generated a dataset in order to validate our approach and to demonstrate that the Dempster-Shafer based methods offer better results.

In Chapter 3, we completed our initial idea by adding the semantic aspect. We adapted the existing parts and added a decision making block. We generated a set of new datasets in order to conduct an extensive validation of our approach. The qualitative study we conducted shows that our approach is fully capable of generating a semantic map from sparse camera data while the state of the art methods implemented in COLMAP fail to give results. Finally, the quantitative study shows that our decision making methods improve the results when the fusion is performed with the method based on Bayesian theory but that the method based on Dempster-Shafer theory always gives better results, no matter the conditions tested.

The work that has been done during this thesis opens new perspectives that have not been explored yet. To develop this approach, it is essential to obtain a set of data sets in which instrumented vehicles evolve, themselves evolving in an area monitored by infrastructure. Indeed, today, the approach proposed in this thesis has no real other work to compare with.

Although Chapter 2 has provided a solution for the management of sensor noise, it remains rudimentary. A real study on the modeling of the noise of sensors laying with each other that could impact the creation of local grids or the BPA and BBA function seems necessary to complete this work. The management of the synchronization and the impact on the data network also needs to be studied, especially if

other data are added to those already shared. Network failures sould also be explored since they could lead to erroneous data or imply huge delay and desynchronization.

In order to focus on the study of cooperative map generation, we have excluded the step of obtaining bounding boxes. Although the task of detecting vehicles and pedestrians (or even other objects) is taken for granted, it would be interesting to test the robustness of our approach to the detection distance, the class noise and the impact of weather conditions. Furthermore, these bounding boxes could be augmented by adding information such as an estimate of the size of the vehicle based on its type, an estimate of the center of the vehicle (which is not necessarily the center of the bounding box) or an estimate of the 3D bounding boxes. We could also imagine multi-modal systems embedded on each agent based on vision such as camera-LiDAR or camera-RADAR systems to better estimate 3D bounding boxes. At the system level, we could imagine verification agents embedding other types of sensors such as hyperspectral cameras to facilitate classification or event-based cameras to interpolate intermediate states between two real map updates. Finally, on the RGB camera only architecture, we could use 2D silhouettes from pixel segmentation images instead of bounding boxes in order to obtain more realistic silhouettes after the reprojection on the ground. In the latter case, the use of RGBD camera could enable the use of finite rays, helping the silhouette estimation on the ground. The management and the integration of these optional information in the information fusion would be an interesting contribution.

BIBLIOGRAPHY

[1] Rozh Abdulmajed and RamazanAlpay ABBAK. "Accuracy comparison between GPS only and GPS plus GLONASS in RTK and static methods." PhD thesis. Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, 2017.

[2] Farhan Ahammed, Javid Taheri, Albert Y Zomaya, and Max Ott. "Vloci: Using distance measurements to improve the accuracy of location coordinates in gps-equipped vanets." In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer. 2010, pp. 149–161.

[3] Lina Altoaimy and Imad Mahgoub. "Fuzzy logic based localization for vehicular ad hoc networks." In: *IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*. IEEE. 2014, pp. 121–128.

[4] Dhiraj Gulati Annkathrin Krämmer* Christoph Schöller* and Alois Knoll. "Providentia - A Large Scale Sensing System for the Assistance of Autonomous Vehicles." In: *Robotics: Science and Systems (RSS), Workshop on Scene and Situation Understanding for Autonomous Driving*. 2019. URL: https://sites.google.com/view/uad2019/accepted-posters.

[5] Shintaro Arai, Yasutaka Shiraki, Takaya Yamazato, Hiraku Okada, Toshiaki Fujii, and Tomohiro Yendo. "Multiple LED arrays acquisition for image-sensor-based I2V-VLC using block matching." In: *IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE. 2014, pp. 605–610.

[6] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. "A survey on 3d object detection methods for autonomous driving applications." In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2019), pp. 3782–3795.

[7] GSM Association et al. *Cellular-Vehicle to Everything (C-V2X)[Internet]*.

[8] Philipp Bender, Julius Ziegler, and Christoph Stiller. "Lanelets: Efficient map representation for autonomous driving." In: *IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 420–425.

[9] Keni Bernardin and Rainer Stiefelhagen. "Evaluating multiple object tracking performance: the clear mot metrics." In: *EURASIP Journal on Image and Video Processing* 2008 (2008), pp. 1–10.

[10] Marcin Bernas, Bartłomiej Płaczek, Wojciech Korski, Piotr Loska, Jarosław Smyła, and Piotr Szymała. "A survey and comparison of low-cost sensing technologies for road traffic monitoring." In: *Sensors* 18.10 (2018), p. 3243. URL: https://www.mdpi.com/1424-8220/18/10/3243/pdf.

[11] David Bétaille and Rafael Toledo-Moreo. "Creating enhanced maps for lane-level vehicle navigation." In: *IEEE Transactions on Intelligent Transportation Systems* 11.4 (2010), pp. 786–798.

[12] BE Bilgin and VC Gungor. "Performance comparison of IEEE 802.11 p and IEEE 802.11 b for vehicle-to-vehicle communications in highway, rural, and urban areas." In: *International Journal of Vehicular Technology* 2013 (2013).

[13] Keshav Bimbraw. "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology." In: *12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Vol. 1. IEEE. 2015, pp. 191–198.

[14] Andreas Birk and Stefano Carpin. "Merging occupancy grid maps from multiple robots." In: *Proceedings of the IEEE* 94.7 (2006), pp. 1384–1397.

[15] Chunjuan Bo, Huchuan Lu, and Dong Wang. "Weighted generalized nearest neighbor for hyperspectral image classification." In: *IEEE Access* 5 (2017), pp. 1496–1509.

[16] Bosch. *Conduite automatisée : comment les voitures et les infrastructures communiquent en milieu urbain*. Website. Accessed 2020-07-29. July 2020. URL: https://www.bosch.fr/actualites/2020/conduite%5C%2Dautomatisee%5C%2Dcomment%5C%2Dles%5C%2Dvoitures%5C%2Det%5C%2Dles%5C%2Dinfrastructures%5C%2Dcommuniquent%5C%2Den%5C%2Dmilieu%5C%2Durbain/.

[17] Gary Bradski and Adrian Kaehler. "OpenCV." In: *Dr. Dobb's journal of software tools* 3 (2000), p. 2.

[18] Alberto Broggi, Pietro Cerri, Mirko Felisa, Maria Chiara Laghi, Luca Mazzei, and Pier Paolo Porta. "The VisLab Intercontinental Autonomous Challenge: an extensive test for a platoon of intelligent vehicles." In: *International Journal of Vehicle Autonomous Systems* 10.3 (2012), pp. 147–164. URL: https://www.inderscienceonline.com/doi/pdf/10.1504/IJVAS.2012.051250.

[19] Federico Camarda, Franck Davoine, and Véronique Cherfaoui. "Fusion of evidential occupancy grids for cooperative perception." In: *2018 13th Annual Conference on System of Systems Engineering (SoSE)*. June 2018, pp. 284–290. DOI: 10.1109/SYSOSE.2018.8428723.

[20] Pietro E Carnelli, Mahesh Sooriyabandara, and R Eddie Wilson. "Large-Scale VANET Simulations and Performance Analysis using Real Taxi Trace and City Map Data." In: *IEEE Vehicular Networking Conference (VNC)*. IEEE. 2018, pp. 1–8.

[21] Juan Castorena and Siddharth Agarwal. "Ground-edge-based LIDAR localization without a reflectivity calibration for autonomous driving." In: *IEEE Robotics and Automation Letters* 3.1 (2017), pp. 344–351.

[22] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds." In: *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. 2019, pp. 88–100.

[23] Qi Chen, Sihai Tang, Qing Yang, and Song Fu. "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds." In: *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2019, pp. 514–524.

[24] Shengyang Chen, Ching-Wei Chang, and Chih-Yung Wen. "Perception in the Dark—Development of a ToF Visual Inertial Odometry System." In: *Sensors* 20.5 (2020), p. 1263. DOI: https://doi.org/10.3390/s20051263.

[25] Xiaobo Chen, Jianyu Ji, and Yanjun Wang. "Robust Cooperative Multi-Vehicle Tracking with Inaccurate Self-Localization Based on On-Board Sensors and Inter-Vehicle Communication." In: *Sensors* 20.11 (2020). Publisher: Multidisciplinary Digital Publishing Institute, p. 3212.

[26] Xiaobo Chen, Jianyu Ji, and Yanjun Wang. "Robust Cooperative Multi-Vehicle Tracking with Inaccurate Self-Localization Based on On-Board Sensors and Inter-Vehicle Communication." In: *Sensors* 20.11 (2020), p. 3212.

[27] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. "Multi-view 3d object detection network for autonomous driving." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1907–1915.

[28] Matthew Cornick, Jeffrey Koechling, Byron Stanley, and Beijia Zhang. "Localizing ground penetrating radar: A step toward robust autonomous ground vehicle localization." In: *Journal of field robotics* 33.1 (2016), pp. 82–102.

[29] Sokemi Rene Emmanuel Datondji, Yohan Dupuis, Peggy Subirats, and Pascal Vasseur. "A survey of vision-based traffic monitoring of road intersections." In: *IEEE transactions on intelligent transportation systems* 17.10 (2016), pp. 2681–2698.

[30] Mickaël Delamare, Remi Boutteau, Xavier Savatier, and Nicolas Iriart. "Static and Dynamic Evaluation of an UWB Localization System for Industrial Applications." In: *Sci* 2.1 (2020), p. 7.

[31] Murat Dikmen and Catherine M Burns. "Autonomous driving in the real world: Experiences with tesla autopilot and summon." In: *Proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications*. 2016, pp. 225–228.

[32] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An Open Urban Driving Simulator." In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[33] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An Open Urban Driving Simulator." In: *Proceedings of the 1st Annual Conference on Robot Learning*. Ed. by Sergey Levine, Vincent Vanhoucke, and Ken Goldberg. Vol. 78. Proceedings of Machine Learning Research. PMLR, Nov. 2017, pp. 1–16. URL: http://proceedings.mlr.press/v78/dosovitskiy17a.html.

[34] Michał Drwiega. "Features Matching based Merging of 3D Maps in Multi-Robot Systems." In: *24th International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE. 2019, pp. 663–668.

[35] Jens Einsiedler, Oliver Sawade, Bernd Schäufele, Marcus Witzke, and Ilja Radusch. "Indoor micro navigation utilizing local infrastructure-based positioning." In: *IEEE Intelligent Vehicles Symposium*. IEEE. 2012, pp. 993–998. URL: https://ieeexplore.ieee.org/abstract/document/6232262.

[36] Cristofer Englund, Lei Chen, Jeroen Ploeg, Elham Semsar-Kazerooni, Alexey Voronov, Hoai Hoang Bengtsson, and Jonas Didoff. "The Grand Cooperative Driving Challenge 2016: boosting the introduction of cooperative automated vehicles." In: *IEEE Wireless Communications* 23.4 (2016), pp. 146–152. DOI: 10.1109/MWC.2016.7553038.

[37] Özgür Erkent, Christian Wolf, and Christian Laugier. "Semantic Grid Estimation with Occupancy Grids and Semantic Segmentation Networks." In: *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. Nov. 2018, pp. 1051–1056. DOI: 10.1109/ICARCV.2018.8581180. URL: https://hal.inria.fr/hal-01933939/document.

[38] TR ETSI. "102 862 V1. 1.1 (2011-12) Intelligent Transport Systems (ITS)." In: *Performance Evaluation of Self-Organizing TDMA as Medium Access Control Method Applied to ITS* (2011).

[39] TR ETSI. *102 863 (V1. 1.1): Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and Guidance on Standardization*. Tech. rep. Technical report, ETSI, 2011.

[40] Andreas Festag. "Cooperative intelligent transport systems standards in Europe." In: *IEEE communications magazine* 52.12 (2014), pp. 166–172.

[41] J.-S. Franco and E. Boyer. "Fusion of multiview silhouette cues using a space occupancy grid." In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. ISSN: 2380-7504. Oct. 2005, 1747–1753 Vol. 2. DOI: 10.1109/ICCV.2005.105.

[42] Michael Gabb, Holger Digel, Tobias Müller, and Rüdiger-Walter Henn. "Infrastructure-supported Perception and Track-level Fusion using Edge Computing." In: *IEEE Intelligent Vehicles*

Symposium (IV). 2019, pp. 1739–1745. DOI: 10.1109/IVS.2019.8813886.

[43] Chao Gao, Guorong Zhao, and Hassen Fourati. *Cooperative Localization and Navigation: Theory, Research, and Practice*. CRC Press, 2019.

[44] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T Barron. "Learning single camera depth estimation using dual-pixels." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7628–7637.

[45] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets Robotics: The KITTI Dataset." In: *International Journal of Robotics Research (IJRR)* (2013).

[46] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3354–3361.

[47] Farouk Ghallabi. "Precise self-localization of autonomous vehicles using lidar sensors and highly accurate digital maps on highway roads." PhD thesis. Université Paris sciences et lettres, 2020.

[48] FORTISS GMBH. *Providentia*. ONLINE : http://testfeld-a9.de/. Accessed 15.05.2020. URL: http://testfeld-a9.de/.

[49] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. "A survey of deep learning techniques for autonomous driving." In: *Journal of Field Robotics* 37.3 (2020), pp. 362–386.

[50] Dhiraj Gulati, Feihu Zhang, Daniel Clarke, and Alois Knoll. "Vehicle infrastructure cooperative localization using factor graphs." In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 1085–1090.

[51] Dhiraj Gulati, Feihu Zhang, Daniel Malovetz, Daniel Clarke, Gereon Hinz, and Alois Knoll. "Graph based vehicle infrastructure cooperative localization." In: *20th International Conference on Information Fusion (Fusion)*. IEEE. 2017, pp. 1–6.

[52] Praveen Gunturi, Nuri W Emanetoglu, and David E Kotecki. "A 250-Mb/s data rate IR-UWB transmitter using current-reused technique." In: *IEEE Transactions on Microwave Theory and Techniques* 65.11 (2017), pp. 4255–4265.

[53] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision 2nd ed., 4th print*. 2006.

[54] O Hassan, I Adly, and KA Shehata. "Vehicle localization system based on ir-uwb for v2i applications." In: *8th International Conference on Computer Engineering & Systems (ICCES)*. IEEE. 2013, pp. 133–137.

[55] Elwan Héry, Philippe Xu, and Philippe Bonnifait. "Pose and covariance matrix propagation issues in cooperative localization with LiDAR perception." In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 1219–1224.

[56] Gereon Hinz, Martin Buechel, Frederik Diehl, Malte Schellmann, and Alois Knoll. "Designing a far-reaching view for highway traffic scenarios with 5G-based intelligent infrastructure." In: *8. Tagung Fahrerassistenz*. 2017. URL: https://mediatum.ub.tum.de/doc/1421303/file.pdf.

[57] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees." In: *Autonomous Robots* (2013). Software available at http://octomap.github.com. DOI: 10.1007/s10514-012-9321-0. URL: http://octomap.github.com.

[58] C. Huang and X. Wu. "Cooperative Vehicle Tracking using Particle Filter Integrated with Interacting Multiple Models." In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 2019, pp. 1–6.

[59] Ruben Jungnickel, Michael Köhler, and Franz Korf. "Efficient automotive grid maps using a sensor ray based refinement process." In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 668–675.

[60] John B Kenney. "Dedicated short-range communications (DSRC) standards in the United States." In: *Proceedings of the IEEE* 99.7 (2011), pp. 1162–1182.

[61] Roozbeh Kianfar, Bruno Augusto, Alireza Ebadighajari, Usman Hakeem, Josef Nilsson, Ali Raza, Reza S Tabar, Naga VishnuKanth Irukulapati, Cristofer Englund, Paolo Falcone, et al. "Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge." In: *IEEE transactions on intelligent transportation systems* 13.3 (2012), pp. 994–1007.

[62] Nam-Seog Kim and Jan M Rabaey. "A high data-rate energy-efficient triple-channel UWB-based cognitive radio." In: *IEEE Journal of Solid-State Circuits* 51.4 (2016), pp. 809–820.

[63] Seong-Woo Kim, Zhuang Jie Chong, Baoxing Qin, Xiaotong Shen, Zhuoqi Cheng, Wei Liu, and Marcelo H Ang. "Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 5059–5066.

[64] Seong-Woo Kim, Baoxing Qin, Zhuang Jie Chong, Xiaotong Shen, Wei Liu, Marcelo H Ang, Emilio Frazzoli, and Daniela Rus. "Multivehicle cooperative driving using cooperative perception: Design and experimental validation." In: *IEEE Transac-*

tions on Intelligent Transportation Systems 16.2 (2014), pp. 663–680.

[65] Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis Mccullough, and Alexandros Mouzakitis. "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications." In: *IEEE Internet of Things Journal* 5.2 (2018), pp. 829–846.

[66] Louis Lecrosnier, Rémi Boutteau, Pascal Vasseur, Xavier Savatier, and Friedrich Fraundorfer. "Camera pose estimation based on PnL with a known vertical direction." In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3852–3859.

[67] Eric Lefèvre. "Fonctions de croyance: de la théorie à la pratique." PhD thesis. 2012.

[68] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. "Map-based precision vehicle localization in urban environments." In: *Robotics: science and systems*. Vol. 4. Citeseer. Citeseer. 2007, p. 1.

[69] Jesse Levinson and Sebastian Thrun. "Robust vehicle localization in urban environments using probabilistic maps." In: *IEEE International Conference on Robotics and Automation*. IEEE. 2010, pp. 4372–4378.

[70] Chuanxiang Li, Bin Dai, and Tao Wu. "Vision-based precision vehicle localization in urban environments." In: *Chinese Automation Congress*. IEEE. 2013, pp. 599–604.

[71] Ruihao Li, Sen Wang, and Dongbing Gu. "DeepSLAM: A Robust Monocular SLAM System with Unsupervised Deep Learning." In: *IEEE Transactions on Industrial Electronics* (2020).

[72] Yiming Li, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. "V2X-Sim: A Virtual Collaborative Perception Dataset for Autonomous Driving." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021.

[73] Zongdian Li, Tao Yu, Ryuichi Fukatsu, Gia Khanh Tran, and Kei Sakaguchi. "Proof-of-Concept of a SDN Based mmWave V2X Network for Safe Automated Driving." In: *IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–6.

[74] Guiqiu Liao, Jiankang Zhao, Chao Cui, Haihui Long, Jianbin Zhu, and Achraf Djerida. "Dynamic Attitude Estimation Improvement for Low-cost MEMS IMU by Integrating Low-cost GPS." In: *arXiv preprint arXiv:2008.10469* (2020).

[75] Wei Liu and Yozo Shoji. "Edge-assisted vehicle mobility prediction to support V2X communications." In: *IEEE Transactions on Vehicular Technology* 68.10 (2019), pp. 10227–10238.

[76] Chenyang Lu, Marinus Jacobus Gerardus van de Molengraft, and Gijs Dubbelman. "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks." In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 445–452.

[77] Bin Lv, Hao Xu, Jianqing Wu, Yuan Tian, Yongsheng Zhang, Yichen Zheng, Changwei Yuan, and Sheng Tian. "LiDAR-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities." In: *IEEE Access* 7 (2019), pp. 79895–79907.

[78] Ehsan Emad Marvasti, Arash Raftari, Amir Emad Marvasti, Yaser P Fallah, Rui Guo, and HongSheng Lu. "Cooperative lidar object detection via feature sharing in deep networks." In: *arXiv preprint arXiv:2002.08440* (2020).

[79] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. "Correction de nuages de points lidar embarqué sur véhicule pour la reconstruction d'environnement 3D vaste." In: *Reconnaissance de Formes et Intelligence Artificielle (RFIA)*. Clermont-Ferrand, France, June 2016. URL: https://hal.archives-ouvertes.fr/hal-01906323.

[80] Pierre Merriaux, Romain Rossi, Rémi Boutteau, Vincent Vauchey, Lei Qin, Pailin Chanuc, Florent Rigaud, Florent Roger, Benoit Decoux, and Xavier Savatier. "The VIKINGS Autonomous Inspection Robot: Competing in the ARGOS Challenge." In: *IEEE Robotics & Automation Magazine* 26.1 (2018), pp. 21–34.

[81] Aaron Miller, Kyungzun Rim, Parth Chopra, Paritosh Kelkar, and Maxim Likhachev. "Cooperative Perception and Localization for Cooperative Driving." In: *IEEE International Conference on Robotics and Automation*. July 2020.

[82] Marius Minea. "Cellular—Sensorless V2I—based traffic information and communications infrastructure: Case study for high class motorways." In: *9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE. 2017, pp. 1–6.

[83] Yong Niu, Yong Li, Depeng Jin, Li Su, and Athanasios V Vasilakos. "A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges." In: *Wireless networks* 21.8 (2015), pp. 2657–2676.

[84] Agnieszka Ochalek, Witold Niewiem, Edyta Puniach, and Pawel Ćwikakala. "Accuracy evaluation of real-time GNSS precision positioning with RTX Trimble technology." In: *Civil and environmental engineering reports* (2018).

[85] Kirtan Gopal Panda, Deepak Agrawal, Arcade Nshimiyimana, and Ashraf Hossain. "Effects of environment on accuracy of ultrasonic sensor operates in millimetre range." In: *Perspectives in Science* 8 (2016), pp. 574–576.

[86] José A del Peral-Rosado, José A López-Salcedo, Sunwoo Kim, and Gonzalo Seco-Granados. "Feasibility study of 5G-based localization for assisted driving." In: *International Conference on Localization and GNSS (ICL-GNSS)*. IEEE. 2016, pp. 1–6.

[87] Fabian de Ponte Müller. "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles." In: *Sensors* 17.2 (2017), p. 271.

[88] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. "ROS: an open-source Robot Operating System." In: *ICRA workshop on open source software*. Vol. 3. Kobe, Japan. 2009, p. 5.

[89] Craig Quiter and Maik Ernst. *deepdrive/deepdrive: 2.0*. 2018.

[90] Sven Richter, Yiqun Wang, Johannes Beck, Sascha Wirges, and Christoph Stiller. "Semantic Evidential Grid Mapping Using Monocular and Stereo Cameras." In: *Sensors* 21.10 (2021). ISSN: 1424-8220. DOI: 10.3390/s21103380. URL: https://www.mdpi.com/1424-8220/21/10/3380.

[91] Mohsen Rohani, Denis Gingras, Vincent Vigneron, and Dominique Gruyer. "A new decentralized Bayesian approach for cooperative vehicle localization based on fusion of GPS and VANET based inter-vehicle distance measurement." In: *IEEE Intelligent transportation systems magazine* 7.2 (2015), pp. 85–95.

[92] Guodong Rong et al. "LGSVL Simulator: A High Fidelity Simulator for Autonomous Driving." In: *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294422.

[93] Seyed Mohammad-Sajad SADOUGH. "A tutorial on ultra wideband modulation and detection schemes." In: *Shahid Beheshti University, Faculty of Electrical and Computer Eng.* (2009).

[94] Davide Scaramuzza and Friedrich Fraundorfer. "Visual odometry [tutorial]." In: *IEEE robotics & automation magazine* 18.4 (2011), pp. 80–92.

[95] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. "Pixelwise View Selection for Unstructured Multi-View Stereo." In: *European Conference on Computer Vision (ECCV)*. 2016.

[96] THRUN Sebastian, Burgard Wolfram, and Fox Dieter. *Probabilistic robotics*. 2005.

[97] Kari Sentz, Scott Ferson, et al. *Combination of evidence in Dempster-Shafer theory*. Vol. 4015. Sandia National Laboratories Albuquerque, 2002. URL: https://www.osti.gov/servlets/purl/800792.

[98] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles." In: *Field and Service Robotics*. 2017. eprint: arXiv:1705.05065. URL: https://arxiv.org/abs/1705.05065.

[99] Yongyue Shi, Xiao-Hong Peng, and Guangwei Bai. "Cooperative V2X for Cluster-Based Vehicular Networks." In: *International Journal on Advances in Networks and Services Volume 12, Number 3 & 4, 2019* (2019).

[100] V Shreyas, Skanda N Bharadwaj, S Srinidhi, KU Ankith, and AB Rajendra. "Self-driving Cars: An Overview of Various Autonomous Driving Systems." In: *Advances in Data and Information Sciences*. Springer, 2020, pp. 361–371.

[101] Isaac Skog and Peter Handel. "In-car positioning and navigation technologies—A survey." In: *IEEE Transactions on Intelligent Transportation Systems* 10.1 (2009), pp. 4–21.

[102] Mario Soilán, Ana Sánchez-Rodríguez, Pablo del Río-Barral, Carlos Perez-Collazo, Pedro Arias, and Belén Riveiro. "Review of laser scanning technologies and their applications for road and railway infrastructure monitoring." In: *Infrastructures* 4.4 (2019), p. 58.

[103] Elias Strigel, Daniel Meissner, Florian Seeliger, Benjamin Wilking, and Klaus Dietmayer. "The ko-per intersection laserscanner and video dataset." In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2014, pp. 1900–1901.

[104] Han-Shue Tan and Jihua Huang. "DGPS-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints." In: *IEEE Transactions on Intelligent Transportation Systems* 7.4 (2006), pp. 415–428.

[105] Sebastian Thrun. "Probabilistic robotics." In: *Communications of the ACM* 45.3 (2002), pp. 52–57.

[106] Sebastian Thrun. "Learning occupancy grid maps with forward sensor models." In: *Autonomous robots* 15.2 (2003), pp. 111–127.

[107] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. "Stanley: The robot that won the DARPA Grand Challenge." In: *Journal of field Robotics* 23.9 (2006), pp. 661–692.

[108] Chris Urmson, J Andrew Bagnell, Christopher Baker, Martial Hebert, Alonzo Kelly, Raj Rajkumar, Paul E Rybski, Sebastian Scherer, Reid Simmons, Sanjiv Singh, et al. *Tartan racing: A multi-modal approach to the darpa urban challenge*. 2007.

[109] J-S Botero Valencia, M Rico Garcia, and J-P Villegas Ceballos. "A simple method to estimate the trajectory of a low cost mobile robotic platform using an IMU." In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 11.4 (2017), pp. 823–828.

[110] Milos Vasic. *Cooperative Perception Algorithms for Networked Intelligent Vehicles*. Tech. rep. EPFL, 2017.

[111] Damien Vivet, Franck Gérossier, Paul Checchin, Laurent Trassoudaine, and Roland Chapuis. "Mobile ground-based radar sensor for localization and mapping: An evaluation of two approaches." In: *International Journal of Advanced Robotic Systems* 10.8 (2013), p. 307.

[112] Lei Wang, Javier Fernandez, Jon Burgett, Richard W. Conners, and Yilu Liu. "An evaluation of network time protocol for clock synchronization in wide area measurements." In: *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*. 2008, pp. 1–5. DOI: 10.1109/PES.2008.4596234.

[113] Yunlong Wang, Ying Wu, and Yuan Shen. "Cooperative Tracking by Multi-Agent Systems Using Signals of Opportunity." In: *IEEE Transactions on Communications* 68.1 (2020). Conference Name: IEEE Transactions on Communications, pp. 93–105. ISSN: 1558-0857. DOI: 10.1109/TCOMM.2019.2944605.

[114] Erik Ward and John Folkesson. "Vehicle localization with low cost radar sensors." In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 864–870.

[115] James Ward, Stewart Worrall, Gabriel Agamennoni, and Eduardo Nebot. "The warrigal dataset: Multi-vehicle trajectories and v2v communications." In: *IEEE Intelligent Transportation Systems Magazine* 6.3 (2014), pp. 109–117.

[116] Jianqing Wu, Hao Xu, and Wei Liu. "Points Registration for Roadside LiDAR Sensors." In: *Transportation Research Record* (2019), p. 0361198119843855.

[117] Renbo Xia, Maobang Hu, Jibin Zhao, Songlin Chen, Yueling Chen, and ShengPeng Fu. "Global calibration of non-overlapping cameras: State of the art." In: *Optik* 158 (2018), pp. 951–961.

[118] Hao Xu Runsheng anpignistiqued Xiang, Xin Xia, Xu Han, Jinlong Liu, and Jiaqi Ma. "OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication." In: *arXiv preprint arXiv:2109.07644* (2021).

[119] Philippe Xu, Gérald Dherbomez, Elwan Héry, Abderrahmen Abidli, and Philippe Bonnifait. "System architecture of a driverless electric car in the grand cooperative driving challenge." In: *IEEE Intelligent Transportation Systems Magazine* 10.1 (2018), pp. 47–59. URL: https://hal.archives-ouvertes.fr/hal-01703415/document.

[120] Bowen Yang, Haiwei Dong, and Abdulmotaleb El Saddik. "Development of a self-calibrated motion capture system by nonlinear trilateration of multiple kinects v2." In: *IEEE Sensors Journal* 17.8 (2017), pp. 2481–2491.

[121] Heng Yang and Luca Carlone. "A polynomial-time solution for robust registration with extreme outlier rates." In: *arXiv preprint arXiv:1903.08588* (2019).

[122] Heng Yang, Jingnan Shi, and Luca Carlone. "TEASER: Fast and Certifiable Point Cloud Registration." In: *arXiv preprint arXiv:2001.07715* (2020).

[123] Shizhe Zang, Ming Ding, David Smith, Paul Tyler, Thierry Rakotoarivelo, and Mohamed Ali Kaafar. "The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car." In: *IEEE vehicular technology magazine* 14.2 (2019), pp. 103–111.

[124] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps." In: *arXiv preprint arXiv:1910.03088* (2019). URL: https://arxiv.org/pdf/1910.03088.pdf;.

[125] Feihu Zhang, Hauke Stähle, Guang Chen, Chao Chen Carsten Simon, Christian Buckl, and Alois Knoll. "A sensor fusion approach for localization with cumulative error elimination." In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE. 2012, pp. 1–6.

[126] Bowen Zheng, Ping Wang, Fuqiang Liu, and Chao Wang. "Cooperative Data Delivery in Sparse Cellular-VANET Networks." In: *6th International Conference on Digital Home (ICDH)*. IEEE. 2016, pp. 128–132.

PUBLICATIONS

[1] Antoine Caillot, Safa Ouerghi, Yohan Dupuis, Pascal Vasseur, and Rémi Boutteau. "Multi-Agent Cooperative Camera-Based Semantic Grid Generation." In: *UNDER REVIEW - IEEE Robotics and Automation Letters* (2022).

[2] Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Rémi Boutteau, and Yohan Dupuis. "Survey on Cooperative Perception in an Automotive Context." In: *IEEE Transactions on Intelligent Transportation Systems* (2022), pp. 1–20. ISSN: 1558-0016. DOI: 10.1109/TITS.2022.3153815. URL: https://hal.archives-ouvertes.fr/hal-03608119/document.

[3] Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Yohan Dupuis, and Rémi Boutteau. "Multi-Agent Cooperative Camera-Based Evidential Occupancy Grid Generation." In: (2022), pp. 203–209. DOI: 10.1109/ITSC55140.2022.9921855.

PERCEPTION EMBARQUÉE / DÉBARQUÉE POUR LA NAVIGATION AUTONOME.

RÉSUMÉ Avec l'arrivée de la navigation autonome, la perception de l'environnement dans lequel évoluent les véhicules est une tâche primordiale. Pour répondre à cette problématique, les véhicules se sont dotés de plus en plus de capteurs pour percevoir leur environnement. Plus récemment, nous pouvons observer l'apparition d'approches coopératives afin d'outrepasser les limitations des capteurs embarqués.

Dans cette thèse, nous faisons un état de l'art des méthodes de perceptions coopératives dans le contexte automobile. Nous y discutons des architectures fréquemment utilisées et des défis qu'elles entraînent. Nous étudions aussi les méthodes de localisation, de détection et suivis ainsi que les méthodes de cartographies coopératives avant de lister les projets et les scénarii dans lesquels la perception coopérative est utilisée aujourd'hui.

En réponse à cet état de l'art, nous avons mis au point une nouvelle architecture coopérative fusionnant les approches Véhicules-Véhicules et Véhicule-Infrastructure actuelles et basée sur l'utilisation de données issues des véhicules et des infrastructures. Cette approche nous permet dans un premier temps de générer des grilles d'occupation des objets dynamiques d'une scène en utilisant uniquement des données limitées issues des caméras. Nous ajoutons ensuite à cette approche un aspect sémantique permettant la création de grilles sémantiques. Afin de fusionner les données issues des différents points de vues, nous avons testé deux méthodes : l'une basée sur la théorie bayésienne et l'autre sur la théorie de Dempster-Shafer.

Les résultats sont obtenus à partir de jeux de données de notre conception et montrent des résultats inatteignables par les méthodes de l'état de l'art aujourd'hui ainsi qu'une supériorité de la méthode basée sur la théorie de Dempster-Shafer.

MOTS-CLÉS: Perception coopérative, grille d'occupation, grille évidentielle, carte sémantique, Théorie de l'évidence

ONBOARD / OFFBOARD EXTENDED PERCEPTION FOR AUTONOMOUS NAVIGATION.

ABSTRACT With the emergence of autonomous navigation, the perception of the environment in which vehicles evolve is a primordial task. To address this issue, vehicles have increasingly been equipped with sensors to perceive their environment. Recently, we can observe the emergence of cooperative approaches to overcome the limitations of onboard sensors.

In this thesis, we present a state-of-the-art of cooperative perception methods in the automotive context. We discuss the frequently used architectures and the challenges they entail. We also study localization, detection and tracking methods as well as cooperative mapping methods before listing the projects and scenarios in which cooperative perception is used today.

As a response to this state-of-the-art, we have developed a new cooperative architecture merging the current Vehicle-Vehicle and Vehicle-Infrastructure approaches based on the use of data from vehicles and infrastructures. This approach allows us to generate dynamic object occupancy grids of a scene using only limited camera data. We then add to this approach a semantic aspect allowing the creation of semantic grids. To merge the data from the different points of view, we evaluated two methods: one based on the Bayesian theory and the other on the Dempster-Shafer theory.

The results are obtained from datasets of our own design and show results unattainable by state-of-the-art methods today as well as a superiority of the method based on the Dempster-Shafer theory.

KEYWORDS: Cooperative perception, occupancy grid, evidential grid, semantic map, evidential theory