



HAL
open science

**Perception pour la navigation et le contrôle des robots
mobiles. Application à un système de voiturier
autonome**
Mihai Chirca

► **To cite this version:**

Mihai Chirca. Perception pour la navigation et le contrôle des robots mobiles. Application à un système de voiturier autonome. Autre. Université Blaise Pascal - Clermont-Ferrand II, 2016. Français. NNT : 2016CLF22763 . tel-01492900v3

HAL Id: tel-01492900

<https://hal.science/tel-01492900v3>

Submitted on 21 Apr 2017

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

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

N° d'ordre D.U : 2763
EDSPIC : 777

UNIVERSITÉ BLAISE PASCAL - CLERMONT II
ÉCOLE DOCTORALE DES SCIENCES POUR L'INGÉNIEUR

Thèse présentée par

Mihai CHIRCA

pour obtenir le grade de

Docteur d'Université

Spécialité : Vision pour la robotique

**Perception pour la navigation et
le contrôle des robots mobiles**
Application à un système de voiturier autonome

Préparée en collaboration entre l'Institut Pascal, IRSTEA et Renault

Soutenue publiquement le 08 / 12 / 2016 devant le jury :

M.	MICHEL	DEVY	Directeur de Recherche, LAAS, Toulouse	Président
Mme.	VÉRONIQUE	CHERFAOUI	Maître de Conférences, HDR, UTC, Compiègne	Rapporteur
M.	PASCAL	MORIN	Professeur, UPMC, Paris	Rapporteur
M.	ROLAND	LENAIN	Chargé de recherche, IRSTEA, Clermont-Ferrand	Examineur
M.	GUILLAUME	MARTIN	Docteur Ingénieur, Renault SA, Guyancourt	Examineur
M.	CHRISTOPHE	DEBAIN	Chargé de recherche, IRSTEA, Clermont-Ferrand	Invité
M.	BENOÎT	THUILOT	Maître de Conférences, Univ. Blaise Pascal, Clermont-Ferrand	Invité
M.	ROLAND	CHAPUIS	Professeur, Univ. Blaise Pascal, Clermont-Ferrand	Directeur

*Dedicată mult iubitei mele soții,
parinților, surorii mele.*

*"To see far is one thing, going there is another"
Constantin Brâncuși*

Abstract

This work covers the conception of a system capable to do automatic parking maneuvers more versatile than those already commercialized, respecting the technical definition of exteroceptive sensors limited by costs and weight. A typical use case is to set a vehicle to park autonomously in the parking lot of a home, function generally called *autonomous home valet parking*.

Taking from the existing and knowing the expected performances, a system architecture and a functional architecture were drawn. This allowed to compose an assembly of interconnected functions that participated in the creation of modular software architecture, as well as in the creation of connection interfaces with the prototype vehicle.

First, we explored the obstacle detection problem. Having a closed property system with ultrasonic sensors, we managed to build an obstacle map with a higher precision level than the build-in product. An increasing limit detection of the ultrasonic sensors was developed using the *Structure from Motion* technique. This obstacle occupancy information was exploited afterwards in order to solve the detection problem of the navigation corridors.

Second, the vehicle localization is addressed. Three localization techniques work for a continuous functioning robustness: the localization by odometry, the localization by occupancy grid map matching and the localization by comparing the current image with the images stored in a database adapted to our needs and improved by computing means.

Last, we interested in the vehicle navigation problem. We considered solved the actuator control problem for the tracking of a given trajectory and we concentrated on an admissible trajectory planning. We developed a local path planning technique for avoiding the unmapped obstacles. In order to build the trajectory we used curves of known geometry and we proved that by using clothoids and eventually two circle arches (if maximum steering angle achieved) it is therefore possible to create trajectories with continuous curves adapted to our situation. We confirmed that using an obstacle map will allow us to predict forehead the possibility to take a specific navigation corridor.

Each part of this work was validated in simulation as well as on real data, proving the pertinence of the proposed approaches for the intended application.

Keywords: self-driving car, autonomous vehicle, autonomous valet parking, parking assistance, perception, localization, mapping, navigation, trajectory, local obstacle avoidance

Résumé

Ce travail porte sur la conception d'un système capable d'effectuer des manœuvres de parking automatique plus polyvalent que ceux actuellement commercialisés, tout en conservant une définition technique des capteurs extéroceptifs limités en prix et en gabarit. Un cas d'usage typique est de permettre au véhicule de se rendre automatiquement dans la zone de garage du domicile de son propriétaire, cette fonction est classiquement appelée *voiturier autonome à domicile*.

Partant de l'existant et connaissant les performances attendues, une architecture système et une architecture fonctionnelle ont été tracées. Cela a permis de constituer un ensemble de fonctions interconnectées qui ont participé dans la création d'une architecture software modulaire ainsi que dans la création des interfaces de connexion au véhicule prototype.

Dans un premier temps, nous explorons la problématique de la détection d'obstacles. Partant d'un système propriétaire fermé de capteurs ultrason, nous avons réussi à réaliser une carte d'obstacle à un niveau de précision supérieur au produit d'origine. Une augmentation de la limite de détection des capteurs ultrason a été réalisée utilisant une technique *Structure from Motion*. Ces informations d'occupation ont été exploitées par la suite pour traiter la problématique de détection du couloir de navigation.

Dans un second temps, la fonction de localisation du véhicule est abordée. Trois techniques de localisation collaborent pour une robustesse de fonctionnement continu : la localisation odométrique, la localisation par appariement des grilles d'occupation et la localisation par appariement entre une image actuelle et une base d'images adaptée à notre besoin et améliorée en termes de temps de calcul.

Enfin, nous nous sommes intéressés à la problématique de navigation du véhicule. Nous avons considéré résolue la problématique de contrôle des actionneurs pour le suivi d'une trajectoire donnée et nous nous sommes concentrés sur la création d'une trajectoire admissible. Nous avons développé une technique de planification locale pour l'évitement d'un d'obstacles non cartographiés. Pour la construction de trajectoire nous avons utilisé des courbes à géométrie connue et avons montré qu'en utilisant trois clothoïdes et éventuellement deux arcs de cercle (si le braquage maximal est atteint) il est possible de créer des trajectoires à courbure continue adaptées à notre situation. Nous avons montré que l'utilisation d'une carte d'obstacles nous permet de prédire plus en avance de la possibilité d'emprunter un certain couloir de navigation.

Chacune des parties de ce travail a fait l'objet de validations en simulation mais aussi sur des données réelles démontrant la pertinence des approches proposées quant à l'application visée.

Mots clés : véhicule autonome, voiturier autonome, valet de parking, assistance au parking, perception, localisation, cartographie, navigation, planification locale.

Remerciements

Ce mémoire représente l'aboutissement de nombreux échanges et de rencontres enrichissantes. Je tiens à exprimer ma reconnaissance aux membres de jury pour l'intérêt qu'ils ont porté à mes travaux et les conseils qu'ils m'ont dispensés. En particulier je remercie Michel Devy d'avoir accepté de présider le jury, Véronique Cherfaoui et Pascal Morin pour avoir accepté de rapporter mes travaux.

J'adresse mes plus vifs remerciements à mon directeur de thèse Roland Chapuis et à mes co-encadrants: Guillaume Martin, Christophe Debain et Roland Lenain.

Je remercie l'ANRT qui a subventionné ma thèse dans le cadre d'un programme CIFRE rendant ainsi possible mon travail de recherche en partenariat avec *Renault*, *Institut Pascal* et *IRSTEA*. Je remercie Cédric Tessier de l'entreprise *Effidence* et Pierre Delmas et Florent Malartre de l'entreprise *4D-Virtualiz* pour leurs support et pour m'avoir fourni les outils informatique qui m'ont permis de travailler tous les jours.

Je remercie tous mes collègues de l'Institut Pascal : Benoit Thuilot, Claude Aynaud, Coralie Bernay-Angeletti, Guillaume Bresson, Clément Demier, Laurent Lequière, Morgan Slade, Ange Nizard, Laurent Malaterre, Thomas Féraud, Laurent Delobel, Alexis Wilhem, Romuald Aufrère, Sébastien Caux, Vanessa Chaudron.

J'adresse également mes meilleurs sentiments aux collègues et amis de Renault: Guillermo Pita-Gil, Nicoleta Minoiu-Enache, Philippe Saint-Loup, Djalil Ourabah, Ablamvi Ameyor, Maud Geoffriault, Guy-Michel Cloarec, Pape Abdoulaye SENE, Sofiane AMARA, Héléne Vorobieva, François Desnoyer, Jean-Marc Diou, Karim Kouki, Iris Ballesteros Tolosana, Joan Davins-Valldaura, Olivier Aubert, Renaud Deborne, Hervé Illy, Pascal Le Corre, Alexandre Armand, Lynda Halit.

Je remercie l'ensemble de mes amis qui m'ont soutenu tout au long de ces années. Je vous remercie tous ceux que j'ai cités et que je n'ai pas cités pour avoir partagé avec moi leurs idées ou simplement pour avoir passé un bon moment ensemble.

Je finis par remercier tous les membres de ma famille, en dédiant ce travail tout particulièrement à ma femme et à mes parents.

Table des matières

Abstract	iv
Résumé	v
Remerciements	vii
Abbreviations	xiii
I Introduction	1
I.1 Le monde de la robotique	2
I.2 La réponse de la robotique aux problèmes de transport	3
I.3 Le véhicule intelligent	4
I.3.1 Les enjeux	4
I.3.2 Les niveaux de délégation de conduite et l'autonomie de certaines tâches	5
I.3.3 Le long chemin vers l'automatisation du véhicule	7
I.3.4 Le problème du dernier kilomètre	14
I.4 Le voiturier autonome	16
I.5 Le cadre de travail	19
I.6 Objectif de la thèse et scénario typique	20
I.7 Organisation du mémoire	21
I.8 Contributions	22
II Perception de l'environnement	25
II.1 Problématique	26
II.2 Perception : État de l'art	27
II.2.1 Perception de l'environnement et moyens d'acquisition de l'information	27
II.2.2 Cartographie	37
II.2.3 Communication V2X	41
II.2.4 Détection et suivi d'obstacles dans le domaine routier	43
II.2.5 Détection d'obstacles pour une situation de parking	44
II.3 Moyens de perception du véhicule prototype	47
II.3.1 Perception par capteurs ultrason	48
II.3.2 Perception par caméra	55
II.3.3 Capteurs proprioceptifs	69
II.4 Bilan sur la perception	71
III Localisation	73
III.1 Problématique	74
III.2 Localisation : État de l'art	74
III.2.1 Techniques de localisation absolue	75
III.2.2 Techniques de localisation relative	78

III.3	Évaluation des techniques de localisation proposées	86
III.3.1	Localisation par odométrie	86
III.3.2	Localisation par techniques de SLAM	95
III.3.3	Localisation par appariement des grilles d'occupation	103
III.3.4	Estimation des a priori de localisation et d'orientation par techniques de comparaison d'images	114
III.4	Stratégie de localisation et bilan	123
IV	Détection d'obstacles et analyse de la zone traversable	127
IV.1	Problématique	128
IV.2	Gestion de la zone traversable : État de l'art	128
IV.2.1	Grille d'occupation	128
IV.2.2	Algorithmes du type <i>Structure from Motion</i>	131
IV.3	Localisation des obstacles détectés par les capteurs ultrason	133
IV.3.1	Calcul de la position des obstacles par une approche géométrique	133
IV.3.2	Résultats de la cartographie par le système ultrasons	135
IV.3.3	Bilan sur la perception de l'environnement utilisant le système de capteurs ultrason	142
IV.4	Enrichissement de la grille d'occupation avec approche <i>Structure from Motion</i> .	142
IV.4.1	Algorithme <i>Structure from Motion</i>	143
IV.4.2	Appariement et suivi des points	145
IV.4.3	Stratégies d'estimation du déplacement relatif de la caméra	147
IV.4.4	Mise en forme de nuages des points 3D sous forme de macro-obstacles 2D	157
IV.4.5	Résultats de la cartographie par approche <i>Structure from Motion</i>	160
IV.5	Analyse des résultats globaux et bilan	162
IV.5.1	La complémentarité des deux systèmes de perception	162
IV.5.2	Limitations	163
IV.5.3	Bilan	165
V	Navigation	167
V.1	Problématique	168
V.2	Navigation : État de l'art	169
V.2.1	Planification globale	169
V.2.2	Planification locale	170
V.3	Stratégie d'évitement d'obstacles	178
V.3.1	Trajectoire locale d'évitement d'obstacle fixe	178
V.3.2	Principe de l'approche d'évitement d'obstacles dans un milieu non-cartographié prenant en compte le gabarit du véhicule	180
V.3.3	Évitement d'obstacles dans un milieu cartographié	181
V.3.4	Déplacement dans un couloir de navigation cartographié avec prise en compte du gabarit du véhicule	183
V.4	Génération d'une section de trajectoire	184
V.4.1	Trajectoire bi-élémentaire avec courbures initiales et finales non nulles . .	184
V.4.2	Généralisation de la méthode de création de trajectoire par séquences VCC	186
V.4.3	Résultats sur la génération de trajectoire	188
V.5	Commande des actionneurs	189
V.5.1	Gestion de la distance de sécurité et du freinage d'urgence	189
V.5.2	Commande des actionneurs pour le suivi de trajectoire	191
V.5.3	Résultats du suivi de trajectoire	196

V.6	Bilan sur la navigation	198
Bilan général, questions ouvertes, conclusions et perspectives		201
	Bilan général	201
	Questions ouvertes	203
	Conclusions et perspectives	204
A	Annexes	207
A.1	DBSCAN - Algorithme de partitionnement de données	207
A.2	χ -Shape - Algorithme pour décèlement de formes	210
A.3	Comparaison des algorithmes <i>Bresenham</i> et <i>Raycast</i>	212
A.4	Zone de recherche de la projection d'un point 3D dans l'image	213
A.5	Algorithmes d'appariements de grilles d'occupation	214
A.6	Détecteur et descripteur de points d'intérêts	217
A.7	Estimation des paramètres d'un modèle par l'algorithme RANSAC	219
A.8	Algorithme PSO - Particle Swarm Optimization	221

Bibliographie

Abbreviations

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AEB	Automatic Emergency Braking
AGV	Automated/Automatic Guided Vehicle
ALV	Autonomous Land Vehicle
ANRT	Association Nationale de la Recherche et de la Technologie
AUV	Autonomous Unmanned Vehicle
AVP	Autonomous Valet Parking
BA	Bundle Adjustment (Ajustement des faisceaux)
BSW	Blind Spot Warning
CACS	Comprehensive Automobile Traffic Control System
CAN	Controller Area Network
CCD	Charge-Coupled Device
CES	Consumer Electronics Show
CIFRE	Conventions Industrielles de Formation par la REcherche
CMOS	Complementary Metal-Oxide-Semiconductor
CVMS	Clustering Views for Multi-view Stereo
CUIK	Complete Universal Inverse Kinematics
DAI	Détection Automatique d'Incidents
DARPA	Defense Advanced Research Projects Agency
DGPS	Differential GPS
DLT	Direct Linear Transform
DSLR	Digital Single-Lens Reflex
EBA	Emergency Brake Assist
eCall	Emergency Call
EKF	Extended Kalman Filter
ESP	Electronic Stability Program
Euro NCAP	European New Car Assessment Programme
FCW	Forward Collision Warning
FLANN	Fast Library for Approximate Nearest Neighbors
FPGA	Field-Programmable Gate Array
GIS	Geographic Information Systems
GLONASS	GLObal NAVigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPS-RTK	GPS-Real Time Kinematic

HBA	H igh B eam A ssist
HFP	H ands F ree P arking
HLDI	H ighway L oss D ata I nstitute
IFR	I nternational F ederation of R obotics
IHS	I nsurance I nstitute for H ighway S afety
IMOB3	I nnovative M OBility : S mart and S ustainable S olutions
IMU	I ntertial M easurement U nit
INPI	I nstitut N ational de la P ropriété I ndustrielle
IRNSS	I ndian R egional N avigation S atellite S ystem
IRSTEA	I nstitut N ational de R echerche en S ciences et T echnologies pour l' E nvironnement et l' A griculture
ISP	I mage S ignal P rocessor
ITS	I ntelligent T ransport S ystems
LADAR	L AsEr D etection A nd R anging
LIDAR	L Ight D etection A nd R anging
LRR	L ong- R ange R adar sensor
LCA	L ane C entering A ssist
LDW	L ane D eparture W arning
LKA	L ane K eeping A ssist
LKC	L ane K ee P C enter
LKW	L ane K eeping W arning
MDAR	M obile D etection, A ssesment and R esponse S ystem
MNT	M odèle N umérique de T errain
MRR	M id- R ange R adar sensor
NNSS	N evada N ational S ecurity S ite
NHTSA	N ational H ighway T raffic S afety A dmistration
OICA	O rganisation I nternationale des C onstructeurs A utomobiles
ONISR	O bservatoire N ational I nterministériel de la S écurité R outière
OpenMP	O pen M ulti- P latform shared-memory parallel programming
PoE	P ower o ver E thernet
PMVS	P atch-based M ulti-view S tereo S oftware
QZSS	Q uasi- Z enith S atellite S ystem
RFID	R adio F requency I Dentification
RADAR	R adio D etection A nd R anging
RANSAC	R andom S ample C onsensus
ROI	R egion of I nterest
ROV	R emotely O perated V ehicle
SAGT	S ystèmes d' A ide à la G estion de T rafic
SAM	S canning A coustic M icroscopy
SCU	S ystème de C apteurs U ltrason
SIVIA	S et I nversion V ia I nterval A nalysis
SfM	S tructure from M otion
SSD	S um of S quared D ifferences
SAD	S um of A bsolute D ifferences
SAE	S ociety of A utomotive E ngineers
SIG	S ystème d' I nformation G éographique
SLAM	S imultaneous L ocalization A nd M apping

SLAG	S imultaneous L ocalization A nd G uidance
SRR	S hort- R ange R adar sensor
SURF	S peeded U p R obust F eatures
TJA	T raffic J am A ssist
TJP	T raffic J am P ilot
ToF	T ime of F light
TPMS	T ire P ressure M onitoring S ystem
TSD	T raffic S ign D etection
UAV	U nmanned A erial V ehicle
UGCV	U nmanned G round C ombat V ehicle
USA	U ltrasound A ssistance
USS	U ltra S ound S can
UUV	U nmanned U ndersea V ehicles
V2I	V ehicle-to- I nfrasturcture
V2V	V ehicle-to- V ehicle
V2X	V ehicle- to - E verything
VaMoRs	V ersuchsfahrzeug für a utonome M obilität und R echnersehen
VIPA	V éhicule I ndividuel P ublic A utonome
WDR	W ide D ynamic R ange
ZNCC	Z ero N ormalized C ross C orrelation

CHAPITRE I

Introduction

I.1 Le monde de la robotique

Le monde de la robotique est en plein essor et les robots font leur apparition dans tous les milieux [Todd, 2012]. Si on s'intéresse à l'origine du mot *robot*, on sait qu'il a été utilisé pour la première fois en 1920 par l'écrivain tchèque Karel Capek dans la pièce de théâtre *Rossum's Universal Robots* [Čapek and Selver, 1925] mais, il a lui-même désigné son frère, Josef Capek, comme l'inventeur réel. La racine du mot *robot* est le proto-slave *orbota*¹ dont sont issus le russe et le bulgare *rabota* ou le mot roumain (*a*)*roboti* (travail insignifiant autour de la maison). Si on reprend la définition de *Larousse*, le robot est un «*appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe, modifiable ou adaptable*».

De nos jours, on trouve des robots dans beaucoup de domaines dont on présente une liste non-exhaustive² : la médecine [Lanfranco et al., 2004], l'exploration spatiale [Diftler et al., 2015], le domaine militaire [Altmann et al., 2013], le domaine du transport de marchandises [Balyo, 2016] ou dans l'utilisation domestique [Cakmak and Takayama, 2013].



(a) Robots ouvriers dans une usine de montage Mitsubishi en Chine



(b) *Scooba* de [iRobot, 2016]



(c) *Buddy* de [Blue Frog Robotics, 2016]

FIGURE I.1 – Exemples de robots dans différents domaines d'application

L'industrie est le premier domaine qui emploie en masse les robots. Pour baisser le coût de production, les industriels ont choisi de remplacer les employés par des robots pour certaines tâches répétitives, simples nécessitant une précision constante [Engelberger, 2012]. La figure I.1(a) illustre une ligne d'assemblage robotisée. Cela a permis l'augmentation des cadences de production tout en augmentant la précision des opérations et en diminuant le nombre de rebuts. L'IFR³ annonce qu'en 2014, le nombre de robots industriels vendus a atteint un nouveau record avec 229 261 unités, en croissance de 29% par rapport à 2013. 70% de la vente sur 2014 a été réalisée uniquement dans cinq pays : Chine, Japon, États-Unis, Corée du Sud et Allemagne.

La robotique est également arrivée dans nos maisons pour nous exempter de certaines tâches. On trouve, par exemple, le robot autonome de nettoyage du sol *Scooba* (figure I.1(b)) de l'entreprise *iRobot* [iRobot, 2016], ou le robot aspirateur *Roomba* ou de nettoyage de piscine *Mirra*. Un autre domaine où les robots entament une percée, est l'accompagnement des personnes. Les robots d'accompagnement sont capables de réaliser une grande variété des fonctions comme : *assistance aux personnes âgées* (détection de chutes et manque d'activités, rappels de prise de médicaments), *ludo-éducation*, *assistant personnel* (mémo, réveil, informations météo ou trafic),

1. *orbota* est issu du mot l'indo-européen *orbh* (petit enfant, orphelin, jeune esclave), duquel sont issus l'arménien *orb*, l'hindi *arbha* (petit enfant), le grec ancien *orphanós* (dont viennent aussi l'anglais *orphan* et le français *orphelin*) le latin *orbis*, ainsi que l'allemand *Arbeit* (travail); 2. Pour une description exhaustive, on invite le lecteur à se référer aux huit axes de recherche du *GdR Robotique* sur le site internet <http://www.gdr-robotique.org/>; 3. International Federation of Robotics;

interaction sociale (télé-présence mobile, partage de photos et de vidéos, facilitation de l'accès aux technologies), surveillance du domicile (détection d'incendies, d'inondations et d'effraction), *domotique* (gestion de l'éclairage ou du chauffage) ou *contrôle à distance* via une application mobile. On peut citer l'entreprise *Blue Frog Robotics* [Blue Frog Robotics, 2016] qui propose son robot *Buddy* (figure I.1(c)), la référence des robots bas coût dans le domaine.

Un autre domaine en pleine expansion également est le transport des personnes et des biens. Depuis l'apparition du moteur à vapeur ce secteur n'a pas cessé son expansion, mais aujourd'hui les enjeux sont majeurs, car une part très importante de l'énergie fossile est consommée dans ce domaine [Multon, 1998] et implicitement cela occasionne un important rejet de polluants dans la nature [Seinfeld and Pandis, 2012]. La contribution de la robotique à ce domaine est multiple.

I.2 La réponse de la robotique aux problèmes de transport

Le flux de données, de personnes ou de marchandises continue à augmenter sans cesse et la robotique joue un rôle significatif grâce aux solutions qu'elle apporte aux problèmes de transport. En fonction du milieu où les robots évoluent, on trouvera des robots terrestres, marins et aériens. Un exemple de robot aérien est le drone (UAV¹) présenté dans la figure I.2(a). Il s'agit d'un robot nommé *hexacopter* capable de déplacer un colis léger. Autour de ce sujet, il existe de nombreux travaux de recherche en cours dans le but de ramener au plus vite sur le marché des flottes de robots capables de remplacer le travail du postier [Stolaroff, 2014] ou de réaliser des missions humanitaires [Gariépy et al., 2015] ou militaires [Giese et al., 2013]. Pour l'exploration sous-marine, on retrouve très souvent des robots sous-marins téléguidés, ROV² [Christ and Wernli Sr, 2013]. Des ROV sont très utilisés pour les films subaquatiques, ainsi que pour les recherches archéologiques subaquatiques [Forney et al., 2011], pour l'évaluation environnementale et notamment pour des inventaires de la biodiversité. Des ROV sont utiles également dans les eaux peu sûres ou très froides (sous des glaciers ou iceberg risquant de s'effondrer) [Hobson et al., 2011] ou pour des applications d'exploration des conduites industrielles pour l'industrie pétrolière [Salgado-Jimenez et al., 2010] ou pour des opérations de maintenances des câbles sous-marins [McLeod, 2013]. Dans le domaine militaire, ils sont connus sous l'appellation UUV³ et sont utilisés notamment pour la surveillance ou pour le déminage [Johnson and Aii, 2015]. Les robots qui sont autonomes dans leurs déplacements et qui sont capables d'effectuer des tâches sans être sous la surveillance d'une personne sont appelés des AUV⁴ [Paull et al., 2014].



(a) Le drone postier [Stolaroff, 2014]



(b) Le quadrupède *Big Dog* [Raibert et al., 2008]



(c) AGV dans la logistique [Balyo, 2016]

FIGURE I.2 – Exemple de robots autonomes utilisés pour le transport des différents objets

1. Unmanned Aerial Vehicle; 2. Remotely Operated Vehicle; 3. Unmanned Undersea Vehicles; 4. Autonomous Unmanned Vehicle;

La figure I.2(b) présente *Big Dog*, un des robots quadrupède de *Boston Dynamics* [Raibert et al., 2008]. *Big Dog* est un robot tout-terrain à quatre pattes articulées qui marche, court, grimpe et porte des charges lourdes jusqu'à 150 kg. Même si la marche à pied est très complexe et que la flexibilité que nous connaissons n'est pas facilement transposable sur les machines, il y a des avancées remarquables pour des robots comme *Nao* [Gouaillier et al., 2009] d'*Aldebaran-Robotics* ou le robot PETMAN [Raibert, 2010] et *Atlas* [Xinjilefu et al., 2014] de *Boston Dynamics*.

Le type de robot qui nous intéresse dans ce mémoire est le *robot mobile terrestre à 4 roues*. Un exemple représentatif est l'*AGV¹-transpalette* [Balyo, 2016] proposé par l'entreprise *Balyo* (figure I.2(c)) : un robot mobile capable de transporter dans l'entrepôt de la marchandise posée sur une palette. Même s'il n'y a pas de rails au sol, ces robots circulent toujours sur la même trajectoire et sont capables de se localiser dans l'environnement où ils évoluent. Pour savoir quelle mission doit être effectuée ou si le couloir de circulation n'est pas occupé par un autre robot, chaque robot communique par Wi-Fi avec une unité centrale qui se charge de gérer la flotte de robots et le flux des palettes.

Les avancées technologiques dans la robotique profitent également aux constructeurs automobiles qui misent fortement sur l'innovation pour développer le confort des occupants et la sécurité routière. Par la suite, nous allons présenter les avantages de l'utilisation du véhicule autonome et la manière dans laquelle celui-ci changera nos habitudes en manière de mobilité.

I.3 Le véhicule intelligent

I.3.1 Les enjeux

L'augmentation du nombre des véhicules sur la route lors des deux dernières décennies a entraîné la saturation de certaines routes pendant les heures de pointe. Dans le rapport réalisé par *Texas A&M Transportation Institute* en 2012 [Schrank et al., 2012] il est détaillé l'évolution du nombre des heures de retard cumulées par personne en fonction de la taille de la ville d'habitation. Selon ce rapport, en 2011, dans une ville de plus de 3 millions d'habitants, chaque personne cumule, par rapport à l'année 1981, 36 heures/an de retard de plus dans le transport, passant de 17 heures (en 1981) à 53 heures (en 2011) de retard par an. On estime que certaines fonctions d'aide à la conduite dans les embouteillages (TJA², TJP³) [Brookhuis et al., 2009] et de gestion du trafic (SAGT⁴) permettraient d'améliorer la dynamique de circulation et la capacité de l'infrastructure routière [Forster et al., 2012].

Une conséquence directe de l'augmentation de la fluidité du trafic est la réduction de la consommation d'énergie et des émissions de polluants. De plus, la circulation dans des *trains de véhicules*⁵ [Tsugawa et al., 2011] permettra de réduire la distance entre véhicules tout en gardant le niveau de sécurité. Les tests prouvent que la diminution de l'interdistance à 2–3 m apportera une réduction significative du coefficient de traînée⁶ et donc de la consommation d'énergie. Les études montrent que la consommation est réduite de 5% pour le véhicule leader et entre 10% et 15% pour les autres véhicules suiveurs [Alam, 2011; Bergenheim et al., 2012].

La délégation de conduite permettra également au conducteur de gagner du temps car, lors du trajet, il pourra réaliser d'autres tâches ou simplement se reposer.

1. Automated/Automatic Guided Vehicle; 2. Traffic Jam Assist; 3. Traffic Jam Pilot; 4. Systèmes d'Aide à la Gestion de Trafic; 5. *Platoon*, en anglais; 6. La *traînée* est la force de résistance qu'exerce un fluide sur un objet lorsque le fluide ou l'objet sont en mouvement l'un par rapport à l'autre;

Une autre attente liée à l'automatisation de la conduite est l'amélioration de la sécurité routière [Hojjati-Emami et al., 2012]. La cause principale des accidents est le manque d'attention des conducteurs et des recherches ont été réalisées [Young and Salmon, 2012] pour trouver la corrélation entre la distraction des conducteurs et leurs erreurs au volant. Les expériences montrent [Klauer et al., 2006] que 78% des accidents et 67% des presque-accidents sont causés par une perte d'attention inférieure à 3 secondes. D'autres études montrent qu'au total 96% des accidents sont dus aux erreurs humaines [Wierwille et al., 2002] et qu'un faible pourcentage est dû aux problèmes mécaniques. D'après ONISR¹, en 2015 en France, 3 464 personnes ont perdu la vie dans les accidents routiers (+2,4% de par rapport à l'année 2014), et 56 109 ont été blessés (-3,6% de par rapport à l'année 2014). Même si la mortalité est en légère hausse depuis 2014, il faut signaler que de 2000 à 2010, elle a été divisée par deux, passant de 8 170 à 3 992 décès.

Un rôle important dans l'amélioration de la sécurité des participants au trafic est joué par le programme *Euro NCAP*², qui évalue les nouveaux modèles de voitures et leur accorde une note de 0 à 5 étoiles. *Euro NCAP* publie les résultats d'une étude montrant que les «systèmes AEB³ peuvent réduire le nombre d'accidents jusqu'à 38% et entraîner une baisse significative du nombre de blessures» [Euro and Protocol, 2013; Chauvel et al., 2013; Seiniger et al., 2015]. Le 28 avril 2015, le Parlement Européen a voté l'équipement obligatoire de tout véhicule produit après avril 2018, des systèmes de *eCall*⁴ ou DAI⁵ qui alerteront automatiquement les autorités lors d'un accident, diminuant ainsi leurs délais d'intervention.

Un indicateur de performance des systèmes ADAS est la diminution de la fréquence des accidents depuis l'installation des systèmes ADAS d'alerte pour prévenir la collision comme FCW⁶ et LDW⁷. Dans ce sens l'institut américain HLDI⁸ annonce : -5,1% pour Honda Accord [HLDI, 2014, 2015a,b], -7,1% pour Mercedes-Benz [HLDI, 2012], -3,7% pour Mazda [HLDI, 2015c], +0,5% pour Subaru [HLDI, 2015d] ou des résultats remarquables de -21% pour Volvo [HLDI, 2015e]. De plus, l'institut américain IIHS⁹ s'intéresse à l'acceptation des systèmes ADAS par les conducteurs et réalise des enquêtes auprès de conducteurs de Volvo, Infiniti [Braitman et al., 2010], ou Toyota [Eichelberger and McCart, 2015]. Les résultats annoncent que 83% des conducteurs ne sont pas prêts à renoncer à ces systèmes, même si 47% parmi eux considèrent que le taux de fausses alarmes est trop important.

Par la suite, nous allons exposer les différents niveaux de délégation de conduite qu'il conviendra d'illustrer par les systèmes d'aide à la conduite correspondants.

I.3.2 Les niveaux de délégation de conduite et l'autonomie de certaines tâches

Conformément à la *Convention de Genève* signée en 1949 et à l'article 8 de la *Convention de Vienne*, dont la France est signataire depuis 1968, la législation impose qu'un conducteur soit dans le véhicule et qu'un lien mécanique existe entre le conducteur et le train roulant du véhicule. Mais cette réglementation est en train d'évoluer.

Aujourd'hui, le véhicule *Infiniti Q50* est équipé d'un système *drive-by-wire* qui remplace la liaison mécanique entre le volant et les roues par des moteurs électriques [Janbakhsh et al., 2013; Ulrich, 2013]. Pour que ce produit soit conforme aux normes, une liaison mécanique s'enclenche à l'aide d'un embrayage dès que la commande filaire est en défaut. De plus, si la réglementation actuelle

1. Observatoire National Interministériel de la Sécurité Routière; 2. European New Car Assessment Programme; 3. Automatic Emergency Braking; 4. Emergency Call; 5. Détection Automatique d'Incidents; 6. Forward Collision Warning; 7. Lane Departure Warning; 8. Highway Loss Data Institute; 9. Insurance Institute for Highway Safety;

permet bien l'utilisation des ADAS, elle interdit qu'un véhicule se substitue à la prise de décision humaine et impose que le geste de l'automobiliste prévale sur l'action du système.

Niveau d'autonomie	Contrôle longi./lat.	Surveillance environnement	Situation d'urgence	Mode de Conduite
Niveau 0	Conducteur	Conducteur	Conducteur	Manuel
Niveau 1	Conducteur et Système	Conducteur	Conducteur	Assistance au conducteur
Niveau 2	Système	Conducteur	Conducteur	Autonomie partielle
Niveau 3	Système	Système	Conducteur	Autonomie Conditionnelle
Niveau 4	Système	Système	Système	Haute Autonomie
Niveau 5	Système	Système	Système	Autonomie totale

TABLE I.1 – Niveaux d'autonomie du véhicule présentés par le SAE *International Standard J3016*

Dans le Tableau I.1 nous retrouvons une synthèse des niveaux d'autonomie du véhicule, tels que présentés par le SAE International Standard¹ J3016. Dans la littérature nous retrouvons également que les niveaux 0–2 s'appellent *eyes-on-hands-on*², le niveau 3 correspond à *eyes-on-hands-off*³, le niveau 4 à *eyes-off* et le niveau 5 à *mind-off* [Jensen et al., 2010].

L'apparition des voitures, comme la *Google Car*, et les avancées dans le domaine ont poussé les organismes à réaliser des classifications plus fines du fonctionnement autonome d'un système. Ainsi, pour qualifier le niveau d'automatisation du véhicule, il existe deux référentiels qui tiennent compte du taux de prise en charge de la tâche de conduite autonome. Selon l'agence américaine NHTSA⁴ il y a 5 niveaux d'autonomie décisionnelle notés de 0 à 4. En Europe, l'OICA⁵ subdivise le niveau 3 américain en 2 échelons distincts pour une quantification plus fine. Par la suite nous nous référerons à la norme européenne. Il est important de signaler que les trois premiers niveaux d'autonomie n'impliquent que des systèmes d'aide à la conduite (sans conduite autonome) et que le conducteur doit rester vigilant et prêt à intervenir rapidement :

- **Niveau 0 - Conduite manuelle** - le conducteur est en charge totale de la conduite du véhicule. Toutefois, le véhicule peut disposer de fonctions d'avertissement comme l'alerte de franchissement de ligne (LDW⁶), l'alerte de collision (FCW⁷), l'alerte de surveillance d'angle mort (BSW⁸), l'alerte de sous-gonflage des pneumatiques (TPMS⁹) [Davins-Valldaura et al., 2015, 2016], le changement automatique entre feux de croisement et feux de route (HBA¹⁰);
- **Niveau 1 - Assistance au conducteur** - à ce niveau d'assistance, le conducteur peut momentanément confier certaines tâches de conduite telle que la régulation de la vitesse (ACC¹¹) dans des environnements facilement interprétables comme l'autoroute. D'autres systèmes de sécurité classés à ce niveau et largement utilisés sont l'ABS¹² et l'ESP¹³;
- **Niveau 2 - Autonomie partielle** - le véhicule est doté des dispositifs d'aide à la conduite plus avancés qui peuvent intervenir dans des situations plus complexes comme : le centrage dans la voie (LKC¹⁴), l'assistance préventive à la sortie de voie (LKA¹⁵), la conduite dans les embouteillages (TJP¹⁶), l'aide au parking (HFP¹⁷, *ParkAssist*). Pendant l'intervention du système, le conducteur a un rôle plutôt de superviseur, mais la limite juridique actuelle

1. Society of Automotive Engineers; 2. Eyes on the road and hands on the wheel 3. Eyes on the road and hands off the wheel 4. National Highway Traffic Safety Administration; 5. Organisation Internationale des Constructeurs Automobiles; 6. Lane Departure Warnig; 7. Forward Collision Warning; 8. Blind Spot Warning; 9. Tire Pressure Monitoring System; 10. High Beam Assist; 11. Adaptive Cruise Control; 12. Anti-lock Braking System; 13. Electronic Stability Program; 14. Lane Keep Center; 15. Lane Keeping Assist; 16. Traffic Jam Pilot; 17. Hands Free Parking;

ne permet pas le transfert de responsabilité à un système même si dans certaines tâches il est plus fiable que l'humain compte tenu de sa vitesse de réaction ;

- **Niveau 3 - Autonomie conditionnelle** - pour des courtes durées le conducteur peut déléguer la conduite au système et peut abaisser la vigilance ; le type d'application est identique au niveau 2, mais la responsabilité est transférée à la machine. À ce niveau, c'est au véhicule d'annoncer en avance au conducteur qu'il arrive dans une situation incompréhensible pour lui et qu'il lui cède le contrôle. La *Google Car* (figure I.6(a)) est considérée à ce niveau ;
- **Niveau 4 - Haute autonomie** - le véhicule a une capacité quasiment complète d'autonomie et il peut contrer la décision du conducteur ; la présence du conducteur dans le véhicule n'est pas indispensable dans des applications comme celle de **voiturier autonome (AVP¹)** ;
- **Niveau 5 - Autonomie totale** - ce niveau d'autonomie est appelé *taxi autonome* car le véhicule peut fonctionner sans avoir à aucun moment un conducteur.

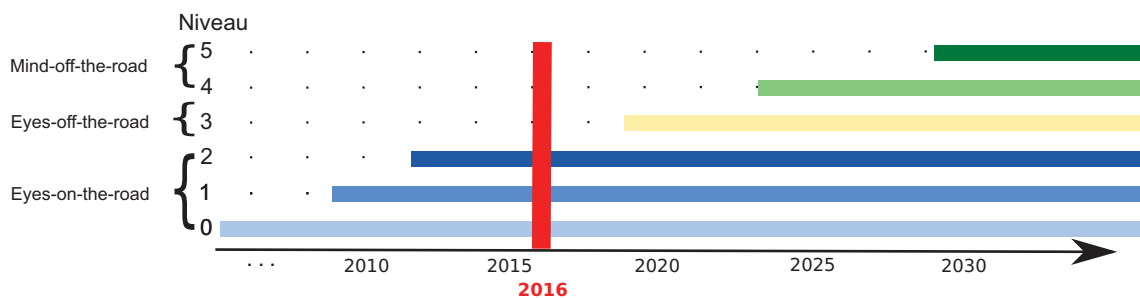


FIGURE I.3 – Prédiction de la commercialisation des systèmes ADAS en fonction de leurs niveaux d'autonomie décisionnelle

En 2016 les voitures sont équipées des ADAS du niveau 0 à 2 (figure I.3). L'arrivée du véhicule 100% autonome est imprévisible, mais les constructeurs s'accordent sur le fait que le marché des ADAS est en pleine expansion [Nakrani, 2015; Davidson and Spinoulas, 2015; Kirk, 2016].

I.3.3 Le long chemin vers l'automatisation du véhicule

Lors de l'exposition *Futurama* à New York de 1939, N. B. Geddes, parrainé par le groupe General Motors (GM), a présenté comment l'arrivée du véhicule autonome et des routes automatisées changera le monde dans les vingt ans qui suivent. Pourtant en 1953, c'est toujours le groupe GM qui annonce à nouveau lors de l'exposition *Motorama* l'arrivée imminente du véhicule autonome et présente un concept car *General Motors Firebird*, sans réaliser pour autant une vraie démonstration.

Selon la littérature scientifique, le premier véhicule automatisé a été construit au Japon en 1977, dans le cadre du projet CACS². Sous la direction du professeur S. Tsugawa [Tsugawa et al., 1979], des démonstrations ont été réalisées avec un véhicule capable de suivre un câble inductif et un peu plus tard c'est toujours lui qui présente un véhicule capable de suivre des lignes de peinture.

Dans les années 1980, Carnegie Mellon University débute les recherches sur les véhicules contrôlés par ordinateur et, en 1986, construit en parallèle les prototypes *Navlab* [Thorpe et al., 1988, 1989] et ALV³ [Turk et al., 1988; Kanade et al., 1986]. Toujours en 1986, l'équipe d'E. Dickmanns de *Bundeswehr University of Munich* en coopération avec Mercedes-Benz développe le projet de véhicule autonome VaMoRs⁴ [Bertozzi et al., 2000]. D'autres versions ont été développées comme VaMP

1. Autonomous Valet Parking; 2. Comprehensive Automobile Traffic Control System; 3. Autonomous Land Vehicle; 4. Versuchsfahrzeug für autonome Mobilität und Rechnersehen;

(en 1990) ou VITA-2. En octobre 1994, dans le cadre du programme *Eureka*¹ PROMETHEUS², les deux véhicules parcourent plus de 1 000 km sur l'autoroute tout en changement de voies de manière autonome après l'approbation de la manœuvre par le conducteur [Dickmanns, 2002].

En 2000 est lancé le projet européen *CarSense*³ [Langheim, 2001], qui avait pour but de mettre au point un système multicateur embarqué sur le véhicule. Peu de temps après, en 2001 démarre le projet *CyberCars*⁴ qui porte sur le développement et l'expérimentation de nouvelles techniques de transport. En 2002, dans le cadre du programme PREDIT⁵ est lancé le projet français AR-COS⁶ [Hernandez, 2004; Aubert et al., 2005] avec l'ambition de réussir une approche globale du système «véhicule-conducteur-infrastructure», en vue de réduire de 30% le nombre des accidents. Le projet RINAVEC⁷ début en 2008 et vise à développer et évaluer des fonctions avancées de perception et de modélisation de l'environnement, pour des véhicules évoluant en convoi sur un itinéraire inconnu a priori, en milieu ouvert (périurbain ou naturel) [Féraud et al., 2010a,b, 2011a; Devy et al., 2011; Márquez-Gámez, 2012].

Même si l'automatisation du véhicule a été plus lente que prévue, les grandes avancées de l'électronique ont permis la création d'une gamme large des capteurs suffisamment performants pour la perception de l'environnement que l'on détaillera dans le deuxième chapitre [§II.2.1].

Aujourd'hui les grands axes de recherche liés au véhicule autonome sont dirigés vers la création de l'intelligence embarquée et vers la conception de capteurs à bas prix.

Véhicules autonomes de patrouille et de combat

Le domaine du véhicule intelligent a connu d'énormes avancées grâce aux applications réalisées dans le domaine militaire où les contraintes ne sont pas d'ordre matériel. En 1958 DARPA⁸ [DARPA, 2016] est créé en réponse au lancement du programme soviétique *Sputnik*⁹. Cette institution joue un rôle essentiel car elle a organisé la première course de véhicules autonomes. Ainsi, le 12 mars 2004 débute la compétition *DARPA Grand Challenge (2004)* dans la région du désert de Mojave des États-Unis. Aucun des véhicules ne termine les 240 km de trajet et aucun gagnant n'est déclaré (la plus longue distance parcourue par les participants étant uniquement de 11,78 km). Un deuxième événement *DARPA Grand Challenge (2005)* est organisé une année plus tard et cette fois-ci plusieurs véhicules terminent la course et le temps record de 6h54 est établi par le véhicule *Stanley* de *Stanford University* [Thrun et al., 2006b].

Grâce à ces compétitions on peut profiter aujourd'hui des LiDAR 3D commercialisés par *Velodyne*. En 2004 cette entreprise commercialisait uniquement des systèmes audio, mais son savoir-faire dans le traitement de signal l'a poussée à créer une petite équipe pour développer un véhicule autonome. Ainsi, lors de la course *DARPA Grand Challenge (2004)*, *Velodyne* présente son véhicule capable de se guider utilisant uniquement la stéréovision. En se rendant compte de l'avantage du LiDAR, mais aussi de ses limites, pour la course de 2005, elle invente un nouveau LiDAR avec balayage 3D. Même si elle a dû abandonner la course après uniquement 41 km à cause de l'endommagement du LiDAR par les vibrations, *Velodyne* a créé un produit qui depuis est devenu incontournable dans la course à l'automatisation du véhicule.

1. Programme lancé en 1985 et destiné à renforcer la compétitivité de l'industrie européenne; 2. **PRO**gramme for a **E**uropean **T**raffic of **H**ighest **E**fficiency and **U**nprecedented **S**afety (1987–1995); 3. Sensing of Car Environment at Low Speed Driving; 4. **Cy**bernetic **C**ars for a New Transportation System in the Cities; 5. Programme de **RE**cherche et **D**'Innovation dans les Transports terrestres; 6. Action de **Re**cherche pour une **CO**nduite **S**écurisée; 7. **Re**connaissance d'**It**inéraires et **NA**Vigation **E**n **CO**nvoi de véhicules communicants; 8. **D**efense **A**dvanced **R**esearch **P**rojects **A**gency est une agence du département de la Défense des États-Unis; 9. Programme lancé par l'URSS dans le but d'envoyer des satellites sur orbite;

Entre 2005 et 2010 les avancées technologiques ont été très importantes. Par exemple, pour la protection des arsenaux nucléaires de NNSS¹, la surveillance humaine a été renforcée depuis 2010 par une flotte de véhicules de patrouille appelés *General Dynamics MDAR*² (figure I.4(a)) [Czapla and Wrona, 2013]. Ces véhicules robotisés sont équipés par des technologies onéreuses comme des LiDAR multi-nappes, des caméras très haute résolution, des caméras infrarouges qui sont capables de détecter de loin toute personne ou véhicule non-autorisé ayant pénétré le site.



(a) Flotte de robots mobiles *General Dynamics MDAR* en patrouille sur le site de NNSS [Czapla and Wrona, 2013]



(b) *Guardium MK III* - véhicule de l'entreprise G-NIUS Unmanned Ground Systems [Stewart, 2011; Dickow et al., 2015]

FIGURE I.4 – Véhicules autonomes de patrouille et de combat

Un autre exemple vient de l'entreprise israélienne G-NIUS qui propose une gamme de dispositifs patrouilleurs téléguidés, tout comme le fait l'entreprise *Samsung Techwin* en Corée du Sud. Le modèle *Guardium MK III* de G-NIUS (figure I.4(b)) va plus loin qu'un simple véhicule autonome [Stewart, 2011; Dickow et al., 2015]. Il est conçu pour une utilisation dans le combat (UGCV³) étant équipé d'armes létales. Ce véhicule très maniable sur le terrain accidenté est présenté comme ayant des capacités d'exploitation d'armes ou d'identification et classification des activités hostiles dans le but de fournir une réponse aux menaces sans mettre en danger le personnel.

Véhicule autonome civil

L'existant a été commencé au début de la thèse (Mai 2013) et mis à jour tout au long de ces trois ans. Pour cela nous nous sommes appuyés sur la littérature scientifique, mais également sur l'expertise du *Département ADAS, Véhicule Autonome et Technologies Électroniques* de Renault. De plus nous nous sommes basés sur des informations fournies par le *Centre d'Analyse à la Concurrence* de Renault, qui a pour but, comme le nom l'indique, d'analyser l'état d'avancement des techniques et technologies qui équipent les véhicules concurrents et qui met à disposition des collaborateurs des véhicules concurrents pour les tester. La collaboration avec ce centre nous a permis d'avoir accès et de tester différents véhicules équipés des systèmes d'aide à la conduite, notamment d'aide au parking et d'en présenter quelques-uns dans ce mémoire. Ces tests ont permis de comprendre comment le monde industriel se positionne par rapport à l'attente des clients, quel est le cas nominal d'utilisation de la fonction d'aide au parking ou de parking autonome et quelles sont les limites surtout en termes de perception. Nous avons pu comparer les capteurs utilisés et le niveau de maturité des différentes technologies. Cette étude a permis d'alimenter ce mémoire et de guider les recherches et le développement tout au long de ces trois ans.

1. Nevada National Security Site ; 2. Mobile Detection, Assessment and Response System ; 3. Unmanned Ground Combat Vehicle ;



(a) Mercedes S500 autonome

(b) Les capteurs et le champ de vue

FIGURE I.5 – Le véhicule autonome *Mercedes Benz S-Class S500 Intelligent Drive*

[Ziegler et al., 2014]

Par la suite nous allons présenter quelques véhicules représentant l'état de l'art de la conduite autonome. *Mercedes Benz* met en avant son véhicule concept *S-Class S500 Intelligent Drive* (voir figure I.5(a)) et l'annonce comme étant l'aboutissement de 20 ans de recherche et développement [Ziegler et al., 2014]. En 2014 l'entreprise fait la démonstration du résultat et le véhicule parcourt en automatique les 103 km de la mémorable route *Bertha Benz Memorial Route*, entre Mannheim et Pforzheim, en hommage au premier long voyage en voiture effectué par *Bertha Benz* en 1888. L'automatisation du véhicule a un prix ; même si le constructeur annonce que les capteurs sont de pré-série et que le prix sera abordable, il faudrait toutefois remarquer leur abondance : bouclier de capteurs ultrason, 4 radars courte portée (utilisés pour la surveillance des intersections), 2 radars longue portée installés sur les côtés du véhicule (pour la surveillance du trafic rapide dans les intersections de campagne), une caméra stéréovision, une caméra mono-vision grand angle en couleur pour la détection de feux rouges, une autre caméra grand angle en gris pour la localisation, deux radars longue portée à l'avant. Remarquons dans la figure I.5(b) que la surveillance de l'environnement se fait également vers l'arrière de la voiture et, à l'exception de la caméra stéréovision et des deux radars latéraux, les autres capteurs se retrouvent en symétrie. On constate rapidement que pour assurer la fonction autonome, le véhicule est équipé de beaucoup de capteurs en redondance ce qui rend encore très cher le véhicule commercialisable.

Un acteur incontournable dans le monde digital est *Alphabet Inc.* avec sa division de recherche *GoogleX*, en charge du développement des projets innovants comme *Google Car*, *Wing*, *Glass* ou *Loon*. Ce projet de développement de technologies pour le véhicule sans conducteur, *Google Car* (figures I.6(a)) est mené par l'ingénieur Google Sebastian Thrun, directeur du *Stanford Artificial Intelligence Laboratory*, co-inventeur de *Google Street View* et gagnant du *DARPA*¹ *Grand Challenge* en 2005 [Levinson et al., 2011].

En juin 2015, Google annonce que son véhicule a parcouru **1 million de miles** en autonome sur route ouverte, ce qui fait de la *Google Car* la voiture autonome la plus avancée [Pinto, 2012; GoogleCar, 2016]. Ce n'est donc pas étonnant qu'en février 2016 l'agence fédérale américaine en charge de la sécurité routière, NHTSA², a répondu à Google dans une lettre officielle lui précisant que le système équipant le véhicule autonome peut être considéré comme le conducteur du véhicule [NHTSA, 2016], ce qui permettra prochainement au dernier modèle de véhicule autonome de Google (figure I.6(b)), prévu sans volant, de rouler en route ouverte sans être surveillé par un opérateur. Même si la *Google Car* avait été impliquée dans plusieurs accidents, l'analyse des enregistrements démontrait que le conducteur de l'autre voiture impliquée était en tort. Ce fait prouvait

1. Defense Advanced Research Projects Agency ; 2. National Highway Traffic Safety Administration ;

(a) Flotte de véhicules autonomes *GoogleCar*(b) La *GoogleCar* sans volantFIGURE I.6 – La *Google Car*

qu'on s'approchait d'un système dont les performances sont telles qu'il conduisait mieux que l'humain. Mais, le 29 février 2016 la *Google Car* provoquait son premier accident [Rapport Accident Google Car, 2016]. L'accrochage s'est produit lors de l'insertion dans une voie où la *Google Car* qui circulait alors à 3-4 km/h avait prédit que le bus qui s'approchait à 15 km/h et qui était en train de ralentir allait s'arrêter pour lui permettre de s'insérer. C'est ce qu'on appelle un cas classique de négociation entre les conducteurs qui est un phénomène normal et souvent rencontré dans le trafic où les conducteurs se comprennent par un regard ou par un signe. Cette situation montre qu'il y a encore un long chemin à parcourir pour transférer à la machine toute la compétence humaine de conduite, mais surtout pour lui apprendre le langage non-verbal.

En octobre 2014 Audi bat le record de vitesse en conduite autonome sur circuit avec son véhicule concept *RS7 Bobby* (figure I.7(a)) qui atteint son maximum de 239,95 km/h lors de l'exécution d'une trajectoire presque parfaite [Audi piloted driving, 2016]. Dans la figure I.7(b), on retrouve *RS7 Robby* équipé, comme son frère, d'un GPS à précision centimétrique qui indique à la voiture la position sur le circuit via un réseau de très haute fréquence. Audi ouvre ainsi la voie vers l'automatisation des tests sur piste pour la vérification de la robustesse des véhicules.

(a) *Audi RS 7 Bobby* autonome(b) *Audi RS 7 Robby* autonome

FIGURE I.7 – Audi - véhicules autonomes sur piste [Audi piloted driving, 2016]

Le 22 mars 2015 l'équipe d'ingénieurs de Delphi, débute un voyage de 5 633km dans leur véhicule *Roadrunner* (figure I.8(a)), une Audi SQ5 équipée d'un système de conduite autonome. Le projet *US Coast-to-Coast Automated Drive* [Delphi Drive, 2016], comme le nom l'indique, a pour but de traverser les États-Unis de la côte Ouest (de San Francisco), à la côte Est (à New York). Delphi annonce que pour 99% du temps, le véhicule a fonctionné en autonome. On constate que pour environ 1 km tous les 100 km (le 1% qui reste), c'était une personne qui prenait la commande du véhicule et on ne connaît pas exactement les détails de cette prise en manuelle. Lors du Salon

*International CES*¹ de 2015 qui s'est tenu à Los Angeles, Delphi a présenté leur véhicule et les capteurs qui l'équipent : 6 radars longue portée, 4 radars courte portée, 3 systèmes de caméras, 6 LiDAR et un système GPS haute résolution.



(a) Le véhicule *Roadrunner* de Delphi et le projet *US Coast-to-Coast Automated Drive* [Delphi Drive, 2016]



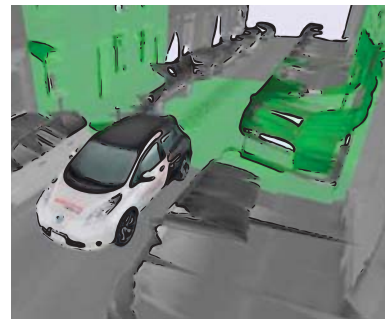
(b) Le véhicule autonome de PSA Peugeot Citroën lors du voyage Paris - Bordeaux [PSA Paris-Bordeaux, 2016]

FIGURE I.8 – Delphi et PSA Peugeot Citroën - véhicules autonome pour des longs trajets

En France, en juillet 2015, PSA Peugeot Citroën devient le premier constructeur qui a obtenu les autorisations pour faire rouler des prototypes autonomes. C'est ainsi qu'une de ses 4 voitures autorisées (figure I.8(b)) part de Paris et arrive à Bordeaux en ouverture du *ITS*² *World Congress 2015*. Dans le communiqué de presse du 2 octobre 2015, le constructeur affirmait que «*Les 580 km ont été parcourus de manière autonome sans intervention du conducteur. La voiture a ajusté seule sa vitesse et ses dépassements en fonction des autres véhicules, des limites réglementaires et de l'infrastructure*» [PSA Paris-Bordeaux, 2016], mais il ne précise pas qu'en dehors de l'autoroute le véhicule a été conduit manuellement comme le montre les vidéos publiées.



(a) La *RobotCar*



(b) Cartographie 3D utilisant un LiDAR installé en plan vertical

FIGURE I.9 – Le véhicule *RobotCar* développé par MRG Oxford et Nissan [Maddern et al., 2015]

Nissan s'est rapproché du laboratoire de recherche MRG³ Oxford et ils créent ensemble la *RobotCar*. Ce véhicule est équipé de plusieurs LiDAR, des caméras et d'un ordinateur qui effectue les calculs nécessaires à la perception, navigation et au contrôle du véhicule. La spécificité de ce projet est la fixation d'un des LiDAR de telle sorte que le faisceau soit en plan vertical, perpendiculaire à la route (figure I.9(b)) [Maddern et al., 2015], donc lors de l'avancement du véhicule, l'accumulation des points observés permet la construction d'une carte 3D.

Renault s'est engagé également dans la course du véhicule autonome et du renforcement de

1. Consumer Electronics Show; 2. Intelligent Transport Systems; 3. Mobile Robotics Group;



FIGURE I.10 – Le projet de véhicule autonome *Next Two* de Renault [Renault Next-Two, 2016]

la sécurité routière et, en février 2014, a dévoilé son projet *Next Two* (Figure I.10). *Next Two* est un véhicule concept présentant une fonction de délégation de conduite [Renault Next-Two, 2016]. Renault compte redonner du temps aux automobilistes en leur proposant un système de connectivité haut-débit avec des fonctionnalités de Cloud, de bureau virtuel à distance ou de vidéoconférence. Ainsi, les conducteurs ont la possibilité de choisir différents trajets et parmi ceux-ci sont proposés même des trajets plus longs mais, en les empruntant, le véhicule est capable de passer en mode autonome.

Sur ce modèle, la délégation de conduite est disponible uniquement sur les voies rapides et les autoroutes où il n’y a pas de piétons ou de cyclistes, ou sur toute route lors des embouteillages si la vitesse est inférieure à 30 km/h. Renault montre également que *Next Two* est capable de se garer ou de venir chercher son conducteur grâce à une fonction de voiturier autonome. Le véhicule présenté est équipé d’un radar placé sur le bouclier avant, une caméra située au niveau du rétroviseur central, une caméra arrière et une ceinture de capteurs d’ultrasons autour du véhicule.



(a) *Motobot* de Yamaha [Borgul et al., 2012; Bobtsov et al., 2013]



(b) Robot de conduite *AB Dynamics* [Ambrus et al., 2013; Praveen Kumar Peter, 2013]

FIGURE I.11 – Des robots de conduite

D’autres *robots de conduite* sont à mentionner. Par exemple, lors du *The 44th Tokyo Motor Show 2015*, Yamaha a présenté *Motobot* (figure I.11(a)) un robot capable de conduire une moto de série à des vitesses de 200 km/h [Borgul et al., 2012; Bobtsov et al., 2013]. Un autre exemple est l’entreprise *AB Dynamics* qui construit également des robots de conduite (figures I.11(b)) capables de réaliser des manœuvres sur piste de manières répétables [Ambrus et al., 2013; Praveen Kumar Peter, 2013].

Nous détaillerons, par la suite, comment le véhicule autonome partagé pourra convaincre plus de personnes de ne pas prendre leur véhicule personnel pour le trajet entre domicile et travail.

I.3.4 Le problème du dernier kilomètre

Le train régional ou le train de banlieue sont des moyens de transport populaires qui permettent d'augmenter la couverture du réseau de transport en commun. Pour maintenir une vitesse moyenne élevée, les stations sont écartées d'un ou plusieurs kilomètres. Cela fait que de nombreuses résidences et des entreprises sont situées au-delà d'une distance de marche non négligeable et constitue donc un obstacle à une meilleure utilisation du réseau de transport en commun régional. Ce problème est connu comme le *problème du premier/dernier kilomètre* et, tel que présenté par la suite, pourrait être résolu par la robotique autonome [Zhou, 2012; Chong et al., 2013].

Michel Parent (INRIA ¹) énonce le problème *Space × Time Efficiency* [Parent, 2007] qui se propose de mesurer la surface occupée par une personne et son moyen de transport lors d'un trajet. Dans ce problème il inclut également une notion de *surface dynamique*, signifiant la surface occupée par le véhicule pendant le déplacement, tout en tenant compte de la distance inter-véhicules. Comme on l'attendait, cette surface est beaucoup plus petite pour les personnes qui prennent le bus et très grande en cas d'utilisation de la voiture personnelle (figure I.12).

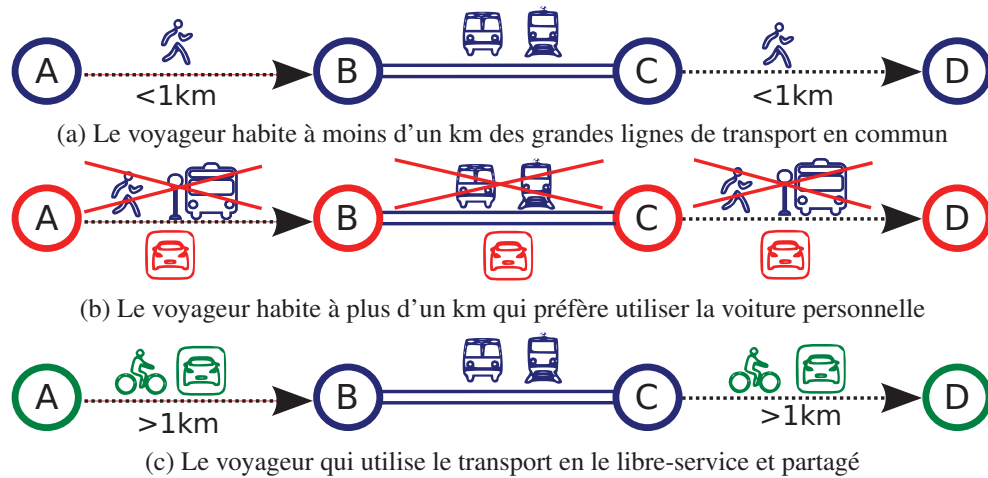


FIGURE I.12 – La surface occupée par 40 voyageurs en fonction du moyen de transport utilisé [Renaud, 2016]

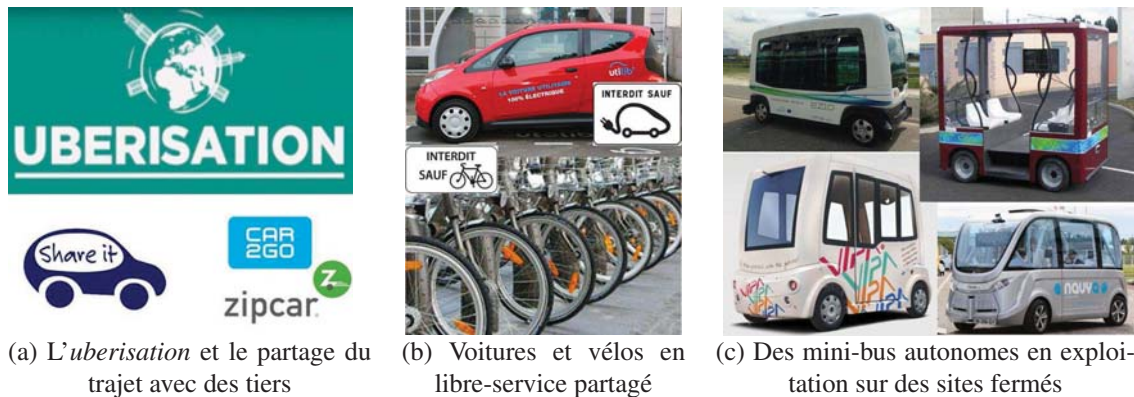
En fonction de l'éloignement entre le domicile/le travail et les stations des grandes lignes, une personne peut se retrouver dans plusieurs situations. La figure I.13(a) illustre le cas où la personne habite à moins d'un kilomètre de la station. Ceci est le cas favorable car le voyageur acceptera de faire une marche pour récupérer le transport en commun. Par contre si le voyageur habite à plus d'un kilomètre et s'il est obligé d'attendre pour prendre un autre bus pour l'amener à la station de grandes lignes de transport en commun, alors il y a une forte chance pour qu'il préfère prendre sa voiture personnelle pour réaliser l'ensemble du trajet (figure I.13(b)).

Pour répondre à ce problème, un marché s'est créé et une gamme de produits est proposée aux voyageurs. Par exemple, pour le partage des frais de transport, le covoiturage (figure I.14(a)) séduit de plus en plus de voyageurs, leur évitant ainsi de prendre leur véhicule personnel tous les jours. Mais, pour des petites distances, le transport en libre-service peut être partagé (comme le *Vélib* ou l'*Autolib* à Paris, figure I.14(b)) et peut permettre de ramener les voyageurs à la station de transports en commun (figure I.13(c)) sans pour autant qu'ils aient à se soucier de la place de parking [Largentaye, 2010; Nair et al., 2013; Hildermeier and Villareal, 2014]. L'efficacité du libre partage se mesure en chiffres : l'entreprise *Bolloré* annonce plus de 200 000 abonnés au service *Autolib* (à Lyon, Bordeaux et dans la métropole parisienne), 14 000 locations par jour, en moyenne, et plus de 70 millions de kilomètres parcourus depuis 2011. De plus on annonce que 50% des abonnés

1. Institut National de Recherche en Informatique et en Automatique;

FIGURE I.13 – Le problème du *premier/dernier kilomètre*

qui utilisent les véhicules *Autolib* en Ile-de-France ne sont pas résidents. Selon l'institut américain *Earth Policy Institute*, en septembre 2014, plus de 700 villes dans 57 pays disposaient d'un système de partage de vélo, la flotte totale s'élevant à 800 000 unités. A Paris, depuis le lancement du système *Vélib* en 2007, le nombre de cyclistes a augmenté de 41%. Les usagers pouvant choisir 24h/24 un vélo parmi les 23 600 unités réparties sur 1 800 stations situées à une distance moyenne de 300 m. Cela montre qu'il y a des gens prêts à abandonner leur véhicule pour certains trajets s'ils peuvent utiliser un service de location flexible avec un coût inférieur à l'utilisation du taxi.

FIGURE I.14 – Exemples de solutions au problème du *dernier kilomètre*

A ces gammes de produits, on ajoute les minibus autonomes (figure I.14(c)). Ceux-ci sont des véhicules capables de circuler en autonomie à des vitesses relativement faibles (10-40 km/h) dans des conditions environnementales connues. On cite ici le véhicule VIPA¹ EZ10, le résultat du partenariat entre les chercheurs de l'Institut Pascal (sous la cotutelle du CNRS et de l'Université Blaise Pascal), et l'industriel *Ligier Automobiles*. Le concept de VIPA est celui de *ascenseur horizontal*, comme le constructeur l'annonce, et apporte une solution aux déplacements dits *du dernier kilomètre*. Il est capable de se déplacer en autonomie entre plusieurs stations. Les utilisateurs, via l'utilisation d'un écran tactile, peuvent indiquer à quelle station ils souhaitent être déposés et le véhicule se chargera de l'optimisation du trajet. Un des avantages notables est un fonctionnement 24h/24, avec des horaires plus flexibles et avec une coordination centralisée et optimale de la flotte. Cela permettra une meilleure coopération avec les systèmes de transport existants. De plus,

1. Véhicule Individuel Public Autonome;

par exemple, si les véhicules *Autolib* disposaient d'un système de conduite autonome, la répartition territoriale des voitures pourrait se faire automatiquement. Ainsi, on ne retrouvera plus, comme aujourd'hui, des zones dans les villes où les stations *Autolib* sont pleines et d'autres zones où elles sont vides.

I.4 Le voiturier autonome

En reprenant la définition du dictionnaire *Larousse*, on définit un *voiturier* comme «un employé chargé de garer les voitures des clients d'un hôtel ou restaurant». L'ajout du mot *autonome* change le sens et un *voiturier autonome* devient un système capable de garer une voiture sans l'intervention d'une personne ou avec une intervention limitée. Pour ce faire il y a plusieurs manières d'agir : en premier lieu on automatise le parking qui est un immeuble dédié, la deuxième possibilité est d'avoir des robots qui transportent ou qui tractent les voitures et la toute dernière possibilité est que la voiture même soit intelligente et qu'elle sache où et comment se garer.



FIGURE I.15 – Le parking robotisé de Volkswagen de *Autostadt*, Allemagne

Dans le premier cas nous présentons le parking intelligent d'*Autostadt*, un parc de loisirs axé sur l'automobile et situé à côté de l'usine Volkswagen en Allemagne. Ici un parking robotisé en forme de tour circulaire (inspiré d'un système de *transtockeur*¹) manipule les voitures (figure I.15).

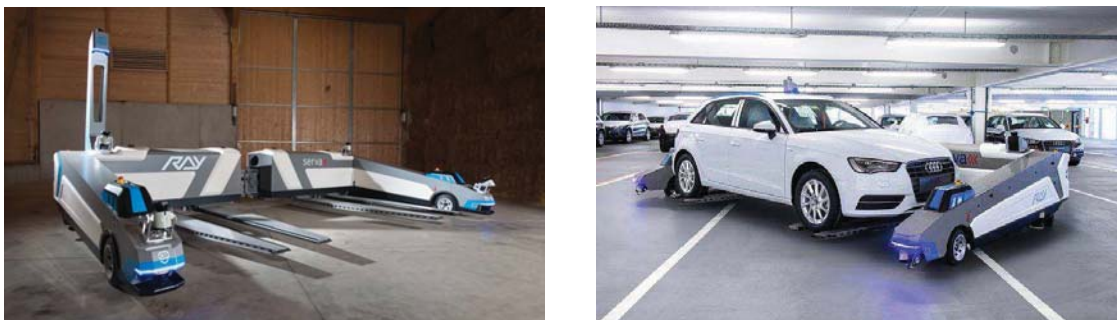


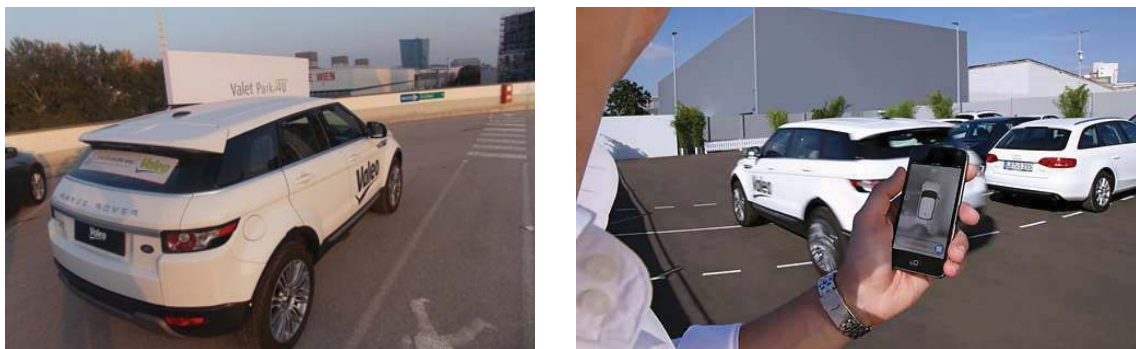
FIGURE I.16 – Le projet RAY, résultat de la coopération entre Audi AG et Serva [Meirer et al., 2013]

Dans le deuxième cas de voiturier autonome l'intelligence est placée à l'extérieur du véhicule et un robot (se charge du déplacement des véhicules figure I.16). L'aéroport de Düsseldorf expérimente un système appelé RAY - AGV² [Meirer et al., 2013]. Ce projet est réalisé par l'entreprise *Serva* en collaboration avec *Audi AG* et vise à rendre le parking de l'aéroport, rapide, facile et agréable sans pour autant attendre l'arrivée de l'automatisation du véhicule à grande échelle.

1. Dispositif automatisé ou non qui permet de ranger des palettes ou des colis dans un rack, souvent à grande hauteur; 2. Automatic Guided Vehicle;

La dernière solution, qui nous intéresse davantage, est de mettre l'intelligence dans le véhicule. L'objectif de ce type de sous-système est de rendre la conduite autonome dans des environnements désignés (par exemple, un parking fermé, un parking extérieur ou à la maison).

Avant de concevoir des systèmes de parking très complexes, comme le *voiturier autonome*, les industriels ont débuté par la commercialisation des dispositifs d'aide à la mesure de la place de parking. Bien que très efficace, beaucoup de constructeurs ont préféré passer directement au niveau supérieur, en proposant des systèmes HFP¹. Ceux-ci gèrent la position du volant lors de la manœuvre et sont utilisables dans des situations de parking latéral. Si les systèmes gèrent le contrôle latéral et également le contrôle longitudinal, alors on les appelle génériquement des systèmes d'*AutoPark*. Pour tous ces systèmes, la présence du conducteur dans le véhicule est exigée.

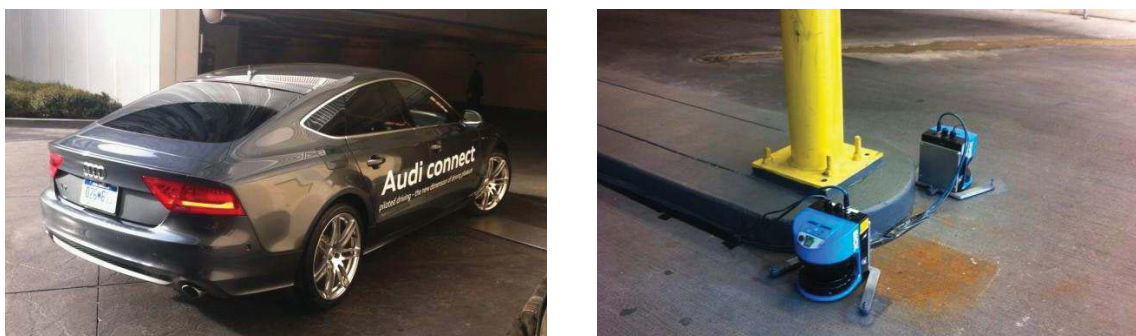


(a) Valeo - Démo parking automatique

(b) Smartphone utilisé pour contrôle à distance

FIGURE I.17 – Le système *Valet Park4U* de Valeo

L'étape suivante sera de proposer des systèmes de *remote parking*. Dans ce sens, l'entreprise Valeo présente son système *Valeo Park4U* [Hamada et al., 2015] installé sur un *Range Rover Sport SVR* (figure I.17(a)). Celui-ci est capable de se garer dans une place de parking [Bammert et al., 2013; Barth and Jecker, 2013] pour un scénario simplifié : une fois que le conducteur passe devant une place de parking et que le véhicule valide la bonne détection, il peut descendre de la voiture et continuer à surveiller son véhicule et, si nécessaire, de l'arrêter depuis son smartphone (figure I.17(b)). Cette option d'arrêt d'urgence permet de garder juridiquement la maîtrise du véhicule. Ce véhicule est richement équipé de capteurs comme : un LiDAR, 12 capteurs ultrason, 4 caméras grand-angle et une connexion 3G/4G. Le prix élevé du LiDAR pour l'instant, rend non commercialisable ce véhicule en série.



(a) Audi A7 - Démo voiturier automatique

(b) Lidars installés à l'extérieur du véhicule

FIGURE I.18 – Voiturier autonome d'Audi présenté lors du salon *International CES 2013*

1. **Hands Free Parking** ;

Lors du salon *International CES 2013*¹ les visiteurs ont eu l'opportunité d'essayer le voiturier autonome d'Audi (figure I.18(a)) [Hachm, 2013]. Pour remédier au problème du prix des capteurs onéreux, Audi a proposé d'installer les LiDARS comme infrastructure (figure I.18(b)) et de guider le véhicule par Wi-Fi. Ainsi les véhicules étant équipés uniquement de capteurs ultrason et odométriques ainsi que d'une unité de commande et contrôle [Reichel and Bouzouraa, 2015].



(a) Scénario type du voiturier autonome

(b) Véhicule de démonstration

FIGURE I.19 – Le projet *V-Charge – Driverless Parking And Charging for eMobility* [V-Charge et al., 2011-2016]

Un exemple d'approche est donné dans le projet *V-Charge - Driverless Parking And Charging for eMobility* [V-Charge et al., 2011-2016]. Dans la description technique, le véhicule de démonstration (figure I.19(b)) est équipé d'un GPS à faible coût, de caméras et de capteurs à ultrasons. Un scénario typique d'un tel concept (figure I.19(a)) pourrait être l'automatisation de dépôt et de reprise d'une voiture en face d'une gare sans que le conducteur ne se donne la peine de veiller au stationnement ou au rechargement.

Ce projet est basé sur la vision que la mobilité va subir des changements importants dans les années à venir. En raison du renforcement des contraintes imposées par les normes, dans le but de réduire les émissions de CO₂ et de réduire également la consommation d'énergie. Cela inclut l'introduction de voitures électriques qui doivent être rechargées d'une manière coordonnée.

Par la suite, nous allons exposer le cadre de travail de la thèse, nos choix en termes de technologies et de capteurs qui nous permettront de concevoir un système de parking automatique polyvalent et plus performant que ceux actuellement proposés en série.

1. Consumer Electronics Show;

I.5 Le cadre de travail

Ce sujet s'intègre dans l'axe de développement des systèmes d'aide à la conduite et de conduite autonome que Renault développe en interne ou en partenariat avec des industriels ou avec des laboratoires de recherche. Ce travail de recherche est réalisé dans le cadre d'une thèse CIFRE¹ en partenariat entre Renault, Institut Pascal et IRSTEA² et consiste à approfondir les problématiques de perception et de guidage pour une application de **voiturier autonome à la maison**.

Dans cette introduction nous avons cité des véhicules avec de fonctionnalités d'aide à la conduite à différents niveaux d'autonomie. Même si les démonstrations ont montré la faisabilité de ces systèmes, il reste beaucoup de travail de recherche et d'industrialisation pour apporter le niveau de fiabilité attendu. La majorité des démonstrateurs qui ont des fonctionnalités d'un niveau d'autonomie supérieur au *niveau 3* [§I.3.2] font appel à des capteurs extéroceptifs performants mais très onéreux cependant. De plus, il y a des acteurs sur le marché qui ont une énorme avance, comme la filiale *GoogleX* de *Alphabet Inc.*, car elle s'appuie sur des cartes réalisées dans le cadre du projet *Google Street View* qui aujourd'hui sont à leur disposition et utilisées pour enrichir les connaissances sur l'environnement de leur *GoogleCar* (même si à la base ce n'était pas conçu pour le projet de véhicule autonome) [Chafkin, 2013].

On considère qu'il y a aujourd'hui trois volets importants de recherche dans l'industrie automobile : *le véhicule connecté, le véhicule autonome et le véhicule écologique/économique*. Pour le domaine écologique, il est important de souligner la performance du projet *Eolab* [Maroteaux et al., 2015] qui a réussi la performance d'atteindre une consommation de seulement 1l/100km comme résultat de la combinaison de trois ingrédients : aérodynamique, allègement et hybridation essence-électrique. En ce qui concerne les deux autres volets, on constate qu'ils ne font pas partie du cœur de métier de l'industrie automobile. Aujourd'hui les industriels sont dans la phase plutôt d'acquisition de compétences en interne ou de création des synergies avec d'autres secteurs High-tech pour développer leurs produits. Afin de diffuser sur leurs véhicules des systèmes d'aide à la conduite, les constructeurs d'automobiles ont restreint le cadre applicatif à des systèmes d'avertissement, à des systèmes de niveau d'autonomie de 1 et 2 qui sont fonctionnels dans des conditions maîtrisées ou facilement interprétables (l'autoroute). Un autre système proposé est le freinage d'urgence automatique qui est déclenché dans des situations où l'impact est imminent. En ce qui concerne les applications d'aide au parking, aujourd'hui en série ne sont proposées que des fonctions du type « park assist ». Celles-ci gèrent uniquement le parking latéral en créneau, en épi ou en bataille. En plus il est exigé que le conducteur soit au volant pour surveiller la manœuvre et pour intervenir au cas où une situation imprévue est rencontrée.

Les technologies conçues pour faciliter la vie des conducteurs lors des manœuvres de parking que Renault commercialise à ce jour sont : l'avertissement sonore de collision d'obstacles (situés à l'avant, à l'arrière ou en latérale), la caméra de recul, le frein de parking assisté ou le système le plus complexe qui est le parking mains libres (HFP³) nommé également *Easy Park Assist*. Ainsi le dernier modèle Renault Espace peut (presque) se garer tout seul. Plus précisément le véhicule aide le conducteur à identifier un espace de stationnement, en mesurant les espaces libres d'une certaine longueur minimale grâce aux capteurs d'ultrasons et, une fois que le conducteur confirme la validité de la place trouvée, le système prend alors en charge la direction lors des

1. Conventions Industrielles de Formation par la REcherche ; 2. Institut National de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture ; 3. Hands Free Parking ;

manœuvres laissant l'accélération, le freinage et le changement de rapport marche avant/marche arrière à la charge du conducteur.

I.6 Objectif de la thèse et scénario typique

Ce projet veut répondre au problème suivant : concevoir une voiture autonome en mesure d'effectuer une manœuvre de parking tout en tenant compte de la diversité des environnements. Le but est de concevoir un système de parking automatique beaucoup plus performant et polyvalent que ceux actuellement disponibles (qui ne permettent que le parking latéral). Dans le cadre de cette thèse il est imposé de conserver une définition technique des capteurs extéroceptifs limités en prix et en gabarit. Le contexte de l'étude portera sur l'automatisation de manœuvres de parking en milieu initialement inconnu mais qui s'améliorera au fur et à mesure de son utilisation. Un cas d'usage typique est de permettre au véhicule de se rendre automatiquement dans la zone de garage du domicile de son propriétaire, cette fonction est classiquement appelée **voiturier autonome**, **parking valet à domicile** ou en anglais **home valet parking**. L'idée générale est de doter le véhicule de fonctionnalités lui permettant d'aller se garer, par exemple dans un garage, après avoir déposé ses occupants devant l'entrée principale de l'habitation (voir figure I.20)). Ce choix de contexte, de parking dans un espace dédié, présente de fortes exigences en termes de perception de l'environnement et développement de fusion de données multi-capteurs. Néanmoins ce contexte est moins contraignant que la conduite sur route ouverte (environnement dynamique, présence d'autres véhicules roulants, signalisation routière, etc.).

Nous ne nous intéressons pas à traiter la notion de *place de parking* comme information de haut niveau. Pour le véhicule, la place de parking représente une position et une orientation dans laquelle il devrait se retrouver. Le scénario typique (figure I.20) fait appel aux notions de **zone de reprise / de dépose du véhicule** correspondent à l'endroit où le propriétaire laisse le véhicule pour qu'il parte se garer seul et à l'endroit où le véhicule l'attendra plus tard.



(a) Conduite manuelle vers la place de parking (trajectoire blanche) puis évolution automatique du véhicule dans le garage (trajectoire bleue)



(b) Conduite automatique vers la zone de reprise du véhicule (trajectoire bleue) puis conduite manuelle (trajectoire blanche)

FIGURE I.20 – Voiturier autonome - scénario typique

Dans le cadre de ce travail, nous allons utiliser les **phases d'apprentissage** et de **conduite autonome**. La première (réalisée manuellement par le conducteur) représente l'enchaînement typique des événements : le conducteur conduit jusqu'à la position de dépose et, utilisant une IHM, il signale au véhicule que la position atteinte est dans la *zone de dépose* ; ensuite il conduit manuellement vers sa place de parking et, une fois arrivé, toujours utilisant l'IHM, il signale la position comme étant la *place de parking*. Enfin, il sort de la place de parking (fait demi-tour si la configuration le permet) et se dirige vers la *zone de reprise* du véhicule enregistrant cette dernière information par l'IHM. Durant la manœuvre, le système collecte des informations liées au trajet

et à l'environnement. À la fin du processus, le système de voiturier autonome devra être exploitable ; dans le cas contraire le véhicule devra signaler une éventuelle défaillance d'enregistrement et demander, le cas échéant, au conducteur de recommencer l'apprentissage.

La deuxième phase est celle d'exploitation ou de conduite autonome. Le véhicule surveille en continu sa position GPS et détecte automatiquement lorsque le conducteur se dirige vers une zone déjà apprise. Ainsi, à l'approche de cette zone, la reconnaissance de l'environnement s'active automatiquement. Par conséquent, au moment où le conducteur est arrêté, le véhicule est censé connaître sa position et pouvoir proposer au conducteur de passer en conduite autonome jusqu'à la place de parking. Pour une utilisation facile, le système doit permettre au conducteur de déposer sa voiture dans une zone assez large. En plus, si un obstacle se trouve sur la trajectoire habituelle et s'il y a de la place, le véhicule devra le contourner. On notera que la zone potentiellement traversable peut ne pas être complètement observable depuis un point de vue. Ultérieurement, l'algorithme de génération de la trajectoire proposera de contourner l'obstacle ou de s'arrêter. Il est bien entendu que s'il n'y a pas de place, le voiturier autonome va rester bloqué et signaler ce fait au propriétaire.



FIGURE I.21 – Cas d'utilisation du voiturier autonome avec possibilité d'activation du système depuis une position non apprise (cas de multiples trajectoires)

La figure I.21 présente une configuration multi-trajets jusqu'à la place de parking. Dans ce cas d'utilisation le conducteur peut engager le système de voiturier autonome depuis une position non apprise. Par conséquent, le système ne devra pas permettre le démarrage en automatique. Cependant, le système peut suggérer à l'utilisateur de continuer en mode manuel jusqu'à ce que sa position soit détectée avant de continuer sa tâche en automatique.

I.7 Organisation du mémoire

Le contexte dans lequel le véhicule va évoluer, la limitation en matière de choix des capteurs et les cas d'usages que l'on envisage nous conduisent à définir les caractéristiques du système :

- fortes exigences en termes de perception de l'environnement ;
- fusion des données multi-capteurs (caméra, ultrasons, odométrie) ;
- détection d'obstacles et de la zone navigable ;
- localisation du véhicule dans l'environnement ;
- navigation et planification de trajectoire.

On retrouve ainsi les fonctions principales d'un système d'automatisation de conduite, néanmoins dans un contexte moins contraignant que la conduite sur route ouverte. Ces grandes fonctions sont développées dans les chapitres II à V. Pour une homogénéité de la présentation, ces chapitres ont la structure générique suivante : problématique, état de l'art, développements réalisés, résultats et un bilan.

- Le chapitre II (Perception de l'environnement) : traite la problématique de la détection et du suivi d'obstacles, les moyens de perception et de l'acquisition de l'information ;
- Le chapitre III (Localisation) : concentre l'état de l'art sur la localisation des robots dans différents environnements avec une focalisation sur les techniques adaptées à l'application de voiturier autonome. Nous présentons les choix effectués et également la stratégie de mise en commun de ces techniques de localisation complémentaires ;
- Le chapitre IV (Détection d'obstacles et analyse de la zone traversable) : traite la problématique de la détection du couloir traversable, les techniques de détection par capteurs ultrason et par caméra et de suivi des obstacles et la manière dont l'information est sauvegardée et réutilisée. Nous présentons également des résultats sous forme de cartes d'occupation construites grâce aux informations issues des capteurs embarqués ;
- Le chapitre V (Navigation) : ce chapitre est dédié à la navigation du véhicule et utilise les fonctionnalités de gestion du couloir traversable et de localisation construites dans les deux chapitres précédents. Nous présentons ici les problématiques de planification globale et locale avec une illustration de la création de trajectoire pour des situations de parking.

La dernière partie du mémoire est réservée aux conclusions et aux perspectives.

I.8 Contributions

Ce travail s'appuie sur les activités de recherche des deux laboratoires ainsi que sur le savoir-faire de l'industriel de Renault qui a mis à disposition des équipements spécifiques. Mon appartenance à trois entités différentes m'a permis d'avoir accès à des simulateurs, robots, véhicules ou installations spécifiques et à des jeux de donnée issues parfois d'autres projets.

A. État de l'art : analyse de la littérature scientifique, du niveau d'avancement interne Renault et analyse et tests des véhicules concurrents ;

B. Définition d'une architecture système : l'étape précédente a permis l'élaboration de la logique de fonctionnement du voiturier autonome, basée sur les services et performances attendus, ainsi les réalisations liées à l'architecture système sont :

- **conception d'une architecture fonctionnelle** - constituée d'un ensemble de fonctions interconnectées, de scénarios et de modes opératoires ;
- **conception d'une architecture software modulaire** - création de briques/modules software facilement réutilisables dans un autre projet ;
- **interfaçage** - adaptation et création des interfaces pour la connexion au véhicule prototype.

C. Gestion de la traversabilité : partant d'un système propriétaire fermé de capteurs ultrason, nous avons réussi à réaliser une carte d'obstacle à des niveaux de précision supérieure au produit d'origine ; l'information de la carte est sauvegardée utilisant des techniques de stockage en arbre qui permet d'économiser de la mémoire et de créer des cartes multi-résolution ; une augmentation de la zone de détection des capteurs d'ultrasons a été réalisée utilisant une caméra et une détection d'obstacle via un technique SfM ¹ tout en gardant l'aspect temps réel ;

D. Localisation : un algorithme de localisation par appariement de cartes d'obstacles en temps réel est proposé ; on propose également des améliorations de temps de calcul et d'adaptations à la technique de localisation par appariement entre une image actuelle et une base d'images ;

1. Structure from Motion ;

nous proposons également une stratégie d'utilisation de plusieurs techniques de localisation dans le but d'assurer le fonctionnement de la localisation par tout temps ;

E. Navigation : nous avons développé une technique de planification locale pour le guidage du véhicule lors de la présence d'obstacles sur la trajectoire habituelle ;

F. Valorisation du travail :

- **Articles :** M. Chirca, R. Chapuis, G. Martin, C. Debain, and R. Lenain. *Autonomous valet parking system architecture*. In 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), pages 2619–2624, Sept 2015 [Chirca et al., 2015a] ;
- **Brevets d'invention déposés auprès de l'INPI**¹ :
 - *Device and method for automatically parking a motor vehicle*, Patent FR3032410A1 WO2016128654A1 [Chirca et al., 2015b] ;
 - *Procédé et dispositif de détermination de positions spatiales et de paramètres cinématiques d'objets dans l'environnement d'un véhicule*, Brevet App. FR n° 15-63078 [Chirca and Martin, 2015] ;
 - *Dispositif et procédé de détection de place de stationnement libre pour un véhicule automobile*, Brevet App. FR n° 15-57491 [Chirca, 2015a] ;
 - *Dispositif et procédé d'aide au stationnement pour un véhicule automobile*, Brevet App. FR n° 15-57593 [Chirca and Minoiu-Enache, 2015] ;
 - *Dispositif d'aide à la conduite pour estimation du danger d'une situation*, Brevet App. FR n° 15-59022 [Minoiu-Enache and Chirca, 2015a] ;
 - *Système de commande de la direction d'un véhicule automobile en cas de collision imminente avec un obstacle*, Brevet App. FR n° 15-59256 [Minoiu-Enache and Chirca, 2015b] ;
 - *Procédé et dispositif de réglage automatique d'éléments du poste de Conduite*, Brevet App. FR n° 15-63083 [Chirca, 2015b].
- **Vulgarisation des travaux en conférences, groupes de travail et écoles d'été à travers des présentations orales ou posters :**
 - 18th IEEE International Conference on Intelligent Transportation Systems, 2015, Espagne ;
 - XXVème Colloque Grets, 2015, France ;
 - 2015 TEMPO Research Review and Automotive Embedded Control Workshop, France ;
 - JNRDM 2015 - Journées Nationales du Réseau Doctoral en Micro-nanoélectronique, France ;
 - 9th International Conference on Advanced Science and Technology, 2014, France ;
 - Journées de l'École Doctorale, 2014, France ;
 - Journée des Jeunes Chercheurs en Robotique, 2014, France ;
 - Ma thèse en 180 secondes, 2014, France ;
 - V-Charge Summer School - Perception and Planning for Autonomous Driving, 2014, Suisse ;
 - ICVSS 2014 - International Computer Vision Summer School, 2014, Italie.
- **Participation à la vulgarisation du sujet et des résultats en interne Renault.**

1. Institut National de la Propriété Industrielle ;

CHAPITRE II

Perception de l'environnement

Sommaire

I.1	Le monde de la robotique	2
I.2	La réponse de la robotique aux problèmes de transport	3
I.3	Le véhicule intelligent	4
I.3.1	Les enjeux	4
I.3.2	Les niveaux de délégation de conduite et l'autonomie de certaines tâches	5
I.3.3	Le long chemin vers l'automatisation du véhicule	7
I.3.4	Le problème du dernier kilomètre	14
I.4	Le voiturier autonome	16
I.5	Le cadre de travail	19
I.6	Objectif de la thèse et scénario typique	20
I.7	Organisation du mémoire	21
I.8	Contributions	22

II.1 Problématique

Quand un véhicule roule de façon autonome, il est impératif que toutes les conditions de sécurité soient réunies. Parmi ces conditions nous évoquons la détection fiable et en temps réel des obstacles et de leurs positions ainsi que la capacité du véhicule à éviter les collisions. Dans ce chapitre nous nous intéressons à la problématique de la détection du couloir traversable et nous présentons les moyens de perception de l'environnement, les techniques de détection et de suivi des obstacles et la manière dont l'information est sauvegardée et réutilisée.

Dans le principe il y a deux façons de traiter le problème de détection d'un chemin admissible. La première méthode consiste à interpréter en temps réel l'environnement, mais dans la pratique il s'avère complexe d'extraire toute l'information nécessaire. Quant à la deuxième méthode, elle requiert l'utilisation d'une carte préalablement construite qui contient les informations relatives aux obstacles fixes et aux zones interdites. Dans ce deuxième cas, le problème se simplifie car l'intelligence de compréhension de l'environnement a été transposée dans la carte et il ne reste qu'à se localiser dans cette carte et éventuellement à éviter les obstacles dynamiques.



FIGURE II.1 – Signalisation routière (obstacles réglementaires)

En ce qui concerne la notion d'obstacle, il faut garder à l'esprit que les obstacles peuvent être répartis en deux grandes classes : l'obstacle réglementaire et l'obstacle physique. Dans le cas des **obstacles réglementaires**, même si le véhicule pourrait passer dans le couloir, les conventions font qu'il n'en a pas le droit. L'ensemble des lois et règlements relatifs à l'utilisation des voies publiques constitue ce qu'on appelle le *Code de la route*. Un exemple d'obstacle réglementaire est le panneau de *Circulation interdite à tout véhicule dans les deux sens* (figures II.1(a)) qui interdit l'accès total. On peut retrouver également des signalisations qui sont liées aux dimensions du véhicule, comme le panneau *Accès interdit aux véhicules dont la hauteur, chargement compris, est supérieure au nombre indiqué*, à l'utilisation du véhicule ou au statut du conducteur; on cite également la voie réservée aux bus (figure II.1(b)), la ligne mixte de signalisation ou simplement le signal d'arrêt de l'agent de police (figure II.1(c)). Même si pour la détection des panneaux [Mogelmose et al., 2012; Timofte et al., 2014], il existe déjà des systèmes (TSD¹) comme ceux proposés par l'entreprise *MobileEye* [MobileEye, 2016] qui équipent certains véhicules, en ce qui concerne la détection des signaux du policier, étant donné la complexité du problème, les études en sont globalement à leurs débuts [Yuan and Wang, 2010]. Précisons également que certains obstacles réglementaires peuvent ne pas être signalés. Typiquement, dans une propriété privée, il peut y avoir des zones où le bon sens dit de ne pas aller, une pelouse par exemple. Nous reviendrons sur ce sujet dans la section §II.2.5 pour approfondir la problématique de détection d'obstacles pour une situation de parking.

La deuxième catégorie est constituée des **obstacles physiques**. Comme le nom l'indique, elle est composée des obstacles physiques qui composent et définissent le monde qui nous entoure

1. Traffic Sign Detection;

et sans précision expresse, dans ce manuscrit, le mot *obstacle* fera référence à la catégorie associée. En fonction de leur possibilité de se déplacer, on les classe en : *obstacles statiques* (bâtiments, arbres, véhicules en stationnement et toute l'infrastructure de manière générale) et *obstacles dynamiques* (piétons, véhicules, animaux, etc.).

Dans ce mémoire, la diversité de l'environnement ne sera pas traitée de façon exhaustive et notre système prendra en compte uniquement les obstacles physiques. Il existe une vaste littérature scientifique qui traite de la détection d'obstacles, la classification [Asvadi et al., 2016; Qi et al., 2016] et leur suivi [Bandyopadhyay et al., 2006; Richards et al., 2014; Petrovai et al., 2015]. Dans un premier temps, on s'intéresse à l'état de l'art des moyens de perception, ensuite de la détection des obstacles, puis nous reviendrons sur les techniques d'analyse de la zone traversable, de localisation et de navigation.

II.2 Perception : État de l'art

II.2.1 Perception de l'environnement et moyens d'acquisition de l'information

Dans cette section seront abordées la manière de percevoir l'environnement utilisant des capteurs ainsi que l'exploitation de la communication entre le véhicule et l'infrastructure ou directement entre plusieurs véhicules.

II.2.1.1 Capteurs pour l'industrie automobile

Le capteur occupe un rôle extrêmement important car, grâce à lui, on est capable d'acquérir de l'information et de la *quantifier*. Si on reprend la définition du *Larousse*, le capteur est un «organe qui élabore, à partir d'une grandeur physique, une autre grandeur physique, souvent de nature électrique, utilisable à des fins de mesure ou de commande». On appelle *stimulus*, le phénomène physique exploité pour récupérer l'information. A titre d'exemple il existe des stimuli¹ électriques, mécaniques, thermiques, optiques, magnétiques, chimiques, nucléaires, etc. Derrière chaque type de stimulus, une large gamme de produits est proposée sur le marché. En ce qui concerne la mesure d'une information, nous faisons référence à 4 notions : *grandeur*², *mesurande ou variable*³, *signal*⁴ et le *capteur* même [Fraden, 2004; Vetelino and Reghu, 2010]. Quant à l'acquisition de l'information, la *chaîne de mesure* est composée de : *l'élément sensible* (actif⁵ ou passif⁶) qui entre en contact avec le phénomène physique et est affecté par la grandeur physique à mesurer, *le transducteur* qui transforme le signal émis par le capteur en information (généralement électrique) et le *transmetteur* qui s'occupe de l'envoi de l'information.

L'industrie automobile n'est pas passée à côté de cette innovation et le nombre de capteurs utilisés est estimé à 20 milliards par an. En 2001, W. J. Fleming [Fleming, 2001, 2008], dans son bilan, affirmait qu'en 1995 le moteur d'une voiture était équipé de 10 capteurs et prévoyait qu'en 2010 il serait équipé d'au moins 30 capteurs. A ce jour, chaque véhicule a en moyenne 60 à 100 capteurs à bord et les versions haut de gamme ont environ 150 capteurs. Parce que les voitures deviennent de plus en plus intelligentes et parce qu'il y a besoin d'information robuste et redondante, le nombre de capteurs devrait dépasser la barre des 200 capteurs par voiture en 2020. Pour répondre à des

1. Latin, pluriel du *stimulus*; 2. Phénomène physique observé et quantifiable; 3. Grandeur qui fait l'objet de la mesure. En industrie on utilise couramment variable ou variable de procédé; 4. Grandeur mesurable; 5. Génère un signal électrique de façon directe, sans l'aide d'une énergie extérieure; 6. Subit une variation d'une propriété physique et génère un signal électrique de façon indirecte, avec l'aide d'une énergie extérieure;

questions liées à la génération future des capteurs et de leur application, de nombreux événements annuels sont tenus partout dans le monde (dont nous rappelons les deux plus importants : Automotive Sensors and Electronics Expo et IEEE Sensors).

Chaque capteur est caractérisé par un certain nombre de critères dont on mentionne les plus courants : *temps de réponse, plage de mesure, sensibilité, précision, fidélité, reproductibilité, bande passante, résolution, hystérésis, gamme de température d'utilisation*, etc. Ce sont toutes ces caractéristiques qui font que les gammes de capteurs proposées sur le marché sont très variées en termes de performances et de prix. Un autre facteur qui joue sur le prix du capteur est la présence, en plus de sa fonction métrologique, d'autres fonctions intelligentes [Yamasaki, 1996]. Prenons quelques exemples de *fonctionnalités métrologiques intelligentes* : capacité d'auto-adaptabilité au signal mesuré, capacité de remplacement des données manquantes, ou de filtrage de l'information, de la validation des mesures et de la détection des mesures aberrantes. D'autres fonctionnalités fortement recherchées dans l'industrie automobile sont les *fonctionnalités de maintenance* comme l'autodiagnostic de l'état de fonctionnement, la capacité à fonctionner en mode dégradé, ou la capacité de communication bidirectionnelle pour une flexibilité instrumentale [Webster and Eren, 2014].

II.2.1.2 Capteurs de perception extéroceptifs

Pour un véhicule, la perception se réalise via des capteurs proprioceptifs, extéroceptifs et via la communication. Dans notre application de voiturier autonome, l'information est issue uniquement de capteurs propres au véhicule qui ont comme objectif de surveiller l'état interne et l'environnement dans lequel le véhicule évolue. Nous nous intéressons d'abord à l'analyse de la zone traversable et aux capteurs capables de percevoir des distances. Dans la section suivante nous passons en revue les capteurs suivants : le RADAR¹, le capteur ultrason et les systèmes de capteurs ultrason, le LiDAR² mono-nappe et multi-nappes, le capteur d'images décliné dans des mono-caméras, des stéréocaméras ou des systèmes multi-caméras, les systèmes de détection par lumière structurée et nous allons finir avec un des derniers capteurs de distance, la caméra RGB-D³, également nommée caméra ToF⁴.

RADAR

Le radar est un instrument de mesure qui indique la présence d'un objet éloigné, sa taille, sa vitesse et sa direction. En 1863, le physicien britannique James Clerk Maxwell met en équations le comportement des ondes électromagnétiques [Maxwell, 1863, 1881], mais c'est uniquement en 1904 que l'ingénieur allemand Christian Hülsmeier suggère l'utilisation du radar pour la détection des navires [Kirsch and Grinberg, 2008] (figure II.2(a)). Même si à l'origine il a été développé pour des raisons militaires, le radar est aujourd'hui également utilisé pour d'autres applications comme : le contrôle de vitesse des automobiles, le contrôle du trafic aérien ou maritime, la météorologie, la volcanologie et, depuis quelques années, dans les systèmes d'aide à la conduite.

Dans la figure II.2(b) nous présentons le radar longue portée LRR3 de Bosch qui, par rapport aux radars militaires (figure II.2(a)), ne pèse que quelques centaines de grammes. Ce radar fonctionne sur la bande 76–77 GHz approuvée à l'utilisation dans l'industrie automobile et est capable

1. **RA**dio **D**etection **A**nd **R**anging ; 2. **L**ight **D**etection **A**nd **R**anging avec les acronymes **LIDAR**, **LiDAR**, **LIDaR** ou **Lidar** ; Connue également sous l'acronyme **LADAR** - **L**aser **D**etection **A**nd **R**anging 3. «D» de Depth ; 4. **T**ime of **F**light ;



FIGURE II.2 – Le radar

de détecter des obstacles situés jusqu'à 250 m pour une ouverture angulaire de 30° . Une application standard, dans l'industrie automobile, est le système de maintien de la distance de sécurité (ACC¹) ou le système de freinage d'urgence (EBA²) tels que présentés dans la figure II.2(c).

Bouclier de capteurs ultrason

Comme le RADAR, le capteur ultrason a été développé également pour une utilisation militaire. En 1916 et 1917, Paul Langevin et Constantin Chilowsky déposent deux brevets, révélant le premier détecteur sous-marin à ultrason.

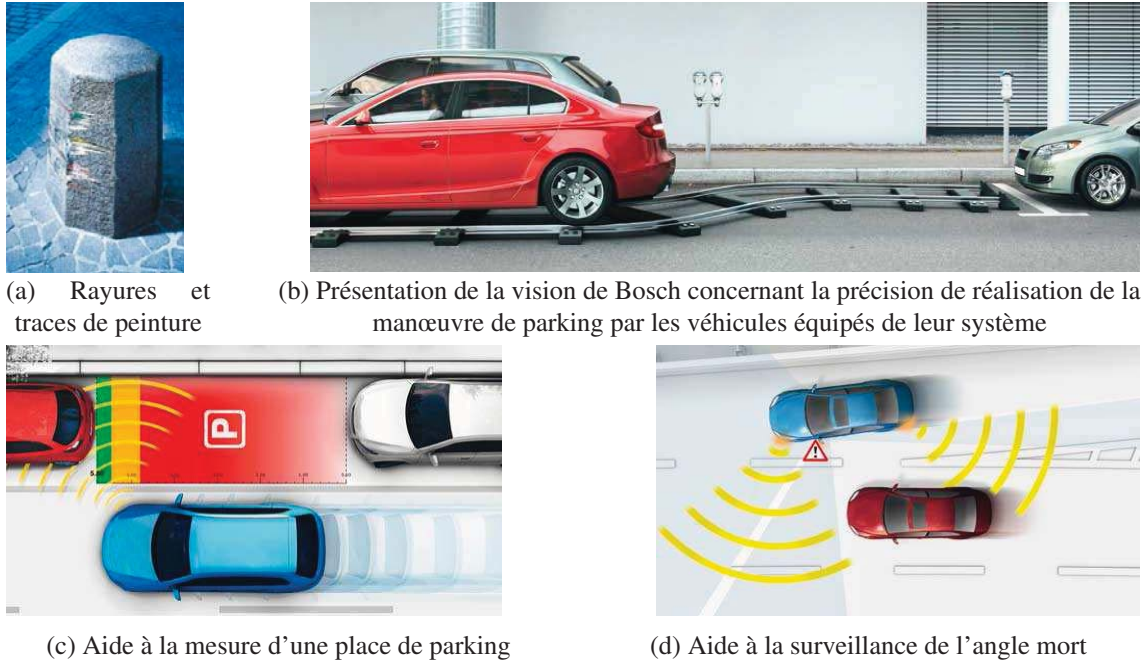


FIGURE II.3 – Systèmes ADAS développés par Bosch utilisant une perception de l'environnement à base de capteurs ultrason

De nos jours, le capteur de détection d'obstacles le plus déployé sur les véhicules est le capteur ultrason. L'utilisation initiale de ce capteur était d'alerter le conducteur lors de la détection des obstacles très proches du véhicule et, ainsi, d'éviter les rayures. Même si initialement la plage

1. Autonomous Cruise Control; 2. Emergency Brake Assist;

de détection était de 0,2 m à 1 m, aujourd'hui les avancées technologiques permettent l'extension de la distance de perception à 4 m voire plus pour les capteurs de présérie. Cette augmentation de la portée a permis l'extension de la gamme d'applications. A titre d'exemple l'entreprise Bosch décrit sa vision sur la manœuvre autonome de parking latéral et sa conviction que cette fonction apportera plus de confort au conducteur par la précision de l'exécution de la manœuvre (voir figure II.3(b)). Ainsi, elle propose aujourd'hui aux constructeurs automobiles d'intégrer son système de détection de place de parking (figure II.3(c)) et de pilotage de la manœuvre ou de détection des objets dans l'angle mort (voir figure II.3(d)).

Nous allons revenir à ces systèmes, dans la section §II.3.1 pour détailler les performances des capteurs ultrason, leurs limites et les raisons pour lesquelles ces capteurs sont devenus si populaires dans l'industrie automobile.

LiDAR

Le LiDAR¹ fait partie des systèmes de télédétection par laser et la technologie permet de mesurer des distances grâce à l'analyse d'un faisceau de lumière renvoyé vers son émetteur. Ce capteur est utilisé pour la première fois en 1963 (peu après l'invention du laser² en 1960) pour une application en météorologie [Roberts et al., 2009; Hedrick et al., 2014]. La communauté scientifique et l'industrie prend conscience de l'utilité de ce capteur uniquement en 1971 lorsque les premières cartes de la surface de la lune (réalisées par les astronautes de la mission Apollo 15) sont publiées.

En associant un mécanisme de balayage angulaire, grâce à un axe rotatif, il est possible de scanner un espace en deux dimensions. Ce capteur est couramment nommé LiDAR 2D. Nous avons identifié deux grandes catégories d'applications où les LiDAR 2D sont utilisés. La première catégorie est utilisée pour la protection contre la collision ou pour la protection contre la pénétration dans un espace protégé. Dans ce sens l'entreprise SICK [SICK, 2016] est le leader du secteur des scrutateurs laser de sécurité. La deuxième catégorie est utilisée pour la perception de l'environnement, principalement pour des applications de cartographie des objets perçus mais également du milieu de propagation de l'onde laser (notamment pour des applications de météorologie). Même si le prix d'un LiDAR 2D est relativement accessible (dès 398\$ pour le moins cher à ce jour, *RPLIDAR 360° Laser Scanner*, à plusieurs milliers d'euro), la perception de l'espace sous forme d'une seule *nappe* limite son utilisation.

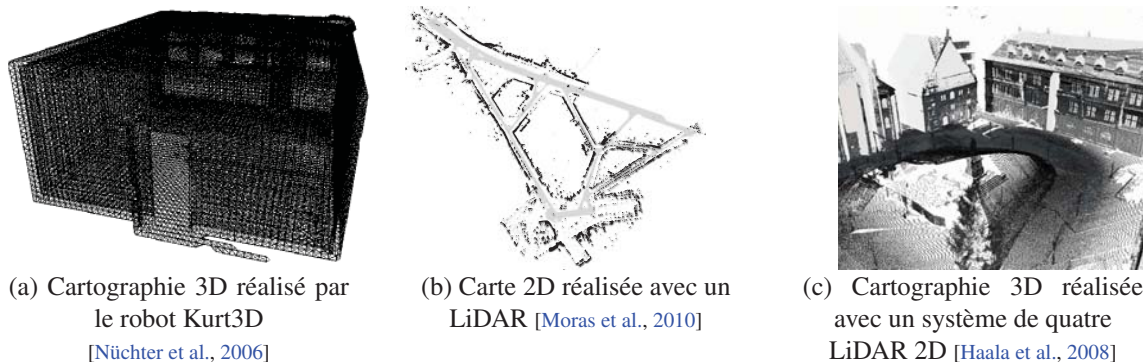


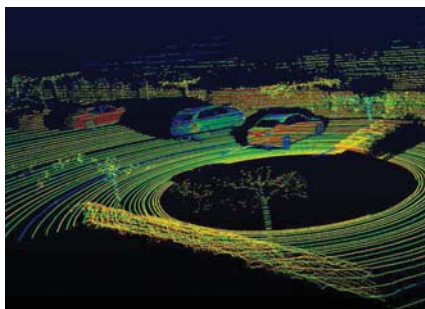
FIGURE II.4 – Cartographie de l'environnement réalisée par un LiDAR

Dans le projet *Kurt3D* [Nüchter et al., 2006], une plateforme robotique mobile est équipée d'un LiDAR monocouche (figure II.4(a)) placé sur un dispositif motorisé permettant le changement du

1. Light Detection And Ranging; 2. Light Amplification by Stimulated Emission of Radiation;

plan de balayage du LiDAR. Ainsi on acquiert un nuage de points 3D qui n'a pas de sémantique. Grâce à des algorithmes de partitionnement de données¹, on peut regrouper et classifier les points appartenant à des objets [Gunther et al., 2013].

L'ajout d'un système de prismes ou de miroirs à un LiDAR 2D ou l'intégration de plusieurs LiDAR 2D dans un même dispositif, permet de scanner un espace en trois dimensions. Le principal constructeur des LiDAR 3D multi-nappes est l'entreprise Velodyne [Velodyne LiDAR, 2016] qui propose aujourd'hui plusieurs produits de très haute précision (figure II.5(b)) comme HDL-64E [Halterman and Bruch, 2010], HDL-32E ou le plus compact et le plus accessible en terme de prix : le PUCK. Généralement, on nomme chaque plan de balayage, *nappe* et à ce jour le capteur le plus complexe en compte 64. En 2009 le HDL-64 coûtait 75 000 \$ et en 2016 son prix est toujours supérieur à 50 000 \$. On trouve des versions moins performantes et également moins chères comme *Velodyne PUCK*, mais son prix de 8 000 \$ est toujours élevé pour une intégration en grande série sur des véhicules. D'autres constructeurs, comme *Quanergy* [Quanergy, 2016], *IBEO* [IBEO ADAS, 2016], *SICK* [SICK, 2016] ou *Hokuyo* [Hokuyo, 2016] qui proposent des produits à bas prix et moins complexes avec 1 à 5 nappes. Un prix si élevé s'explique surtout par le faible nombre de produits fabriqués. L'entreprise *Quanergy* annonce la commercialisation d'un *3D Solid State LiDAR*² dès 2017. D'après *Quanergy*, sa production en masse permettra une baisse du prix jusqu'à 250 €, ce qui rendrait l'utilisation envisageable à grande échelle.



(a) Données brutes Velodyne HDL-64E



(b) LiDAR Velodyne : HDL-64E, HDL-32E, PUCK

FIGURE II.5 – Capteurs LiDAR Velodyne [Velodyne LiDAR, 2016]

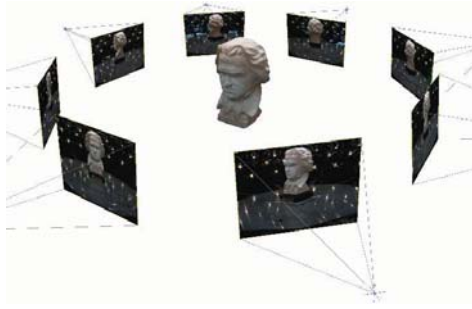
Les données brutes issues de ce type de système sont des ensembles des distances d'impact avec les objets et les positions du faisceau laser lors de l'impact; ceci revient à réaliser un nuage de points comme dans la figure II.5(a). Un premier défi est de réaliser une bonne calibration du scanner [Zalud et al., 2015] et de s'assurer de la qualité de la perception. Ces données en nuages de points trouvent leur utilité dans des applications comme la détection et la classification des objets, par exemple les piétons [Misu and Miura, 2016]. L'intégration du mouvement du capteur 3D dans le temps et l'association des paires de points tout au long de l'enregistrement [Douillard et al., 2012] permet la réalisation d'une carte de points 3D [Schwarz, 2010]. L'association d'un nuage de points courant à une carte précédemment apprise revient à faire de la localisation par rapport à cette carte. Le principe porte le nom de SLAM³, en français *cartographie et localisation simultanées*, et comme son nom l'indique, a pour objectif de créer une carte de l'environnement tout en essayant de localiser le capteur dans cette carte [Moras et al., 2010, 2011; Moosmann and Stiller, 2011] (figure II.4(b)).

Vision monoculaire

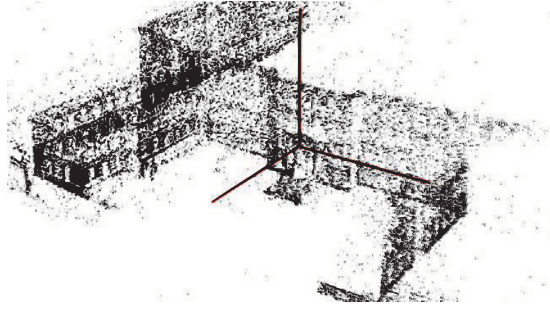
En utilisant une seule caméra, on peut également percevoir l'environnement et les distances,

1. Clustering en anglais; 2. LiDAR sans composants mécaniques en mouvement; 3. Simultaneous Localization And Mapping;

mais il faut que la caméra soit en mouvement et que le mouvement soit connu (ou estimé). Même si le premier capteur silicium d'image est conçu en 1969, les premiers algorithmes dans le domaine de la vision par ordinateur sont proposés uniquement dans les années 1980. C'est alors la «méthode Lucas–Kanade» de flux optique qui est développée et dont l'idée essentielle est l'hypothèse que chaque point d'intérêt se retrouve dans l'image suivante dans une zone de voisinage imposée ou calculée en fonction de la tendance du mouvement [Lucas et al., 1981; Lucas, 1985].



(a) Cartographie 3D d'un objet [Kolev et al., 2012]



(b) Nuage de points 3D d'un bâtiment [Cohen et al., 2012; Sattler et al., 2015]

FIGURE II.6 – Reconstruction 3D de l'environnement utilisant plusieurs vues

Quand on se réfère à la reconstruction 3D de l'environnement, on se rapporte implicitement au calcul de la position de la caméra, donc aux techniques de SLAM¹ (localisation et cartographie simultanée) [Féraud et al., 2011b; Bresson et al., 2012, 2015] et de SfM² [Moulon et al., 2012; Song and Chandraker, 2014]. Dans la figure II.6(a) on présente le principe de la reconstruction dense 3D d'un objet [Kolev et al., 2012] et dans la figure II.6(b), le nuage de points correspondant à la structure des bâtiments d'une ville [Cohen et al., 2012; Sattler et al., 2015]. Pour ce faire, plusieurs images sont prises avec un champ de vue recouvrant. Nous allons revenir sur cette problématique dans ce mémoire dans la section §IV.4 dédiée à la détection d'obstacles par approche *Structure from Motion*.

Stéréovision

La stéréovision repose sur un principe de *stéréoscopie*³, déjà connu au XVI^{ème} siècle par le peintre Jacopo Chimenti (1554 - 1640). Le premier stéréoscope est construit deux siècles après, en 1838 par le physicien anglais Charles Wheatstone [Wheatstone, 1838, 1852].



(a) Le sténoscope au XIX^{ème} siècle



(b) Anaglyphe réalisée par Phillips en 1923

FIGURE II.7 – La stéréoscopie

1. Simultaneous Localization And Mapping; 2. Structure from Motion; 3. Du grec *stéréo* : solide et *scope* : vision; «Ensemble des principes qui régissent l'observation binoculaire et sa reproduction» [Larousse];

Aujourd'hui cette technique est largement utilisée pour la réalisation des films 3D mais pas uniquement. Le principe de percevoir la profondeur grâce à deux images prise depuis deux points de vue décalés est exploitée par la communauté de traitement d'image¹ et de plus en plus utilisé dans l'industrie automobile pour la détection des obstacles. Un exemple représentatif d'utilisation de la stéréovision est dans le véhicule prototype *Mercedes - Bertha* [Ziegler et al., 2014] présenté dans le chapitre I [page 10]. Dans la figure II.8(a) on retrouve un modèle de stéréo caméra installée sous le pare-brise du véhicule. La figure II.8(b) présente une image de profondeur de la scène synthétisée par la méthode SGM² [Hirschmüller, 2005; Gehrig et al., 2009]. Pour réaliser cette image, 400 000 appariements sont faits pour chaque paire d'images à une fréquence de 25 Hz. Ceci est possible grâce à la puissance des FPGA³. Dans la figure II.8(c) on retrouve une représentation 3D d'un nouveau concept nommé *stixel* [Franke et al., 2005; Pfeiffer and Franke, 2010, 2011; Franke et al., 2013] dont l'objectif est de diminuer la quantité de calcul tout en préservant la qualité des résultats. Le suivi dans le temps des *stixels* permet une segmentation et un isolement des éléments dynamiques de l'environnement statique (figure II.8(d)) [Erbs et al., 2012].

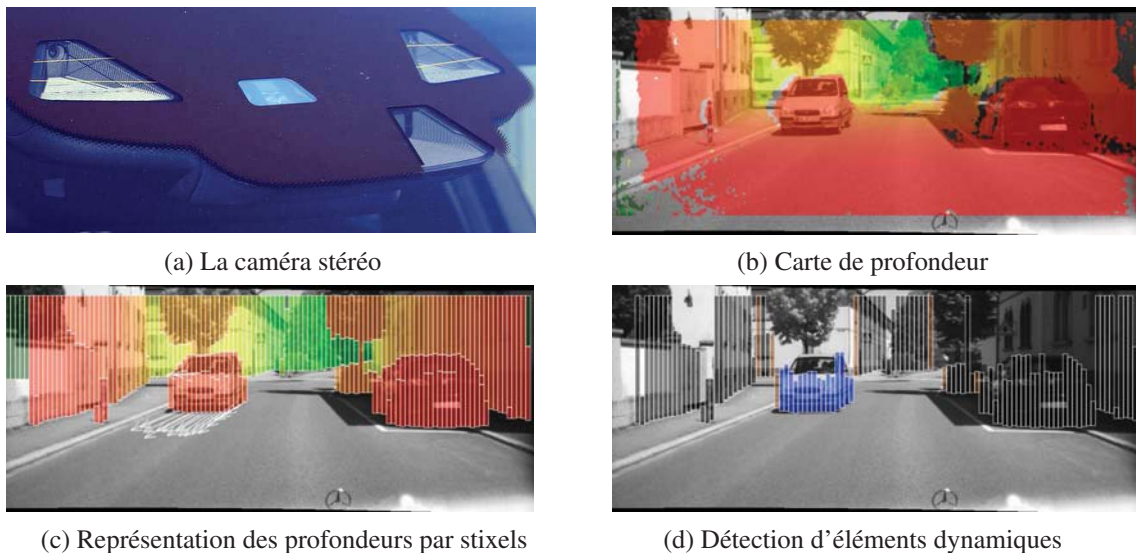


FIGURE II.8 – La stéréovision et le processus d'extraction de la profondeur et des éléments dynamiques dans la scène utilisé par Mercedes dans leur véhicule prototype *Mercedes-Bertha* [Franke et al., 2005; Pfeiffer and Franke, 2010, 2011; Franke et al., 2013]

Même si la communauté scientifique de traitement d'image a débuté le développement des techniques de monovision et stéréovision en même temps, la stéréovision s'est déployée plus rapidement et plus fortement dans l'industrie automobile. L'avantage majeur de la stéréovision consiste dans la possibilité d'avoir une information sur la profondeur de la scène à échelle connue, même quand la caméra ne bouge pas. En ce qui concerne la mono-vision, la reconstruction de la scène se réalise par des technique de SfM⁴ qui exigent un déplacement de la caméra en plus du suivi des différents points d'intérêt au long du déplacement. De plus, avec une seule caméra, le résultat de l'algorithme SfM est un nuage de points 3D mais les distances entre les points sont calculées à un facteur d'échelle près. Cet inconvénient est facilement contournable sur les véhicules car la connaissance de l'information odométrique permet de calculer le facteur d'échelle. Ces problématiques seront détaillées dans la section §II.3.2 dédiée à la perception par caméra.

1. *Computer vision*, en anglais; 2. *Semi-Global Matching*; 3. *Field-Programmable Gate Array*; 4. *Structure from Motion*;

Systèmes multi-caméras

Des versions des systèmes avec plus de deux caméras, avec champ recouvrant ou non, sont proposées (figure II.9(a)) et utilisées, entre autres, pour la détection des obstacles et la reconstruction de scènes [Litvinov et al., 2013]. Dans l'industrie automobile on propose déjà des systèmes à 4 caméras et une recombinaison des images pour une vue à 360° autour de la voiture (*birdview* en anglais) - comme dans la figure II.9(b) où on observe une interface standard d'un véhicule haut de gamme Volkswagen.



(a) Systèmes multi-caméras



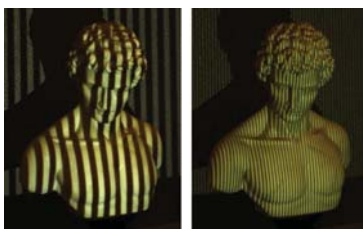
(b) Vue 360° autour de la voiture réalisé par synthèse d'images issues de 4 caméras grand angle (*birdview*)

FIGURE II.9 – Systèmes multi-caméras pour l'élargissement du champ de vue

Nous considérons que dans les systèmes d'aide à la conduite un champ de perception à 360° sera nécessaire. Cette perception pourrait être faite à terme en utilisant des caméras ou d'autres dispositifs de perception. Les inconvénients majeurs de la caméra sont la difficulté et la complexité d'interprétation de l'environnement car l'extraction de l'intelligence des images doit être réalisée avec des algorithmes fiables dont la puissance de calcul requise ne doit pas dépasser celle disponible dans les calculateurs embarqués dans les véhicules. Par contre son énorme avantage est le prix réduit et la richesse de l'information contenue dans chaque image. C'est exactement cette richesse d'information et les progrès faits tous les jours dans le domaine de *computer vision* qui soutiennent notre conviction que la plupart de l'information liée à l'environnement proviendra de la caméra, même dans des conditions de faible éclairage.

Système de perception à base de lumière structurée

Un autre moyen de percevoir l'environnement et les profondeurs est l'utilisation de l'*éclairage actif* ou de la *lumière structurée* [Lanman et al., 2007, 2009]. On projette des franges parallèles de lumière comme dans la figure II.10(a) et, avec une caméra, on capte une image de l'objet partiellement illuminé. La déformation des lignes de lumière permettra de ressortir la forme 3D de l'objet ou l'environnement considéré.



(a) Processus de scan multi-résolution par lumière structurée [Lanman et al., 2007, 2009]



(b) Gamme de produit *LMI Technologie* pour la cartographie 3D [LMI Technologies, 2016]

FIGURE II.10 – Utilisation de la lumière structurée pour la cartographie 3D des objets

Sur le marché, on retrouve des dispositifs [LMI Technologies, 2016] (figure II.10(b)) qui permettent la cartographie d'objets avec des résolutions centimétriques ou même submicroniques, mais pour des prix très élevés et avec l'inconvénient supplémentaire que ces dispositifs ne peuvent pas être utilisés sur des scènes dynamiques.

Caméra ToF

Une des dernières technologies pour la perception de l'environnement et les distances aux obstacles, déployée sur les applications civiles au début des années 2000, est la caméra ToF¹. Une image résultante est appelée image RGB-D² contient pour chaque pixel l'information de la couleur et une distance calculée par le temps de vol de la lumière. Pour ce faire, on éclaire la scène avec des flashes de lumière en spectre infrarouge, à fréquence constante et on mesure le temps que la lumière met pour éclairer l'obstacle.



(a) Reconstruction 3D

[Khoshelham and Elberink, 2012]



(b) Classification et labellisation d'objets [Wu et al., 2014]

FIGURE II.11 – Extraction d'objets des images RGB-D et cartographie 3D de l'environnement

Plusieurs projets utilisent ce type d'image en entrée et le capteur le plus connu est la célèbre Kinect de Microsoft [Khoshelham and Elberink, 2012]. Dans la figure II.11(a) on retrouve le résultat de la reconstruction 3D d'une pièce utilisant des images RGB-D [Audras et al., 2011]. Dans la figure II.11(b) sont illustrées la détection d'objets et leur classification utilisant une technique d'arbre sémantique [Wu et al., 2014]. L'inconvénient majeur de ces systèmes est que, pour obtenir de bons résultats, la zone de détection des obstacles est uniquement de 0,5 m à 5 m, elle peut être étendue à 10 m avec une baisse considérable de la qualité de détection de la profondeur à cause de l'éclairage parasite. De plus, à ce jour, leur utilisation est adaptée à l'environnement intérieur, ce qui rend ce type de capteurs pour l'instant inappropriés pour notre application.

II.2.1.3 Bilan sur les capteurs

Dans la section précédente nous avons passé en revue différents capteurs. Même si les LiDAR affichent des résultats très bons en termes de qualité de perception de l'environnement, leur prix les rend inutilisables en série à ce jour. En plus de son prix, le capteur présente l'inconvénient d'avoir un gabarit très imposant par rapport aux autres capteurs utilisés aujourd'hui sur un véhicule, ce qui rend son intégration difficile. De plus, à cause de l'utilisation d'un mécanisme rotatif de balayage, le phénomène de distorsion de l'acquisition devient très important avec l'augmentation de la vitesse du véhicule et la complexité mécanique rend le capteur fragile vis-à-vis des chocs et des vibrations. Un autre problème que l'on peut citer, est le *speckle*³ [Fix et al., 2014] qui se traduit par l'influence de la surface sur laquelle l'onde se réfléchit, sur la qualité de la détection. Un

1. Time of Flight; 2. «D» pour Depth; 3. «Tavelure» ou «granularité» en français, car l'image donne l'impression d'être formée de grains, phénomène dû au nombreuses petites impuretés dans le milieu de propagation;

dernier problème qui n'est pas négligeable vient du fait que la surface du plexiglas de protection doit être en permanence très propre car toute impureté influe sur la qualité et la direction du rayon laser, ce qui est évidemment incompatible avec une utilisation pour l'automobile. L'annonce réalisée par l'entreprise *Quanergy* sur la commercialisation dès 2017 du *3D Solid State LiDAR* à uniquement 250€ relance le débat de l'utilisation des LiDAR dans les systèmes d'aide à la conduite sur les véhicules de série.

Une deuxième classe de systèmes de perception de distances, dont nous avons déjà parlé, sont les systèmes qui utilisent la lumière structurée. Même si ceux-ci montrent également une qualité de perception élevée, ils sont utilisables, à ce jour, sur des scènes statiques et en environnement intérieur avec un niveau contrôlé de l'intensité de l'éclairage.

	RADAR			Ultrason	LIDAR		Caméra				Système à lumière structurée
	SRR	MRR	LRR	SCU	2D	3D	Caméra	RGB-D	Stéréo	Multi	
Prix	o	o	o	++	--	--	++	o	o	o	--
Distance perception <4m	++	+	o	++	++	++	+	++	++	++	++
Distance perception 4 m–40 m	+	++	+	-	++	++	+	o	++	++	-
Distance perception 40 m–200 m	-	o	++	--	o	o	o	--	+	+	--
Champ visuel - 2D	o	o	o	+	++	++	++	++	++	++	++
Champ visuel - 3D	o	-	-	+	-	++	++	++	++	++	++
Calcul vitesse	++	++	++	-	o	o	o	o	o	o	--
Fonctionnel par pluie	++	++	+	o	o	o	-	-	-	-	--
Fonctionnel par brouillard et neige	++	++	+	o	-	-	-	-	-	-	--
Fonctionnel si sale	+	+	+	+	--	--	--	--	--	--	--
Vision nocturne	++	++	++	++	++	++	--	++	--	--	++

TABLE II.1 – Comparaison de capteurs extéroceptifs pour notre application (++ : idéal; + : bon; o : possible, mais avec des inconvénients à prévoir; - : très difficile, mais possible; -- : impossible)

En ce qui concerne le RADAR, certains véhicules de série en disposent déjà. En fonction de leurs capacités et de leurs distance de détection, les RADAR sont classifiés en longue (LRR¹), moyenne (MRR²) et courte portée (SRR³). Même si ces capteurs sont utilisés pour la détection des vitesses des objets, tels qu'ils sont conçus aujourd'hui pour l'industrie automobile, ils présentent l'énorme inconvénient de ne pas détecter de manière très fiable les objets fixes. Dans certains cas d'utilisation, comme dans les systèmes de régulation de la vitesse en fonction du véhicule précédent, cette limite ne se manifeste pas dès lors que le véhicule en face n'est pas arrêté, mais pour des situations de parking où la plupart de l'environnement est fixe ou à l'arrivée à un feu rouge, le RADAR n'est qu'à ses débuts d'utilisation. Pour palier le problème, les RADARs sont utilisés conjointement avec d'autres capteurs comme des caméras [Bombini et al., 2006; Wu et al., 2009] qui assurent la redondance ou la complémentarité des informations.

Un capteur largement répandu sur les véhicules de série est le capteur ultrason. Nous allons utiliser un système composé de 12 capteurs disposés sur le bouclier avant et arrière du véhicule et nous allons exploiter ce système pour tirer le maximum d'information sur l'environnement d'évolution du véhicule. Ce système de capteurs sera détaillé dans la section §II.3.1.

Nous avons précédemment présenté des systèmes de détection par vision, avec une caméra, deux (stéréovision) ou plusieurs. Dans notre application de voiturier autonome, nous ne pouvons

1. Long-Range Radar sensor; 2. Mid-Range Radar sensor; 3. Short-Range Radar sensor;

pas nous dispenser de l'utilisation de la caméra, car elle présente l'avantage d'avoir un prix abordable et de nous fournir le plus d'information sur l'environnement (voir comparatif des capteurs dans le tableau II.1). Dans la section §II.3.2 nous justifierons notre choix d'utiliser une seule caméra, nous traitons en détails la manière dont la caméra est utilisée pour la détection des obstacles. Enfin, ensuite, dans le Chapitre III, nous abordons les techniques de localisation du véhicule également basées sur la vision.

II.2.2 Cartographie

II.2.2.1 Introduction

L'abstraction de la réalité par l'utilisation d'une représentation graphique du monde réel est souvent utilisée sous le nom de *carte*. De nos jours, les cartes sont devenues des outils indispensables pour l'orientation dans l'espace et pour la planification. Il n'est pas rare qu'on dise qu'*une image vaut mille mots* [Confucius]. Tous ceux qui les ont déjà utilisées savent qu'elles contiennent beaucoup d'informations utiles que les mots ne peuvent pas facilement traduire, dont des informations caractérisant la traversabilité de l'espace. Même si la création des cartes demande des compétences et des outils spécifiques (topographie), la lecture de ces cartes ne demande que de connaître certaines conventions et symboles. Utiliser des cartes revient à reconnaître des symboles comme des lignes et des couleurs différentes pour montrer des caractéristiques telles que les rivières, les routes, les villes ou les montagnes. La plupart des cartes nous aident aussi à connaître les distances entre ces éléments grâce à la lecture de l'échelle.

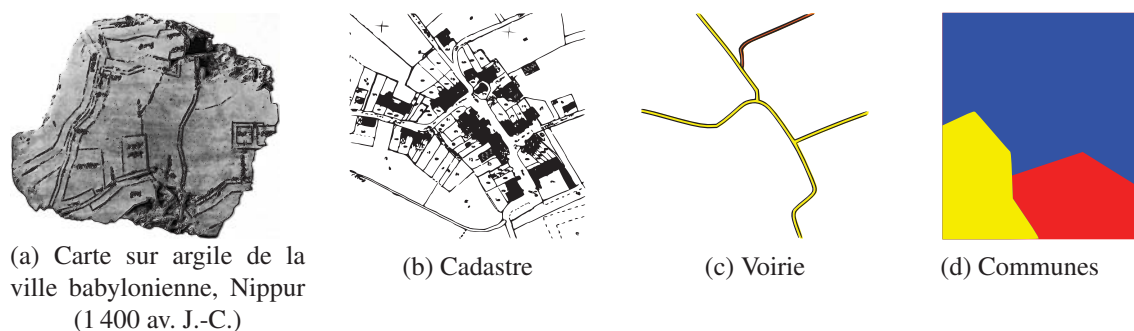


FIGURE II.12 – Cartes à différents niveaux de détails

Les archéologues et les historiens ont du mal à s'accorder sur la datation de la première carte terrestre. On retrouve une représentation contenant une montagne, des vallées, une rivière et des routes datant de 25 000 av. J.-C. de la zone autour de Pavlov en République Tchèque. On sait déjà que la première carte des étoiles date de 16 500 av. J.-C. et a été trouvée dans la grotte de *Lascaux* en France [Smith, 2004; Cunningham, 2011]. Par contre la première carte précise a été trouvée en 1930 et représente la ville Nippur de l'ancienne Babylonie et pour sa création, des techniques d'arpentages¹ précises ont été utilisées. On peut retrouver une grande ressemblance entre la figure II.12(a) représentant la ville Nippur et la figure II.12(b) représentant un plan cadastral d'une ville moderne. Comme on peut l'observer dans les figures II.12(b) à (d), la même zone est représentée par des niveaux de détails différents [Antoine, 2014]. Des systèmes de géocodage nommés SIG², ou GIS³ en anglais, ont été conçus pour manipuler des données alphanumériques qui décrivent les cartes.

1. «Technique de la mesure des éléments géométriques des parcelles de terrain» [Larousse]; 2. Système d'Information Géographique; 3. Geographic Information Systems;

Les informations sauvegardées sont l'adresse postale, les coordonnées GPS ou bien les différentes formes géométriques des bâtiments et des routes [Aronoff, 1989; Foresman, 1998; Jakubiec and Reinhart, 2013].

II.2.2.2 Cartographie dans le domaine automobile

En revenant à notre sujet central, nous nous sommes intéressés à la cartographie pour réaliser la fonction de localisation et de traversabilité de notre véhicule autonome. Dans ce sens, on sait que la communauté scientifique s'est d'abord intéressée à la façon dont les humains et les animaux se localisent dans l'espace. Dans les travaux de Tolman [1948] on retrouve déjà une analyse de la *fonction spatiale* des rats et des humains. Campan and Scapini [2002] s'intéressent également à cette problématique et dans *Étymologie : approche systémique du comportement* ils dédient un chapitre à la *Cognition spatiale chez les mammifères* avec des détails sur les représentations mentales qu'un individu se fait de l'organisation de l'espace dans lequel il se trouve (représentations qu'ils nomment *cartes cognitives*).

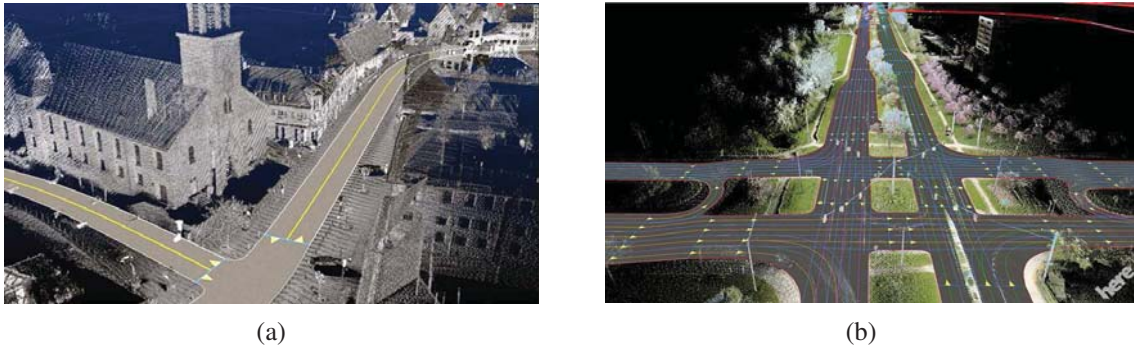


FIGURE II.13 – Exemples de cartes de haute définition *Nokia HERE* qui combinent de détails de l'environnement avec d'informations de flux de trafic et de voies de circulation

La construction des cartes riches en informations destinées aux systèmes d'aide à la conduite ou de véhicule autonome est une tendance forte. Cette technique permettra de contourner le problème d'interprétation de l'environnement en temps réel par une déportation de l'intelligence du véhicule dans les cartes.

Le fait que le projet de développement de technologies pour le véhicule sans conducteur, *Google Car*, soit mené par Sebastian Thrun n'est pas une coïncidence. Celui-ci est le directeur du projet *Stanley* (gagnant du *DARPA*¹ *Grand Challenge* en 2005 [Thrun et al., 2006a; Levinson et al., 2011]), le fondateur de *Google X* et le co-inventeur de *Google Street View*. A ce jour, la stratégie de navigation du véhicule autonome annoncée par Google est d'utiliser des cartes riches en information. En analysant l'ampleur du projet *Google Street View* et sa couverture internationale, nous pouvons affirmer que la cartographie va jouer un rôle important dans le bon fonctionnement du véhicule autonome du futur.

A titre d'exemple, Nokia se lance dans la collecte de l'information et annonce que plus de 100 milliard de points de données GPS sont recensés par mois. Avec ces données les trajectoires et les vitesses des véhicules peuvent être reconstruites. Dans le cadre du projet *Nokia HERE*, une fusion de ces données GPS se fait avec des données LiDAR pour la construction de cartes de l'environnement en haute résolution. Dans les figures II.13(a) et (b) on retrouve des exemples

1. Defense Advanced Research Projects Agency ;

de cartes résultantes qui ont vocation à être utilisées dans des projets d'aide à la conduite et de véhicule autonome.



(a) Cartographie haute précision des voies de circulation



(b) San Francisco Bay - zone cartographié en haute résolution par la flotte Tesla

FIGURE II.14 – Collecte de données de navigation à travers de toute la flotte de véhicules Tesla et création de carte de haute précision

Sur le même principe, Tesla collecte des données de navigation au travers la flotte de véhicules et crée une carte de haute précision (figures II.14(a) et (b)). Les données enregistrées par chaque caméra et GPS de chaque véhicule sont envoyées par internet à un service Cloud qui collecte ces informations relatives à 2 millions de km par jour, voir section §II.2.3 sur la communication V2X. Des algorithmes d'apprentissage par *Machine Learning* traitent ces données et réalisent des améliorations de la carte et des fonctions d'aide à la conduite. Tesla annonce également que cette carte de haute précision permet au véhicule de se guider uniquement en utilisant le GPS dans des conditions de visibilité réduite quand les marquages au sol ne sont pas visibles (pluie, neige).

II.2.2.3 Représentation des cartes

En plus de la détection d'obstacles, la gestion de l'information en elle-même est aussi importante. En fonction de leur représentation interne on retrouve trois types de cartes :

- **cartes métriques** - elles sont les plus employées dans la robotique; elles contiennent de l'information directement utilisable, car les objets sont placés à des coordonnées connues. L'inconvénient de ce type de carte est leur sensibilité à la précision du placement ou aux dimensions des objets (figure II.15(a)). Un deuxième inconvénient est qu'il n'y a pas de sémantique ou de classification liée aux objets;
- **cartes topologiques** - ce sont des cartes qui contiennent des ensembles d'objets reliés par des opérations de connectivité, de proximité, d'ordre ou d'inclusion (figure II.15(b));
- **cartes métriques et topologiques** - ce sont des cartes mixtes (figures II.15(c) et (d)) qui contiennent aussi bien des informations topologiques que métriques [Thrun, 1998; Filliat, 2011; Corcoran and Mooney, 2013]. Souvent dans la littérature elles sont appelés uniquement cartes topologiques, considérant la métrique comme un des éléments constituants.

Pour la représentation de ces cartes, les éléments qui les composent sont des points, des lignes, des polygones ou des collections de ces éléments (par exemple dans la figure II.16, on retrouve les données de l'indexation spatiale tel que proposé par *SQL Server 2008 R2*). L'information spatiale est sauvegardée soit avec des *données précises* (dans une représentation vectorielle),

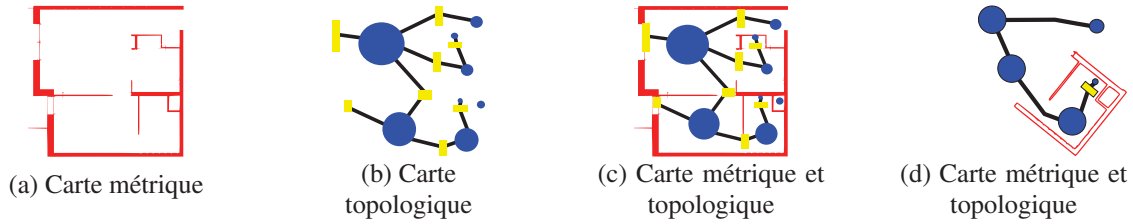


FIGURE II.15 – Exemple de carte topologique, métrique et hybride pour un environnement intérieur [Filliat, 2011]

soit par des *approximations*. L'approximation la plus courante est la *boîte englobante* ou REM¹, connue aussi sous le nom de *boîte limite* ou *enveloppe*.

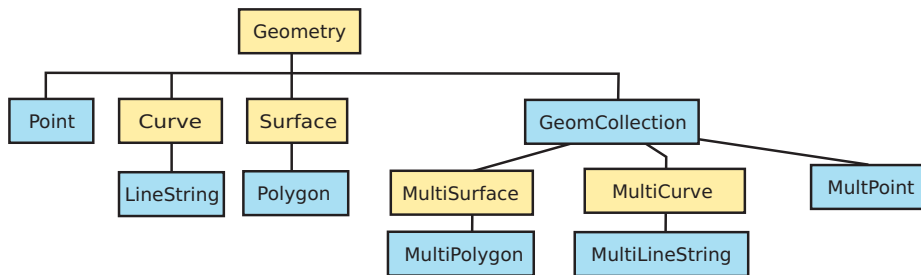


FIGURE II.16 – Types de données pour l'indexation spatiale (structure utilisée dans [SQL Server 2008 R2, 2008]). En bleu les types simples et en jaune les types *collection*

Un des outils le plus utilisé pour stocker les données cartographiques est le système *SQL Server*. Par exemple dans [SQL Server 2008 R2, 2008], les index spatiaux sont construits à l'aide d'arbres B (B-trees), par conséquent une décomposition uniforme hiérarchique de l'espace est implémentée avant de lire des données dans un index spatial. Le processus de création d'index, décompose l'espace en une hiérarchie de grille à quatre niveaux (figure II.17(b)).

SQL Server a fait le choix d'utiliser les *B-trees* parmi une longue liste des techniques de cartographie. Pour manipuler des données cartographiques (spatiales) dans la littérature scientifique on retrouve le terme *d'indexation spatiale* que nous allons utiliser par la suite.

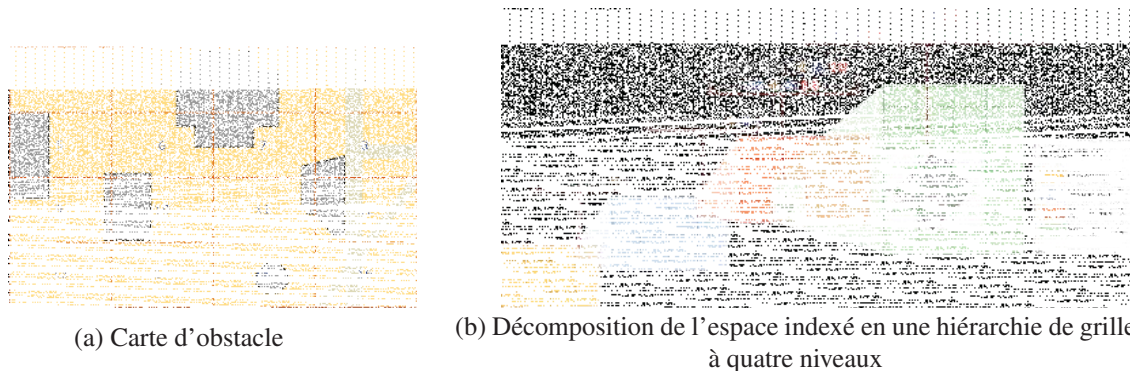


FIGURE II.17 – Indexation spatiale utilisé dans [SQL Server 2008 R2, 2008]

Le processus d'indexation spatiale a été conçu pour gagner en efficacité (temps et espace) lors de la recherche d'objets (points, polygones) et également pour améliorer le processus de mise

1. L('hyper) Rectangle Englobant Minimum ou **MBR** (Minimum Bounding Rectangle), en anglais;

à jour de la base de données. Pour ce faire, on retrouve deux grandes catégories d'index : l'index *space-driven*, où le partitionnement de l'espace est indépendant de la collection d'objets spatiaux et l'index *data-driven*, pour un partitionnement de l'espace dépendant de la distribution des données.

Les techniques d'indexation sont très variées, comme la *grille régulière* pour les points, la *Grid File*, l'indexation de rectangles (REM¹) avec une grille, la grille alphanumérique, la *geodesic grid* [Ringler et al., 2000; Sahr et al., 2003]. Mais les plus utilisées sont les structures en arbre : *Region Quadtree*, *Point Quadtree*, Octree, Z-order tree, k-d-b-tree, R⁺-tree, BSP²-tree, R-tree, LSD-tree, X-tree, SR-tree, A-tree, M-tree, Pyramid tree, etc.

Ce dont nous avons besoin pour pouvoir accomplir la fonction de traversabilité est une carte des obstacles qui signale l'espace occupé. Ainsi, la notion d'espace traversable sera incluse dans la carte. Nous reviendrons sur le sujet dans la section §IV.2.1 pour détailler la carte métrique de type grille avec ses avantages et inconvénients tout en donnant des travaux de référence issus de la littérature scientifique.

Par la suite nous nous intéressons à la communication, comme moyen d'acquisition de l'information et également d'enrichissement des cartes.

II.2.3 Communication V2X

Une manière de connaître l'environnement est d'exploiter la communication. Les flottes des robots autonomes commercialisées par Balyo [Balyo, 2016] et présentées succinctement dans la section §I.1 sont un exemple concret d'utilisation de la coordination centralisée des robots. Seule l'unité centrale connaît la position de tous les robots et les couloirs de circulation occupés. Le transport ferroviaire utilise ce principe depuis toujours. En effet, on constate que lors d'un problème de signalisation de l'occupation de la voie, la vitesse très élevée des trains fait que, même si les conducteurs voient le train arrivant en face sur la même voie, il est déjà trop tard pour éviter l'accident, d'où l'intérêt d'une alerte préalable. Dans l'industrie automobile, les systèmes V2X³ (figure II.18(a)) sont des systèmes de communication sans fil qui cherchent à améliorer la sécurité routière et la fluidité du trafic. Dans ce cas, ce sont des tiers qui envoient de l'information pour alerter ou signaler certains événements. Par exemple lors de l'utilisation d'une application de navigation, l'état du trafic sur le trajet est reçu via une connexion 3G/4G, permettant ainsi le calcul des trajets alternatifs plus rapides.

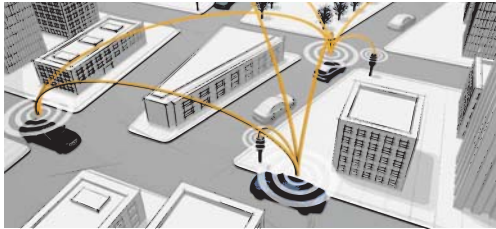
La communication V2X englobe les systèmes de communications suivants :

- V2V⁴ – systèmes de communication sans fil entre véhicules ;
- V2I⁵ - systèmes de communication sans fil entre véhicules et l'infrastructure.

Un véhicule équipé d'un tel système, lors de la détection d'une situation dangereuse, sera capable d'envoyer cette information aux autres véhicules se trouvant dans son alentour. Sinon il peut communiquer avec les balises installées au bord de la route, celles-ci prendront en charge l'acheminement des messages vers le centre de gestion ou vers d'autres véhicules. Une autre utilisation de la communication V2X est pour des applications de *trains de véhicules* ou *platoon* en anglais. Par la diminution de la distance entre véhicules, le coefficient de frottement global est réduit et donc la consommation de carburant [§I.3.1]. Au même titre, cela représente une fonction

1. L(hyper)e Rectangle Englobant Minimum; 2. Binary Space Partitioning; 3. Vehicle-to-Everything; 4. Vehicle-to-Vehicle; 5. Vehicle-to-Infrastructure;

de délégation de conduite parce que c'est le véhicule en tête qui aura une responsabilité partielle ou totale du train de véhicules [Dávila and Nombela, 2010; Chan, 2014; Poorsartep and Stephens, 2015; Shladover et al., 2015].



(a) Communication V2X : V2I & V2V



(b) Le projet *Safe Road Trains for the Environment* [Dávila and Nombela, 2010; Chan, 2014]

FIGURE II.18 – Communication V2X

Un autre exemple de système V2X a été réalisé au travers du projet AMULETT¹, en partenariat entre l'industriel *BMW AG* et le *Ministère bavarois de l'Économie, des Médias, de l'Énergie et de la Technologie*², qui permet au véhicule de communiquer via un émetteur radio porté par des enfants (figure II.19). Ainsi, même si les enfants ne sont pas dans le champ de vision de la caméra, ou si le champ de vision est obstrué, le système pourra détecter leur position, estimer leurs trajectoires et agir par un freinage d'urgence si nécessaire [Lill et al., 2010]. Ce projet, même s'il est très efficace du point de vue technologique, n'a jamais été accepté, car devoir toujours porter un émetteur était considéré comme une atteinte à la vie privée même si on sait bien que beaucoup d'enfants disposent de smartphones qui connaissent en continu leur position GPS.



FIGURE II.19 – Le projet V2X *AMULETT* porté par l'entreprise *BMW* [Lill et al., 2010]

Comme nous l'avions présenté dans la section §II.2.2, la communication est utilisée également pour la mise à jour des cartes et dans le partage des informations pour une flotte de véhicules. L'information perçue à un moment donné par un véhicule et remontée dans un serveur pour un post-traitement sera réutilisée par un autre véhicule à un moment ultérieur.

Dans les travaux de G. Bresson on retrouve une autre application de la communication, cette fois pour la «localisation d'une flotte de véhicules communicants par approche de type SLAM visuel décentralisé» [Bresson, 2014]. On retrouve ici, une technique de localisation de chaque véhicule de la flotte par filtre de Kalman étendu avec un enrichissement de la perception par des échanges permanents via la communication sans fil. De plus une structure générique est présentée afin de permettre le remplacement de l'algorithme de SLAM par un autre, tout en tenant compte des éventuelles latences ou coupures de communication.

1. *Aktive Mobile Unfallvermeidung und Unfallfolgenminderung durch kooperative Erfassung und Trackingtechnologie*;
 2. *Bayerische Staatsministerium für Wirtschaft und Medien Energie und Technologie*;

II.2.4 Détection et suivi d'obstacles dans le domaine routier

En ce qui concerne la détection de la zone traversable, on considère qu'il existe deux grandes familles de techniques : celles qui utilisent des cartes et les autres qui n'en utilisent pas. L'utilisation ou non de la carte va nous diriger également vers deux approches en fonction de l'existence ou pas d'une connaissance globale ou locale sur l'environnement d'évolution. Ces deux techniques de planification de trajectoire s'appelleront planification globale et locale de trajectoire et seront détaillées dans le chapitre V.



FIGURE II.20 – Systèmes de sécurité active et d'aide à la conduite

La première technique, qui ne nécessite pas d'utiliser des cartes, demande à réaliser du traitement de l'information de l'environnement au fur et à mesure qu'elle est acquise, sans pour autant pouvoir la comparer à une référence. Cette technique est très utilisée dans les ADAS pour des environnements faciles à interpréter comme les autoroutes ou les routes nationales. Le système du type LCA¹ utilise les lignes blanches pour se centrer de la voie et le système ACC² utilise la distance au véhicule en face pour réguler sa vitesse [Hillel et al., 2014; Grove et al., 2015] (voir figure II.20(a)). A titre d'exemple le système AEB/AEBP détecte la distance aux obstacles et actionne le freinage d'urgence en cas de collision imminente [Jang et al., 2013; Kim et al., 2015] (voir figure II.20(b)). Les avancées technologiques et le choix de ce cadre limité permettent aux constructeurs automobiles d'apporter en 2016 des solutions de conduite autonome de niveau 1 et 2 [§I.3.2], sans pour autant avoir besoin d'attendre le développement de systèmes d'intelligence artificielle qui permettront de conduire le véhicule comme un humain.



FIGURE II.21 – Détection des lignes de marquage par temps de jour et de nuit

Dans cette approche il existe principalement deux catégories d'éléments que le véhicule doit détecter : la route et les obstacles (statiques ou dynamiques). Détecter la route peut se réaliser à plusieurs niveaux d'interprétation, mais le plus courant est de détecter les lignes de marquage [Bertozzi and Broggi, 1998; Aufrère et al., 2001; Aly, 2008] (figures II.21(a) et (b)) et de faire l'hypothèse que ces lignes appartiennent à la route. Cette approche trouve rapidement ses limites sur des routes où le marquage est partiellement ou complètement effacé. Dans ce cas, les techniques de détection de la route [Fernandez et al., 2015; Casapietra et al., 2015] s'avèrent plus utiles, comme présenté dans la figure II.22.

1. Lane Centering Assist; 2. Adaptive Cruise Control;

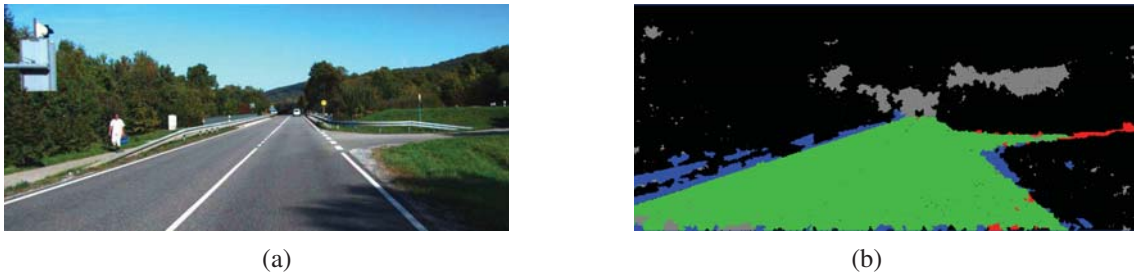


FIGURE II.22 – Détection de la surface de la route

La complexité de ce problème réside surtout dans la très grande fiabilité demandée aux systèmes ADAS. On exige qu'elles fonctionnent par toute condition météorologique et, de plus, on souhaite que le passage d'un mode automatique à un mode manuel se fasse suffisamment en avance de la situation critique, pour que le conducteur puisse comprendre la scène et se remettre en situation avant d'entamer une manœuvre.

Pour réaliser cela, il ne suffit pas de rouler sur la voie et de regarder la présence et l'absence d'un objet sur la route, il faut également donner une sémantique à cet objet. Pour ce faire, la communauté scientifique et les industriels travaillent sur la détection, la classification et l'étiquetage des objets (figures II.23(a) et (b)). A titre d'exemple il faudrait faire la distinction entre le piéton qui est sur la route, celui qui est sur le trottoir, mais à terme il faudrait détecter le piéton qui est sur le trottoir et a l'intention de commencer à traverser la route [Benenson et al., 2014; Zhang et al., 2016].

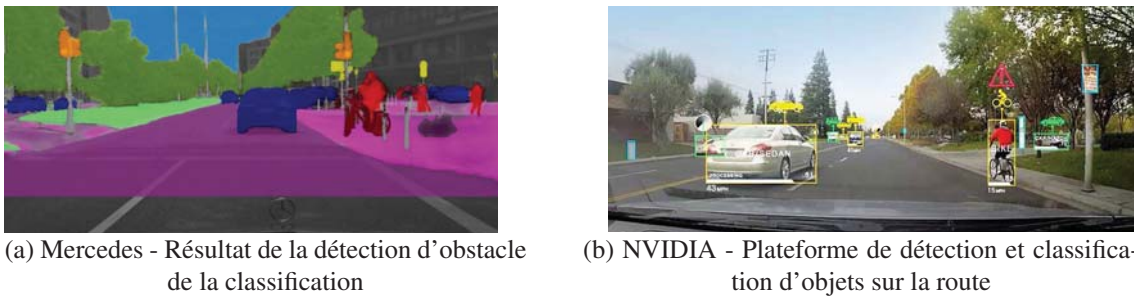


FIGURE II.23 – Détection et classification des objets/obstacles dans la scène

Pour pallier le problème de compréhension de la scène et de la vision limitée due à l'interprétation immédiate, la communauté robotique s'est inspirée du fonctionnement des humains. Ceux-ci font appel à des informations déjà apprises et auxquelles ils se réfèrent chaque fois que nécessaire. Une de ces pratiques est l'utilisation de la carte. Ainsi nous nous proposons de revenir par la suite en détail sur rôle de la cartographie dans notre problématique de voiturier autonome.

II.2.5 Détection d'obstacles pour une situation de parking

Dans le cas des manœuvres de parking, la diversité de l'environnement peut rendre la tâche de perception plus difficile. C'est pour cela que l'industrie automobile avance à petits pas dans ce monde de l'automatisation et propose des solutions de parking automatique utilisables dans des cadres restreints [Rhofs, 2007; Salvador, 2011].

En termes de perception de l'environnement, le choix du capteur influe énormément sur la qualité de détection. Si nous nous référons au tableau II.1, le constat est que même si le LiDAR

donne les meilleures informations en termes de qualité des distances aux obstacles, le prix le rend inutilisable pour ce type d'application de parking. En revanche, ce capteur a été utilisé par la majorité des équipes participantes à *DARPA Urban Challenge*. Cet événement est le plus important défi dans le domaine des systèmes de conduite autonome et a eu lieu le 3 novembre 2007, en Californie. Les équipes ont dû construire des véhicules autonomes capables de conduire dans le milieu urbain [Buehler et al., 2009]. Des critères précis d'évaluation ont été mis en place pour qualifier le degré de maturité des systèmes de conduite autonome dans des situations de circulation urbaine avec différents degrés de complexité. Les véhicules ont dû également se confronter aux situations de parking. Cet événement a été révolutionnaire car c'était la première fois que les véhicules autonomes ont interagi avec le trafic à la fois avec conducteur mais également sans conducteur, dans un environnement urbain. Les premières trois places ont été gagnées dans l'ordre par les équipes *Tartan Race*¹ [Urmson et al., 2007], *Stanford Racing*² [Montemerlo et al., 2008] et *Victor Tango*³ [Bacha et al., 2008] (voir figures II.24(a) à (c)).

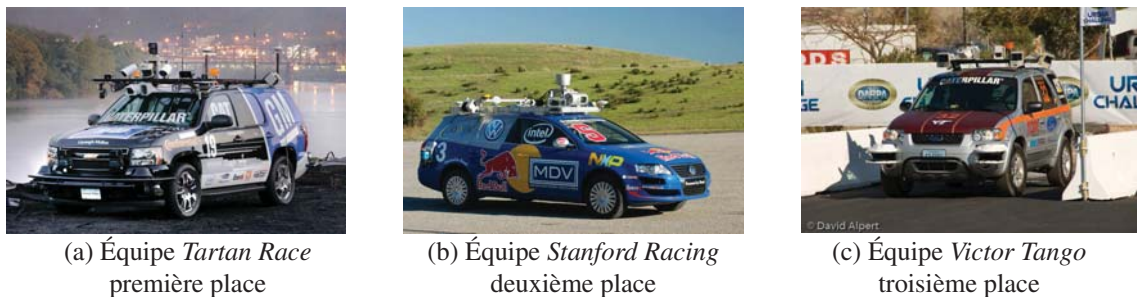


FIGURE II.24 – Les trois gagnants de DARPA Urban Challenge 2007

Les deux seuls capteurs de perception de l'environnement qui sont massivement déployés dans l'industrie automobile et que nous allons également utiliser sont le capteur ultrason et la caméra. Nous retrouvons une riche littérature scientifique sur deux problématiques liées : la détection d'obstacles et la cartographie de l'environnement utilisant la vision [Xu et al., 2000; Tseng et al., 2013; Hane et al., 2014; Seo, 2014; Hane et al., 2015].

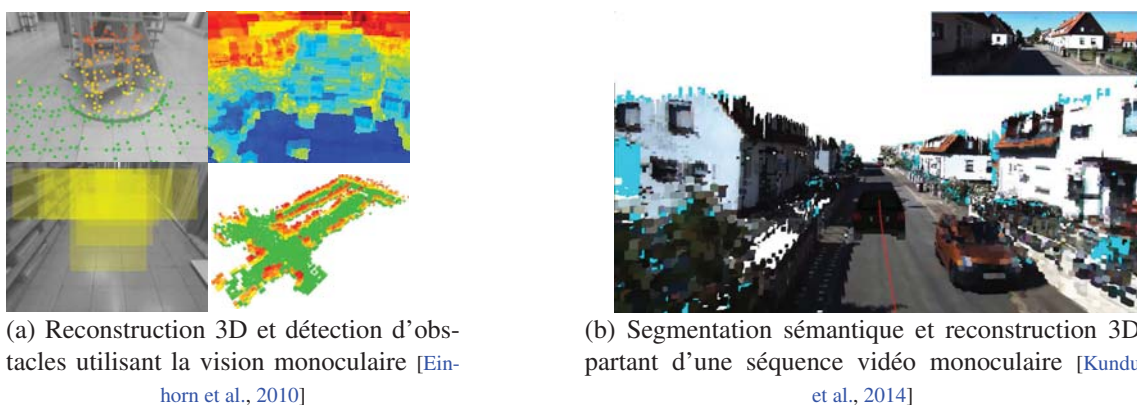


FIGURE II.25 – Exemple de travaux de recherche dans le domaine de reconstruction 3D de la scène et de la détection d'obstacles

De manière générale on retrouve trois grandes familles d'algorithmes de détection d'obstacles. La première technique suppose de mettre en correspondance des points d'intérêt dans plusieurs images successivement acquises et d'extraire la position 3D de chaque point d'intérêt [Weng

1. Carnegie Mellon University, Pennsylvania; 2. Stanford University, California; 3. Virginia Tech, Blacksburg, Virginia;

et al., 2013]. Plus il y a de points mis en correspondance, plus le nuage de points 3D est riche en information. La deuxième technique se propose d'utiliser de l'information de haut niveau. Ceci est possible grâce à des techniques d'extraction de formes et de mise en correspondance de celles-ci par des algorithmes de segmentation, classification ou de recherche dans des dictionnaires de données [Shi and Horvath, 2012; Dean et al., 2012; Le, 2013; Zhang and Zhou, 2014]. La troisième technique utilise un mélange des deux techniques [Lai et al., 2012, 2014; Kundu et al., 2014; Golodetz et al., 2015].



FIGURE II.26 – Interface R-Link de Renault Twingo - détection d'obstacle utilisant le système de capteurs ultrason et visualisation d'obstacles utilisant une caméra de recul

Comme nous l'avons déjà précisé, certains obstacles réglementaires peuvent ne pas être signalés. En analysant les figures II.27(a) à (f), on constate une large variété de matériaux pour le revêtement du sol : du gravier, du ciment, des dalles en béton ou en pierre, de l'herbe, etc. Dans une propriété privée il peut y avoir des zones où le bon sens dit de ne pas aller, la pelouse par exemple. En observant les figures II.27(a) et (b), on distingue facilement (en tant qu'humains) que, par la convention, on devrait circuler sur la partie bétonnée. Mais cette logique ne s'applique plus dans la figure II.27(c) car le sol est recouvert par ce que les industriels appellent *dalles à gazon* ou *stabilisateur de gazon et graviers* et qui permettent une fixation du sol tout en rétablissant une couverture végétale. On ne peut pas suivre une logique de couloir de circulation car on observe dans la figure II.27(d) un jeu de lignes au sol avec un rôle purement esthétique et des géométries difficilement interprétables.

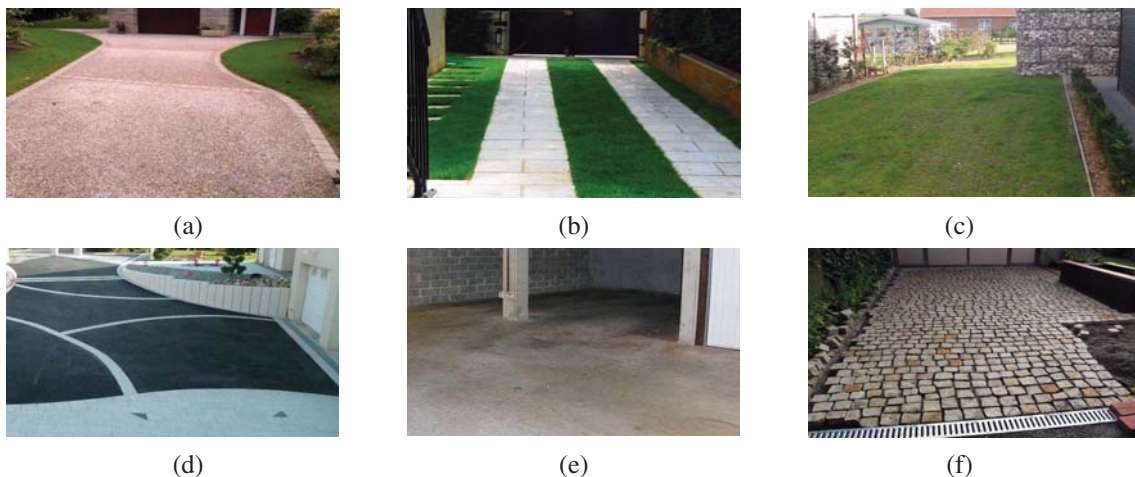


FIGURE II.27 – Variétés de revêtement du sol : gravier, goudron, ciment, herbe, dalles

Dans notre cas, de voiturier autonome, le chemin admissible peut être traité de plusieurs manières. Une première solution est de réaliser un système qui interprète l'environnement, mais nous avons vu que ceci est souvent très difficile. Une deuxième manière est de se fonder uniquement sur les obstacles physiques et, si nécessaire, demander au propriétaire de baliser les zones interdites

avec des objets physiques (typiquement des poteaux). La dernière solution est d'utiliser une carte préalablement construite qui contient des informations relatives à ces obstacles réglementaires. Dans ce dernier cas, étant donné que tous les obstacles sont connus et étiquetés, l'analyse de la zone traversable et la planification globale peuvent se réaliser à l'avance. Cette carte peut en pratique être créée par le véhicule lors d'un premier passage et enrichie à posteriori par l'utilisateur via une interface graphique. Le problème de planification globale se simplifie ainsi et ce qu'il reste à réaliser est une localisation précise du véhicule dans cette carte et de traiter la problématique de planification locale (ou trajectoire locale) pour l'évitement des obstacles non cartographiés. Nous verrons que ce sera l'approche que nous retiendrons.

Exploration de la zone navigable par le système de voiturier autonome

Comme nous l'avons déjà présenté, l'objectif de la thèse est de réaliser un système de voiturier autonome permettant au véhicule de se déplacer en toute sécurité entre la zone de prise ou de dépose et la place de parking [§I.6]. De plus on précise que la notion de *place de parking*, comme information de haut niveau ne fait pas l'objectif de nos recherches et que, pour le véhicule, la place de parking représente une position et une orientation dans laquelle il devrait se retrouver suite à la demande de son propriétaire. En cas de présence d'obstacle sur cette position, le véhicule choisira une position libre à proximité.

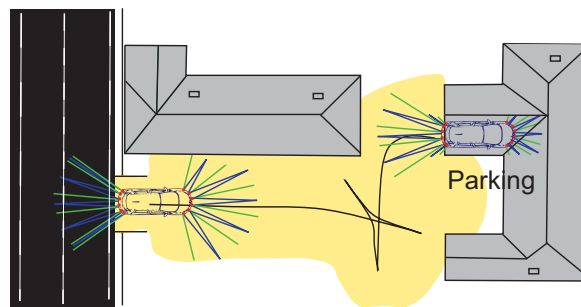


FIGURE II.28 – Schéma de principe : d'exploration de la zone navigable entre la zone de prise/dépose et la place de parking

De la même façon, le problème de détection d'un chemin admissible peut être traité à plusieurs niveaux. Rappelons-nous le problème des obstacles réglementaires [§II.1] : même si le véhicule peut physiquement convenir, les conventions font qu'il n'a pas le droit. Le type de diversité ne sera pas traité dans ce mémoire et notre système de voiturier autonome à la maison prendra en compte seulement les obstacles physiques. Dans la figure II.28, nous représentons schématiquement un environnement d'évolution du voiturier autonome. La couleur jaune indique l'espace navigable autour de la trajectoire d'apprentissage où le véhicule aura le droit d'évoluer. Nous allons définir par la suite quelle sera la largeur de ce couloir de navigation en fonction des capteurs utilisés et nous allons détailler les moyens de perception de l'environnement qui sont embarqués sur notre véhicule prototype.

II.3 Moyens de perception du véhicule prototype

Nous ne cherchons pas à détailler l'architecture du système et l'interface au véhicule prototype dans ce chapitre, mais uniquement de présenter les choix qui ont été faits en termes de moyens

de perception au vu de l'analyse précédente. Pour ce travail de thèse, comme présenté dans l'introduction, nous nous appuyons sur l'expérience de Renault, de l'Institut Pascal et d'IRSTEA. Chaque acteur a joué un rôle important dans la réalisation de ce projet mais nous allons tout d'abord nous attarder sur la présentation des moyens d'essais mis à disposition par Renault. Pour ce travail de thèse, Renault a mis à disposition un véhicule prototype sur une base mécanique de Zoé utilisée dans les projets de recherche et d'ingénierie avancée et pour le test des différents algorithmes d'automatisation du véhicule développés en interne. Au début du projet, en termes de contrôle du véhicule, le moteur, les freins et la direction étaient pilotables à travers des actionneurs de présérie ou spécifiques qui équipent le véhicule. En ce qui concerne la perception de l'environnement, initialement le véhicule était équipé d'un système de capteurs ultrason (SCU¹ ou USS² ou USA³) et une caméra arrière. Pour enrichir le champ de vision, une caméra grand angle a été ajoutée à l'avant.

Par la suite, nous concentrons notre exposé sur les moyens de perception de l'environnement que nous avons à disposition dans le but de satisfaire la fonction de détection d'obstacle pour la traversabilité.

II.3.1 Perception par capteurs ultrason

II.3.1.1 Mesure de distance par ultrason

L'industrie automobile propose sur tous les nouveaux véhicules (en standard ou en option) des capteurs ultrason pour la détection des obstacles lors des manœuvres de conduite à basse vitesse. Dans les sections qui suivent, nous allons approfondir le sujet relevant de la mesure par ultrason ainsi que les capteurs en rapport. Nous allons essayer d'exploiter au maximum les données issues de ces capteurs en vue d'assurer la sécurité du véhicule lors des manœuvres automatiques, mais nous verrons que nous chercherons aussi à les utiliser pour la localisation du véhicule.

Le son⁴ est une onde mécanique et élastique se propageant dans un milieu physique sous forme d'ondes longitudinales ou de compression. Ce phénomène est par exemple mis à profit par les hauts parleurs qui font vibrer une membrane qui à son tour fait vibrer l'air. Les ultrasons (voir figure II.29) sont des ondes sonores avec des fréquences plus élevées que la limite supérieure audible par l'oreille humaine. L'ultrason n'est pas différent du son (audible) dans ses propriétés physiques, sauf que les humains ne peuvent pas l'entendre. Cette limite audible varie d'une personne à l'autre mais elle est d'environ 20 kHz. Les dispositifs à ultrasons fonctionnent avec des fréquences de 20 kHz à plusieurs GHz. Le son/ultrason se propage d'autant plus vite que le milieu est dense, ce qui explique que le son soit plus rapide dans les liquides que dans l'air. C'est exactement cette propriété qui est utilisée dans l'échographie.

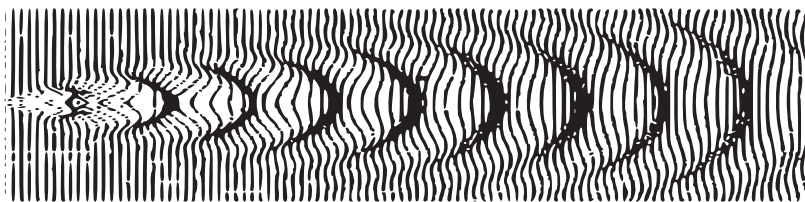


FIGURE II.29 – Exemple de forme d'onde ultrasonore

1. Système de Capteurs Ultrason; 2. UltraSound System ou UltraSound Scan; 3. Ultrasound Assistance; 4. «Sensation auditive engendrée par une onde acoustique» [Larousse];

On distingue trois types d'ultrasons selon la gamme de fréquence : les *ultrasons de puissance*, les *ultrasons de diagnostic* et les ultrasons de très haute fréquences¹. La première catégorie d'ultrasons (de 20 kHz à 1 MHz) peut entraîner une modification mécanique, thermique ou chimique dans le milieu dans lequel ils se propagent [Chemat, 2011]. Quant à la deuxième catégorie, les ultrasons sont des plus hautes fréquences (de 1 MHz à 10 MHz) et sont utilisés pour déterminer les caractéristiques physico-chimiques du milieu qu'ils traversent ou sont utilisés en télémétrie [Sato et al., 2014], en écholocalisation [Mann et al., 1998]), en contrôle non destructif ou en échographie [Bishko et al., 2008]. Nous nous intéressons uniquement aux applications télémétriques des capteurs ultrason. La troisième catégorie d'ultrason, de très haute fréquence (supérieures à 10 MHz), est pour l'instant utilisée à très petite échelle dans des applications de microscopie acoustique (SAM²) [Störkel et al., 1998; Brand et al., 2008] ou pour des applications de caractérisation des matériaux [Bryner, 2009].

En ce qui concerne la composition technique des capteurs ultrason, on trouve deux parties essentielles : l'émetteur et le récepteur. L'émetteur transmet un son à une fréquence définie et le récepteur collecte une partie du son répercuté par les obstacles. La distance aux objets est calculée par rapport au temps que le son met pour revenir au récepteur [Sarabia et al., 2013]. Il existe des configurations appelées des *antennes de capteurs ultrason* (figure II.30) dans lesquelles on aligne un émetteur et deux ou plusieurs récepteurs ou plusieurs émetteurs et plusieurs récepteurs [Bazulin, 2014]. Ces antennes des capteurs sont utilisées pour augmenter la précision de réception ou pour réaliser de la reconnaissance de forme grâce au déphasage temporel des différents échos. Très souvent, les capteurs ultrason sont montés sur des parties rotatives pour permettre l'acquisition des différentes mesures depuis plusieurs positions.

Les figures II.30(a) à (c) présentent trois cas de mesure de la distance. Le premier cas est le cas idéal : l'obstacle est situé bien en face et perpendiculaire au capteur. Le deuxième cas génère bonne mesure également, mais seul l'obstacle le plus proche sera détecté. Dans le dernier cas, la mesure sera imprécise puisque l'angle d'incidence sera relativement petit.

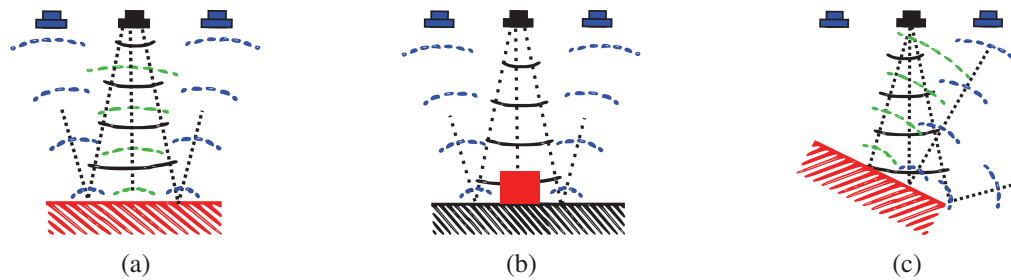


FIGURE II.30 – Cas favorables à la bonne détection d'obstacles par capteur ultrason. La couleur noir représente le capteur avec la partie émetteur et récepteur d'ondes, en bleu, le capteur avec uniquement le rôle de récepteur

Il est essentiel de bien connaître la structure du faisceau pour le capteur que l'on utilise si l'on souhaite réaliser une carte, ce qui est moins le cas pour l'évitement d'obstacle. Il faut également tenir compte du fait qu'à très courte distance, les capteurs ultrason sont aveugles. Ceci est dû à la temporisation entre l'émission de l'onde sonore et de début de la détection de l'onde réfléchi qui est nécessaire pour ne pas perturber cette mesure. Ceci est nécessaire car dans l'industrie automobile c'est le même composant qui émet et réceptionne les ondes, alors que d'autres technologies ont un émetteur et un récepteur séparés. Pour les SCUs³ utilisés sur les véhicules, cette zone aveugle est entre 0,15 m et 0,25 m, diminuant jusqu'à 0,10 m pour SCU de présérie. Pour pallier le problème, les SCUs utilisent également l'information odométrique du véhicule. En

1. *ultrahigh frequency*, en anglais; 2. *Scanning Acoustic Microscopy*; 3. *Système de Capteurs Ultrason*

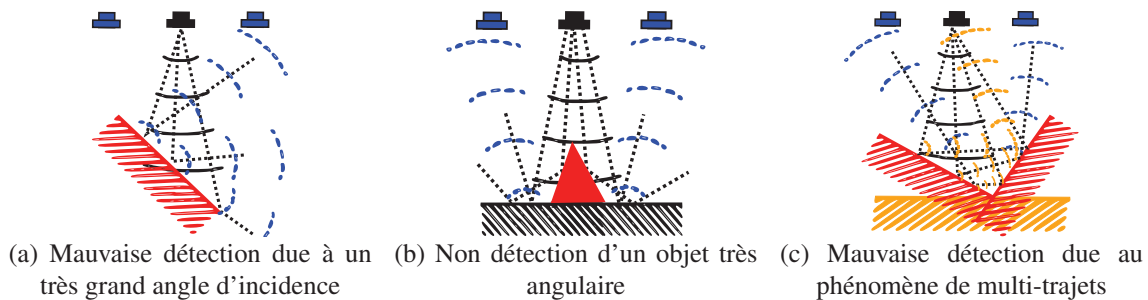


FIGURE II.31 – Influence de la forme et la position de l'obstacle dans qualité de détection de distance par capteurs ultrason

faisant l'hypothèse que l'obstacle le plus proche vers lequel le véhicule se dirige est statique, le SCU sera capable d'estimer la distance restante jusqu'à la collision et, en conséquence, d'alerter le conducteur. Cette approche a une limite facilement atteignable dans les grandes agglomérations urbaines où les distances entre les véhicules garés sont trop petites. Si au démarrage du véhicule, un obstacle est situé dans la zone aveugle ou si le véhicule est déjà en collision, le SCU ne pourra pas le savoir. C'est pour cela que les constructeurs s'intéressent à intégrer dans les nouvelles générations de SCU, une détection par contact physique permettant ainsi d'avoir une information précise sur l'état de collision ou de non collision du véhicule.

II.3.1.2 Limitations et facteurs de perturbation des capteurs ultrason

- **la forme et la position relative de l'obstacle** - jouent un rôle essentiel dans la détection car elles peuvent amener le système à ne pas interpréter correctement son environnement (figures II.30 et II.31);
- **la texture de l'obstacle/du sol** - un mur recouvert de moquette répercutera moins bien l'onde qu'un mur recouvert de peinture; certains vêtements absorbent plus les ondes et rendent les piétons difficilement détectables; le sol recouvert de gravier ou de bitume très rugueux augmente les réflexions parasites et entraîne un taux élevé de fausses détection;
- **le cross-talk** - deux capteurs ultrason ne peuvent pas être utilisés simultanément côte à côte car s'ils ont la même fréquence, il n'est pas possible de distinguer lequel des deux a émis les ondes; la solution est d'activer les capteurs les uns après les autres, diminuant ainsi le taux de rafraîchissement global;
- **les échos parasites** (figure II.34(b));
- **la zone aveugle** - pour des distances inférieures à 0,20 m, dans notre cas, les capteurs ne détectent pas d'obstacles;
- **le temps de latence de détection**;
- **l'obstruction** - un capteur obstrué (par de la boue ou de la neige) n'est pas capable de se rendre compte qu'il se trouve dans une situation de défaillance fonctionnelle;
- **le perceptual aliasing** - ce problème désignant l'incapacité du capteur à distinguer de manière unique tous les lieux d'un environnement; à titre d'exemple si le capteur ultrason permet à un robot de se situer latéralement le long d'un mur, il ne permet pas de savoir la plupart du temps où il se situe le long de ce mur.

II.3.1.3 Système de 12 capteurs ultrason utilisé sur le véhicule prototype

Le véhicule prototype dont nous disposons est équipé d'un ensemble de 12 capteurs, 6 placés dans le bouclier avant et 6 dans le bouclier arrière du véhicule. La figure II.32(a) donne une vue d'un capteur ultrason et de son connecteur. Chaque capteur a 3 fils : un fil de masse, un fil d'alimentation (plage d'alimentation de 9 V à 16 V) et un fil, que l'on appelle fil de signal, qui donne une information numérique (modulé en tension de 0 V à 5 V) en corrélation avec la distance observée jusqu'à l'obstacle.

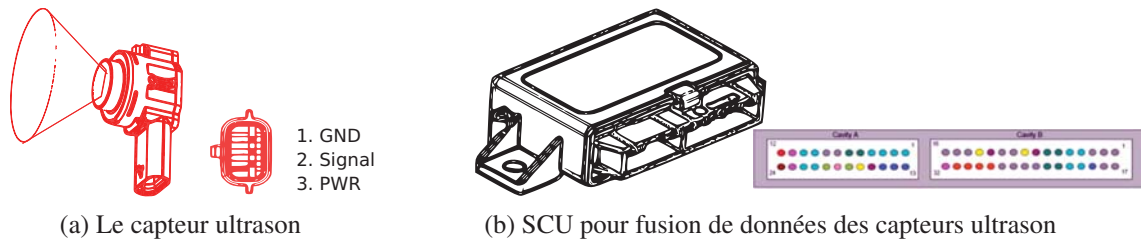


FIGURE II.32 – Le système des capteurs ultrason

Pour calculer une distance, des ondes ultrason sont émises et lors de la réception du premier écho, en connaissant la vitesse du son et le temps écoulé entre l'émission et la réception, on détermine la distance à l'obstacle. Dans la génération actuelle des SCU, seul un capteur ultrason peut émettre à la fois pour éviter le cross-talk. En revanche, les échos peuvent être reçus également par le capteur voisin (figure II.31(a)). Le fait de recevoir les mêmes échos par deux ou plusieurs capteurs permet d'améliorer la précision de calcul. L'utilisation d'algorithmes de triangulation permet la réduction de l'incertitude angulaire. La génération future des SCU, en développement aujourd'hui, permettra d'éviter le phénomène de cross-talk en utilisant des fréquences et des modulations différentes des ondes émises.

Tous les signaux de mesure sont utilisés par une unité centrale (figure II.32(b)) qui est capable de réaliser une estimation de la distance aux obstacles situés à moins de 4,095 m (information écrite sur 12 bits) dans le repère des capteurs. Le SCU utilise, en plus des échos, la distance parcourue par le véhicule et réalise une meilleure estimation de la position lors du déplacement.

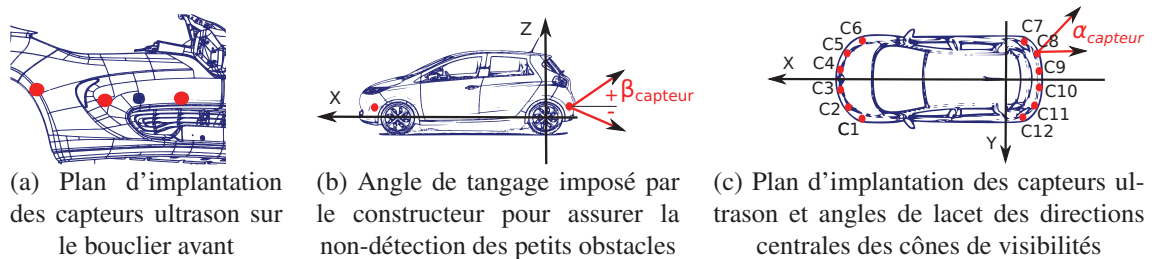


FIGURE II.33 – Informations relatives à l'emplacement des capteurs ultrason

Capteurs	C ₁ /C ₆	C ₂ /C ₅	C ₃ /C ₄	C ₇ /C ₁₂	C ₈ /C ₁₁	C ₉ /C ₁₀
α_{capteur}	$\pm 77,5^\circ$	$\pm 40,8^\circ$	$\pm 12,8^\circ$	$\pm 40,0^\circ$	$\pm 36,5^\circ$	$\pm 6,8^\circ$
β_{capteur}	$+0,3^\circ$	$+3,2^\circ$	$+3,0^\circ$	$-0,2^\circ$	$-2,2^\circ$	$-2,1^\circ$

TABLE II.2 – Angles d'orientation des capteurs ultrason, lacet (α_{capteur}) et tangage (β_{capteur}) imposés par le fournisseur

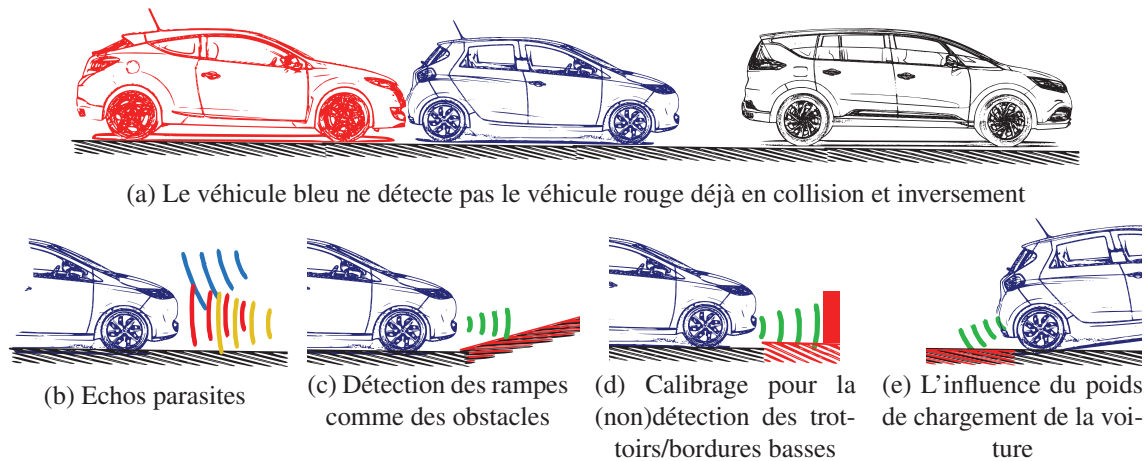


FIGURE II.34 – Cas de fausses détections d'obstacles par le SCU

Les emplacements des capteurs sur les boucliers ainsi que les angles de lacet (α_{capteur}) et de tangage (β_{capteur}) ont été décidés conjointement entre les équipes de Renault et l'entreprise qui a développé et fabriqué le SCU, après une phase d'analyse de l'impact des positions des capteurs sur le résultat attendu. Nous remarquerons dans le tableau II.2 que les valeurs requises en particulier sur l'angle de tangage sont très précises. Un mauvais calibrage de l'angle de tangage a un impact important sur la détection ou non-détection des petits obstacles. En plus il existe des ondes échos qui rebondissent du sol et qui risquent d'être interprétées comme des obstacles (figure II.34(e)). Il est nécessaire de connaître ces angles et leurs tolérances pour pouvoir maîtriser la zone de non détection des obstacles très bas, comme certaines bordures basses ou les ralentisseurs. Les figures II.34(b) à (e) illustrent ces cas de fausses détections.

II.3.1.4 Caractéristiques du SCU

Avant d'entrer dans les détails du SCU, nous précisons que la connexion au bus CAN¹ du véhicule se réalise à travers un adaptateur PCAN-USB (figure II.35) produit par Peak System [PEAK-System, 2014]. A travers cette connexion, nous sommes capables de récupérer tous les flux d'information passant sur le bus CAN, dont l'information issue du calculateur SCU.

FIGURE II.35 – Adaptateur *PEAK PCAN-USB Pro*

Le SCU est conçu par un fabricant expert du domaine et adapté aux exigences imposées par Renault. Ainsi le système est fourni sous la forme d'une boîte noire (figure II.37) dont on connaît uniquement le principe de fonctionnement mais pas les détails des algorithmes. L'unité centrale communique avec le véhicule à travers le réseau CAN. Utilisant les 12 signaux issus des capteurs ultrason, l'unité centrale du SCU réalise une fusion entre les données des capteurs ultrason et l'information odométrique, issue du système ESP, et l'information des angles des roues avant, issue du capteur d'angle volant installé sur la colonne de direction.

1. Controller Area Network ;

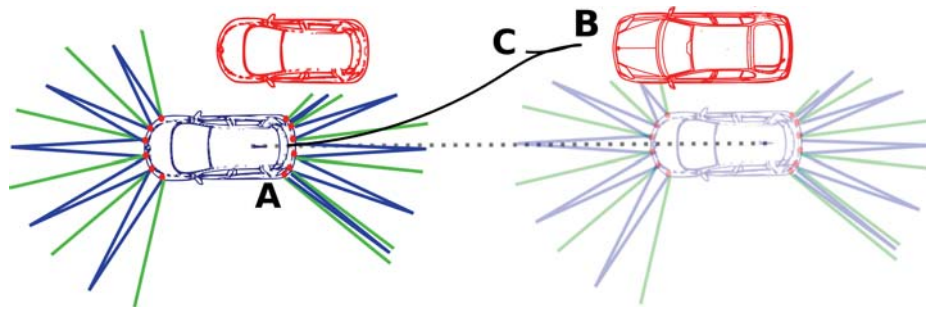


FIGURE II.36 – Trajectoire lors d'une manœuvre standard de HFP latéral

Le SCU envoie des informations qui permettent d'afficher sur l'interface la zone où se trouvent les obstacles autour du véhicule et une distance approximative exprimée par des blocs de couleurs différents (voir figure II.38(a)). Ce système était conçu également comme partie intégrante du système de parking semi-automatique HFP¹. Le type de manœuvre de parking visé utilisant ce moyen de perception est le *parking latéral en créneau* (figure II.36), *en épi* et *en bataille*, donc des cas d'usage bien précis. Pour réaliser cela, le conducteur actionne un bouton de l'IHM² pour signaler au véhicule le début de la recherche d'une place de parking puis avance le long des rangées de véhicules. Dès qu'un emplacement vide est reconnu, il sera proposé au conducteur pour validation avant le début de la manœuvre de parking. Il faut bien souligner que, pour le véhicule, l'espace trouvé n'a pas de signification de haut niveau, donc le véhicule ne sait pas si l'emplacement vide est une place de parking ou non, cette tâche restant toujours à réaliser par le conducteur, d'où la nécessité d'une validation.

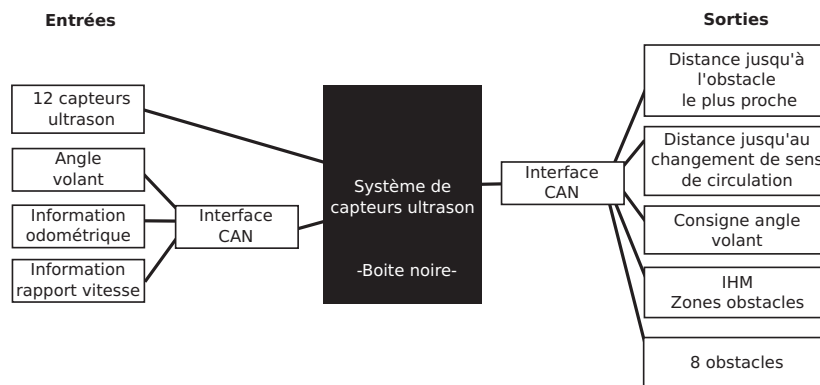


FIGURE II.37 – Fonctionnement en boîte noire du système HFP

Ce que le SCU renvoie consiste en 8 paires de points placées dans le repère véhicule et qui forment des segments de longueurs égales aux largeurs apparentes des obstacles observés. La limite d'envoi de 8 paires de points est liée à des contraintes d'occupation du réseau CAN. Par conséquent, le système choisit les obstacles les plus proches. Le SCU fournit également pour chaque paire de points, la distance de collision entre les objets et la carrosserie du véhicule (figure II.39(a)). Une longueur du segment nulle signifie que le système n'a pas pu estimer la largeur de l'objet mais uniquement sa position approximative.

L'inconvénient de ce système vient du fait qu'il fournit l'accès uniquement aux 8 plus proches objets et que très souvent la largeur de ces obstacles n'est pas estimée. A titre d'exemple, si un piéton se situe devant un objet large, nous recevons uniquement la position du piéton car la

1. Hand Free Parking; 2. Interface Homme-Machine;

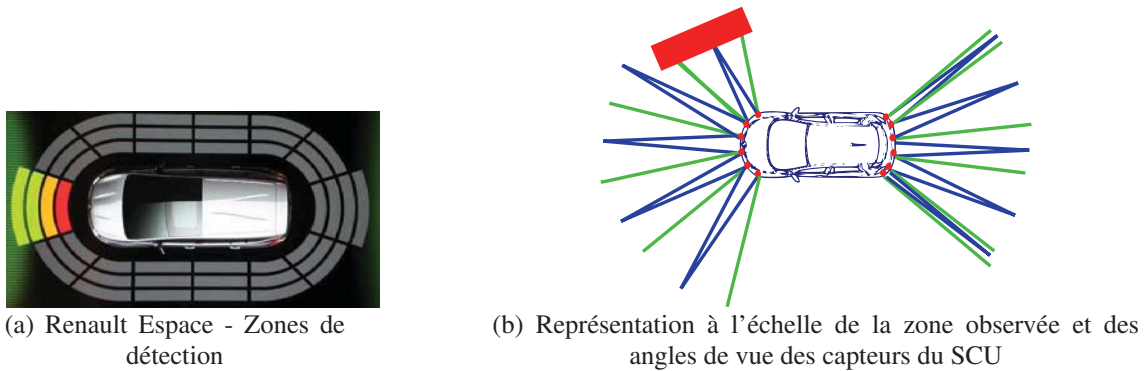


FIGURE II.38 – Dans la figure (a) on retrouve une illustration de l'interface graphique avec les zones de détection d'obstacles colorées différemment en fonction du danger de collision. Dans la figure (b) on retrouve les zones effectives observées par le SCU

collision du véhicule se fera évidemment avec l'objet le plus proche. Pour ces raisons, nous n'avons pas pu réaliser une cartographie complète de l'environnement utilisant ces données post-traitées.

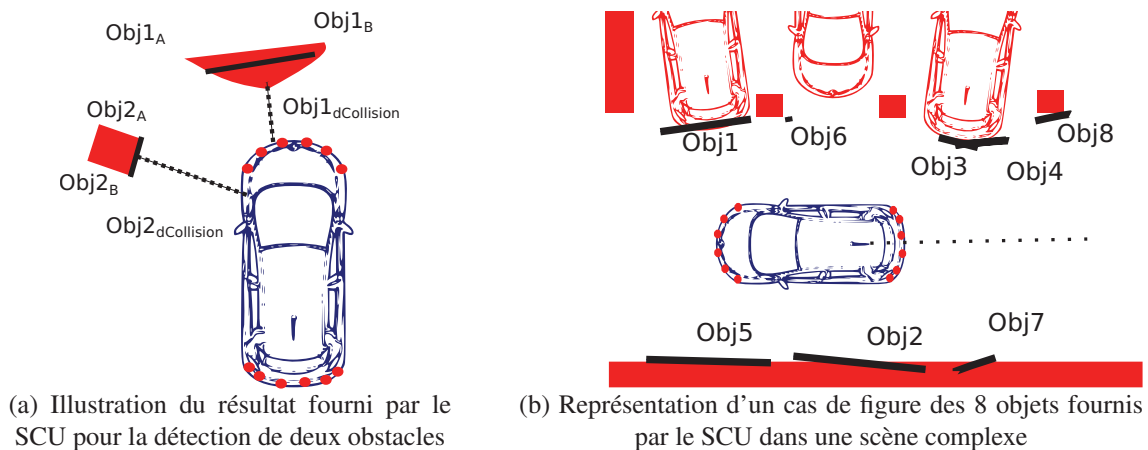


FIGURE II.39 – Illustration des résultats produits par le SCU dans deux situations de différents niveaux de complexité de la scène

De manière uniquement expérimentale, le constructeur a accepté de nous mettre à disposition des mesures de distances brutes calculées par leur système, donc des données toujours de haut niveau. Ces mesures seront nommées distance calculées à partir des *échos directs*, d_{EchosD} , (en vert dans la figure II.38(b)) et distances calculées utilisant les *échos croisés*, d_{EchosC} , (en bleu dans la figure II.38(b)). Les échos directs, sont les échos reçus par le capteur C_i qui les a envoyés. Si les échos sont reçus par un autre capteur (C_{i-1} ou C_{i+1}), ils seront appelés échos croisés. Nous ne connaissons pas la fréquence d'acquisition des données brutes capteur car c'est une donnée non partagée par le constructeur, en revanche nous savons que le SCU nous envoie sur le bus CAN du véhicule ces distances à une fréquence de 10 Hz. Ces données ne sont pas datées, ce qui a un impact significatif avec l'augmentation de la vitesse.

Le diagramme d'émission est une caractéristique importante pour chaque capteur ultrason et relie l'angle de vue du capteur à l'énergie rayonnée par l'émetteur. Dans notre cas, cette information est confidentielle et elle n'est pas partagée par le constructeur, donc il a fallu réaliser une procédure d'identification des capacités du capteur par mesures expérimentales. Cela consiste à déterminer si le capteur est capable ou pas de détecter et sous combien de temps, certains objets

de tailles connues situés dans son champ de vision.

La performance des capteurs ultrason est mesurée sur la plage de distance de 0 m à 4,095 m suivant les critères de précision, fidélité et reproductibilité de la mesure. Les obstacles testés ont différentes formes et dimensions, sont statiques ou se déplacent à différentes vitesses ou qui surgissent dans le cône de perception. De plus, des tests sont réalisés avec d'objets de différents matériaux pour identifier les limites en termes d'absorption des ondes pour différents matériaux (les cas critiques ayant été détectés pour certains matériaux textiles type mousse). Ces résultats seront utilisés par la suite dans la création de carte d'occupation.

II.3.2 Perception par caméra

II.3.2.1 Généralités

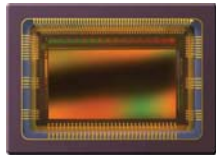
Même si aujourd'hui il semble naturel d'utiliser des photos numériques, leur démocratisation s'est réalisée uniquement après les années 2000. Auparavant, les gens utilisaient la *photographie¹ argentique²*, technique photographique à base d'un processus photochimique qui utilise une pellicule sensible à l'exposition à la lumière. Une fois l'exposition réalisée, la pellicule subit un processus chimique nommé couramment *développement* qui permet de rendre la pellicule immune à une nouvelle exposition à la lumière et, suite à cela, un tirage de la photo sur papier est possible. Toujours depuis les années 2000, le terme *argentique* est devenu de plus en plus utilisé comme un besoin de distinguer la photographie *chimique* de celle dite *numérique*. L'élément à la base des caméras numériques qui permet la perception de l'environnement est le *capteur photographique*.

Capteurs CCD et CMOS

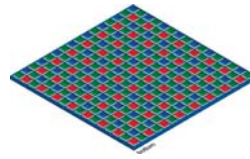
Le capteur d'image est le composant de base des caméras numériques et sur le marché deux types de familles de capteurs sont commercialisés : les CCD³ et les CMOS⁴. Le CCD a été inventé par George Smith et Willard Boyle chez *Bell Telephone Laboratories* en 1969 [Boyle, 1961; Boyle and Smith, 1971, 1974]. L'invention du CCD est pleinement reconnue en 2009 quand les deux inventeurs recevront la moitié⁵ du Prix Nobel de physique. Ce capteur contient un réseau de condensateurs qui rassemble des charges proportionnelles à la quantité de lumière qui atteint chaque cellule photosensible. La quantité de charge dans chaque condensateur est ensuite convertie en valeur numérique et leur cumul produira une image numérique.

En ce qui concerne l'appellation CMOS, celle-ci ne désigne pas uniquement un capteur mais toute une technologie pour la construction de circuits intégrés qui partagent une certaine conception. La technologie CMOS a été brevetée en 1963 par Frank Wanlass dans l'entreprise Fairchild Semiconductor [Wanlass, 1967] et aujourd'hui elle est très généralisée. Le capteur d'image CMOS est juste un type de CMOS-Integrated Circuit qui a été construit spécifiquement pour capturer d'images. Il mesure l'intensité de la lumière en utilisant un tableau de photodétecteurs qui sont couplés à des amplificateurs pour obtenir une quantité suffisante de charge.

1. *Photo* : du mot grec *photos* qui signifie lumière ou clarté; *graphie* : du mot grec *graphein* qui signifie peindre ou dessiner;
2. a) Procédé permettant d'enregistrer, à l'aide de la lumière et de produits chimiques, l'image d'un objet. b) Ensemble des techniques d'enregistrement de rayonnements électromagnétiques par des procédés photochimiques [Larousse]; 3. Charge-Coupled Device; 4. Complementary Metal-Oxide-Semiconductor; 5. La deuxième moitié du prix étant gagnée par Charles Kao pour des avancées dans les communications par fibre optique;



(a) Capteur d'image CMOSIS CMV2000



(b) Matrice de couleurs du capteur d'image

FIGURE II.40 – Le capteur en couleurs intégré dans la caméra *Baumer VisiLine VLG-22M/C* utilise des filtres de couleurs placés dans un motif en damier (appelé filtre de Bayer), généralement dans les proportions suivantes : 50% vert, 25% rouge et 25% bleu. Pour un capteur monochrome, un seul filtre est utilisé et la valeur du pixel correspond à une seule couleur

Les puces CMOS peuvent être fabriquées sur la majorité des lignes de production de silicium standard, de sorte qu'elles ont tendance à être très bon marché par rapport aux capteurs CCD. De plus, les capteurs CMOS consomment traditionnellement peu de puissance, allant jusqu'à 100 fois moins d'énergie par rapport aux capteurs CCD équivalents. C'est pour cela que la plupart des caméras de téléphones mobiles et d'ordinateurs portables intègrent ce type de capteur. Mais les avantages ne sont pas seulement côté capteurs CMOS, car les capteurs CCD sont plus sensibles et permettent d'obtenir des images d'une meilleure qualité surtout dans des conditions de faible éclairage. Les capteurs CCD ont été produits durant une plus longue période de temps, de sorte qu'ils sont plus matures. Pour cela, toutes les caméras DSLR¹ utilisent des capteurs CCD. On ne peut pas faire du rabais en ce qui concerne la qualité, surtout dans des métiers comme celui de photographe professionnel. Les capteurs CCD sont un facteur contributeur dans le prix d'une caméra DSLR, étant donné que ceux avec des résolutions² très élevées sont beaucoup plus chers.

Technologie	Inconvénients	Avantages
CCD	- Pas de possibilité de fenêtrage (ROI ³) - Saturation en cas de fort éclairage (effet d'éblouissement)	+ Images de haute qualité + Forte sensibilité à la lumière + Faible bruit + Technologie plus mature
CMOS	- Moins sensible à la lumière - Images plus bruitées	+ Possibilité de définir une ROI + Plus compact + Faible consommation + Moins cher

TABLE II.3 – Avantages et inconvénients des technologies CCD et CMOS de capteurs d'image

Chacun de ces capteurs a ses avantages et inconvénients (tableau II.3) de telle sorte que chacun des deux convient parfaitement à une niche spécifique. Les CCD sont excellents pour prendre des images de très haute qualité pour les photographes professionnels. Les caméras CMOS sont moins chères et consomment moins d'énergie ce que les rendent parfait pour l'intégration dans des appareils portables comme les téléphones mobiles, ordinateurs portables, les consoles de jeux, mais également pour notre véhicule. La différence entre les deux types de capteur diminue avec les avancées technologiques.

Sensibilité à la lumière

La sensibilité⁴ à la lumière, ou l'éclairage minimal, se réfère à la plus petite quantité de lumière dont le capteur de la caméra a besoin pour produire une image de qualité utilisable. Les

1. Digital Single-Lens Reflex; 2. Nombre de pixels du capteur; 3. Region of Interest; 4. Variation électrique pour un éclairage donné;

capteurs sont sensibles uniquement sur une gamme ¹ de lumière et cette caractéristique se mesure par la réponse (ou sensibilité) spectrale ² (figure II.41(c)).

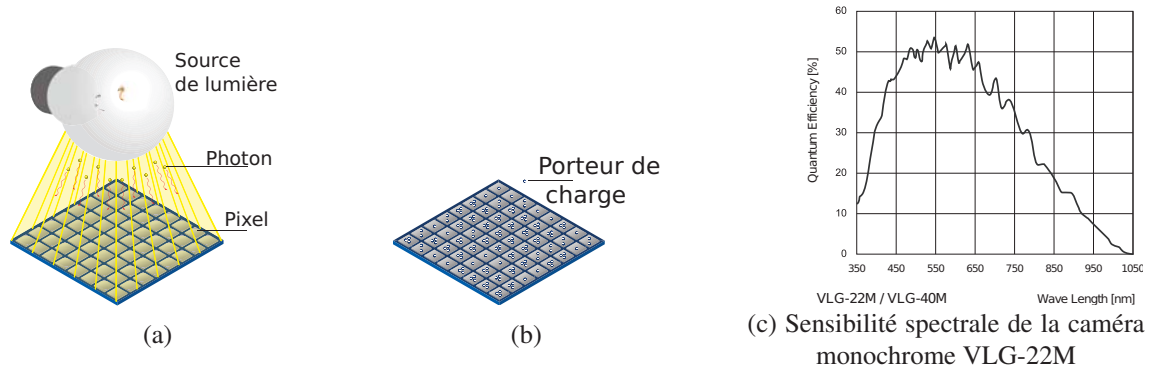


FIGURE II.41 – L’incidence de la lumière sur la surface de contact induit la séparation de la charge sur le semi-conducteur du capteur. Le résultat est une différence de potentiel qui est ensuite utilisée pour extraire le signal, donc l’information. Le temps d’exposition influe sur la quantité de charge accumulée et donc sur la puissance du signal

La mesure de l’éclairement lumineux se fait en lux et, à moins que le capteur ne soit sur-exposé, l’image sera meilleure dans une scène mieux éclairée. La quantité de lumière qui est nécessaire pour produire une image de bonne qualité dépend de la caméra, de la taille, de la qualité et de la sensibilité à la lumière de son capteur, de la durée d’exposition, le rapport focal, de la température de couleur. Dans les situations de faible ou fort éclairage, une caractéristique importante à rechercher pour une caméra est la capacité de collecte ou de limitation de la lumière dans la lentille. Cela peut être réglé par l’ouverture de l’objectif qui définit la quantité de lumière pouvant passer à travers une lentille. Ce calcul du *rapport focal* se fait en rapportant la distance focale de l’objectif au diamètre de l’ouverture ou de l’iris.

En fonction de la technologie utilisée et de la puissance de calcul, l’acquisition du capteur se réalise suivant plusieurs techniques d’échantillonnage nommées techniques de balayage d’images. Dans les figures II.42(a) à (d) nous présentons deux grandes classes de balayage : balayage instantané (*global shutter*) et balayage progressif (avec des version *rolling shutter*, *codding rolling shutter* et *interlaced shutter*).

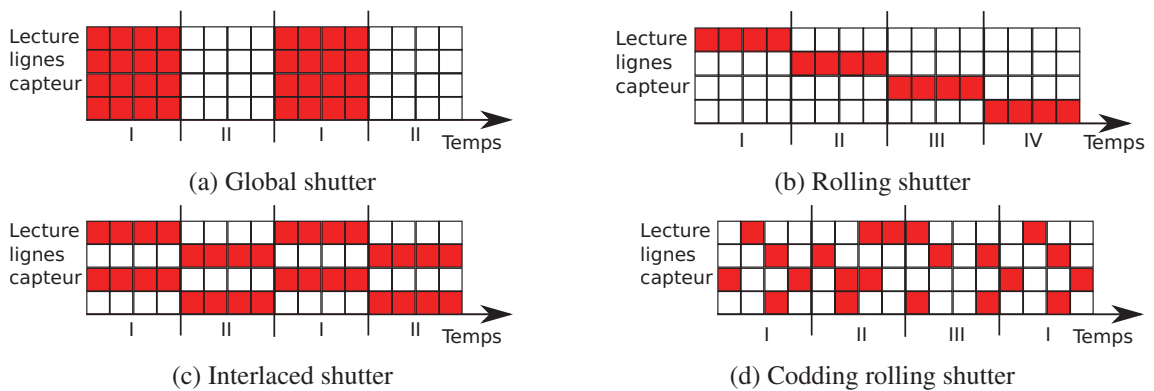


FIGURE II.42 – Techniques de balayage de l’image (d’acquisition de l’information capteur)

1. Les capteurs sont sensibles dans le spectre visible mais également dans le proche infrarouge (IR) et ultraviolet (UV); 2. Région du spectre de lumière dans laquelle le capteur est sensible;

La technique d'acquisition de la lumière sur toutes les photocellules du capteur au même temps s'appelle *global shutter* (figure II.42(a)). Mais cette technique est la plus complexe à mettre en place car elle demande une puissance de calcul grande pour traiter en parallèle tous les signaux électriques. La caméra utilisée dans notre projet de parking autonome est équipée d'un global shutter. Le choix est volontaire car nous voulons éviter le décalage temporel des pixels dans une même image qui complexifie énormément les algorithmes d'analyse de scène ou de localisation. Un paramètre important à prendre en compte est le nombre de bits alloués pour la description de l'information liée à un pixel. L'opération qui réalise cette spécification s'appelle *qualification*, elle consiste à spécifier le nombre de bits utilisés dans la représentation numérique du signal acquis.

Comme nous l'avons déjà précisé, la qualité des images est importante et un critère est la résolution. Nous utilisons couramment le terme *résolution* mais, en réalité il y a deux termes dont on doit tenir compte : la *définition* et la *résolution*. La *définition* est le nombre total de pixels dans l'image et se mesure en pixels. Par exemple une image de $2\,000\text{px} \times 3\,000\text{px}$ peut être également une image de 6 Mpixels et elle aura toujours cette définition quelle que soit la taille de l'image imprimée qu'on lui donne ensuite. La définition n'indique pas la qualité esthétique d'une photo (flou, effet de compression etc.), mais uniquement ses dimensions. La résolution, en revanche, est le lien entre la taille en pixels et la taille dans le système métrique. Elle se caractérise par le nombre de pixels par unité de longueur, donc elle représente la densité de pixels de l'image.

L'objectif

Un objectif (figure II.43(a)) sur une caméra remplit principalement trois fonctions. D'abord, il définit le champ de vision, c'est-à-dire qu'il permet de choisir à quel niveau de détail une partie de la scène devra être capturée. Le champ de vision (figure II.43(b)) est déterminé par le ratio entre la distance focale de l'objectif et la taille du capteur d'images. En fonction de la distance focale on retrouve des objectifs dits standards, à grands angle ou fish-eye, des objectifs à petits angles et des macro-objectifs. La capacité d'adaptation de la distance focale classe les objectifs en objectifs fixes et objectifs à foyer progressif.

Focale [mm]	2,7	14	24	28	35	50	70	80	100	135	200
Angle du champ [deg]	185	114	84	75	63	47	34	30	24	18	12
Type d'objectif	Fish-eye		Grand angle		Normal			Téléobjectif			

TABLE II.4 – Correspondance entre la distance focale [mm] et l'angle de champ diagonal [°]



FIGURE II.43 – Types d'objectifs de caméras photo et champs de perception associés

La deuxième fonction de l'objectif est le réglage de la quantité de lumière passant à travers de lui et qui arrive sur le capteur d'image. Cela se réalise en utilisant un élément mécanique

interposé sur le trajet lumineux nommé diaphragme [Becherrawy, 2005]. De cette manière, le problème de sous-exposition et surexposition est réglé. La capacité à contrôler l'ouverture de l'iris (figure II.44(a)) d'une caméra joue un rôle important dans la qualité de l'image car il est utilisé pour maintenir le niveau de lumière optimal pour le capteur d'image afin que les images soient nettes, claires et correctement exposées avec un bon contraste. Le diaphragme peut être réglé manuellement ou automatiquement et dans ce cas il est appelé diaphragme à iris automatique.

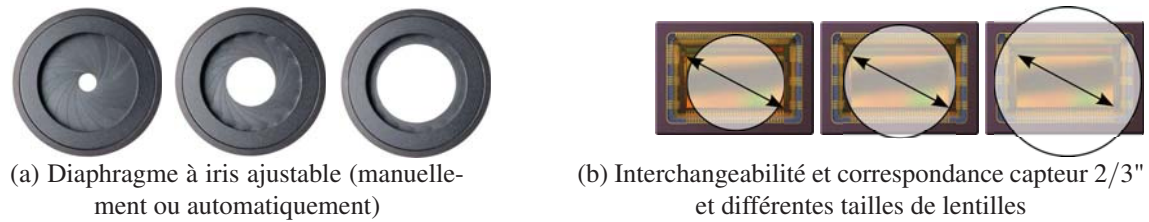


FIGURE II.44

Les systèmes d'aide à la conduite qui emploient des caméras et sont destinés à être utilisés sur autoroute ou route nationale, ont besoin d'un champ de vue assez étroit¹ (40° – 60°) et utilisent des objectifs avec des focales supérieures à 50 mm à une très faible distorsion de l'image. Par contre, pour des applications de parking ou pour la conduite en ville où l'objectif est de regarder une zone plus ample, les ADAS sont basés sur des caméras grands angles ou *fisheyes*.



FIGURE II.45 – Illustration de la continuité de recouvrement de la scène dans des images successives même pour une grande vitesse de rotation de la caméra dans le virage

Pour un champ de vue élargi, nous avons choisi également un objectif avec une focale de 2,7 mm qui donne un champ de vue de 185° . Ceci nous permet d'avoir un aperçu d'objets situés dans la proximité du véhicule. De plus, comme la plupart des algorithmes de traitement d'images destinés à la détection d'obstacles ou à la localisation ont besoin d'analyser des suites d'images pour tirer de l'information, un large champ de vision permet d'avoir un recouvrement important entre deux images successives, y compris pour des mouvements de rotation très importants (figures II.45). Ceci permet de suivre facilement les points d'intérêts d'une image à l'autre à des fréquences moins élevées et d'éviter les problématiques dites de discontinuité d'appariement.

Nous nous intéressons maintenant aux fonctionnalités de la caméra ou autrement dit aux fonctions de prétraitement.

1. Une meilleure maîtrise de la déformation de l'image permet à Renault de préparer la nouvelle génération de caméra de série pour la détection de la route équipées d'objectifs à 85° d'ouverture ;

II.3.2.2 Prétraitements

Correction gamma, Gain, Offset

Le lien entre le signal lumineux (I) et le signal électrique (V) est donné par la *fonction de transfert*. Généralement il y a un écart entre la fonction de transfert du capteur et celle de l'œil humain et pour remédier à cela, une correction gamma est souvent utilisée dans l'étape de prétraitement. Comme l'intensité lumineuse suit une loi du type $I = V^{gamma}$, la correction gamma consiste à compenser la linéarité en appliquant une loi inverse sur la caméra $V = I^{1/gamma}$.

D'autres prétraitements sont également utilisés. Par exemple, l'augmentation du *gain* permettant d'amplifier le signal de l'image avant qu'il ne soit numérisé est un moyen de compenser un manque de lumière mais avec un inconvénient d'amplification du bruit présent dans le signal. Également, lorsque le niveau de luminosité est trop faible, le capteur de la caméra n'est pas sensibilisé car un minimum de luminosité est nécessaire. En appliquant un *offset*, nous pouvons forcer le capteur à envoyer une réponse pour des niveaux de luminosités inférieurs au niveau minimum. Ces deux paramètres, *gain* et *offset*, peuvent être fixés manuellement et automatiquement en fonctions des fonctionnalités de prétraitement embarquées sur la caméra.



FIGURE II.46 – Amplification du signal de l'image par augmentation du gain

Le bruit, le rapport signal/bruit, la correction et la réduction du bruit

Le *rapport signal/bruit* représente le rapport entre la puissance du signal de sortie et le niveau du bruit présent en absence de signal et dépend de l'électronique associée au capteur et du processus de fabrication. Les capteurs CMOS présentent un bruit de configuration fixe qui est souvent appelé *Fixed Pattern Noise*. Cependant, ceci n'est pas un bruit à proprement parler, mais une variation fixe de pixel en pixel qui peut être corrigée.



FIGURE II.47 – Manifestation du Fixed Pattern Noise dans les images prises dans un espace sombre. Mise en évidence du phénomène à travers une variation du paramètre *gamma* et de la désactivation de la correction

Toute image contient une certaine quantité de bruit et les valeurs des pixels ne sont pas des représentations exactes de la scène. Le bruit est capté à partir d'une variété de sources, y compris le capteur d'image et les autres composants électroniques de l'appareil, mais aussi de la lumière elle-même. Différents facteurs influent sur les niveaux de bruit. Par exemple, les capteurs des caméras haute résolution ont généralement des pixels plus petits, donc plus sensibles au bruit.

Dans le même sens, l'augmentation de la fréquence d'acquisition, donc la diminution du temps d'exposition joue un rôle dans l'augmentation du niveau de bruit.

Une technique pour la minimisation du bruit est le traitement par analyse de voisinage, où une image est analysée pour trouver des pixels qui sont très différents dans la couleur ou l'intensité de leurs pixels voisins. Une autre technique est le traitement temporel, où des images consécutives sont comparées pour trouver le bruit potentiel.

Zones d'exposition

Un réglage automatique de l'exposition de la caméra doit décider quelle zone de l'image devrait déterminer la valeur d'exposition. Dans ce sens il y a des caméras qui acceptent des paramètres d'exposition multizones. Par exemple, le premier plan, généralement la partie inférieure d'une image, peut contenir des informations plus importantes que l'arrière-plan, généralement la partie supérieure d'une image. Les zones moins importantes d'une scène, comme le ciel, ne doivent pas déterminer l'exposition globale.

Compensation de contre-jour (éblouissement)

L'exposition automatique tente d'obtenir la luminosité d'une image pour qu'elle apparaisse comme si l'œil humain voyait la scène, mais cela n'est pas simple à réaliser. Un fort rétro-éclairage induit une perception sombre des objets situés au premier plan. Les caméras avec compensation du rétroéclairage cherchent à ignorer des zones de haute illumination. Un exemple de zones de haute illumination est le soleil. Les images présentées dans la figure II.48 montrent une tache noire à la place du soleil. Éliminer ces sources de forte illumination, permet aux objets au premier plan d'être vus, bien que les zones claires soient surexposées.



FIGURE II.48 – Compensation de contre-jour : détection et isolement des sources de forte illumination (comme le soleil)

WDR (Wide Dynamic Range) ou HDR (High Dynamic Range)

Dans une scène avec des zones très claires et sombres, une caméra typique produira une image où les objets dans les zones sombres ne seront guère visibles. La technique WDR¹ ou HDR résout ce problème en utilisant d'expositions différentes pour les objets de la même scène.

Un pixel d'une caméra peut être comparé à un récipient à remplir avec des photons. Le *container* prendra en photons jusqu'à ce qu'une limite d'absorption soit atteinte. Avec ce mode, la réponse du capteur ne sera plus linéaire. Les pixels lumineux qui atteignent un certain niveau de tension programmable seront *écrêtés*². Le mode HDR permet l'assombrissement très lumineux

1. **Wide Dynamic Range** (Gamme dynamique étendue); 2. «Remplacement des parties d'un signal dont la valeur dépasse, en valeur absolue, un certain seuil, par la valeur de ce seuil» [Larousse];

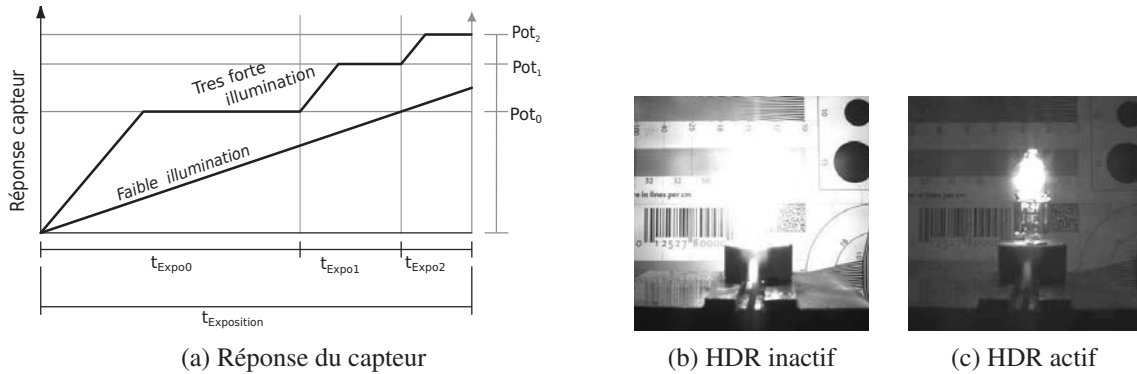


FIGURE II.49 – High Dynamic Range - Graphique de l'ajustement de la réponse du capteur d'image avec illustration des trois paliers d'écrtage ajustables dans une seule exposition en configurant les tranches de temps respectives (t_{Exp0} , t_{Exp1} , t_{Exp2}), et les niveaux de potentiel d'écrtage (Pot_0 , Pot_1 , Pot_2) [Baumer, 2016]

(saturés) de certains pixels, mais il est impossible de faire des pixels sombres plus lumineux. Pour plus de détails nous invitons le lecteur à consulter le *Programmer's Guide for feature HDR* mis à disposition par l'entreprise Baumer [Baumer, 2016].

II.3.2.3 Caméra du véhicule prototype

Comme nous l'avons déjà précisé, le véhicule prototype dispose d'une caméra avant positionnée sous le pare-brise, en complément du système de capteurs ultrason.

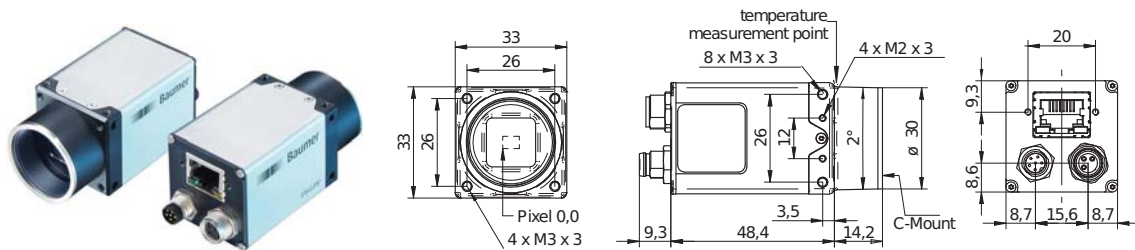


FIGURE II.50 – Caméra Baumer VisiLine VLG-22M

Nous avons équipé le véhicule d'une caméra VLG-22M de la gamme VisiLine de Baumer (figure II.50), disposant d'un capteur CMOS 2/3" (CMOSIS CMV2000) nous permettant d'obtenir, via une interface Gigabit Ethernet PoE¹, une image numérique monochrome (en niveaux de gris : mono8² ou mono12³) de dimensions 2040 × 1084 pixels à une fréquence maximale de 55 fps. L'objectif choisi est un fish-eye, avec une focale de 2,7 mm qui permet d'apercevoir l'environnement sous une ouverture angulaire approximativement 185°. Les figures II.58(a) à (c) illustrent la largeur du champ de vue.

II.3.2.4 Calibration de la caméra

Si l'on souhaite utiliser la caméra comme instrument de mesure, une étape préliminaire est de déterminer les paramètres intrinsèques (propriétés propre au capteur et au système optique) et extrinsèques (position du capteur d'image sur le véhicule) de la caméra.

1. Power over Ethernet; 2. 2⁸ niveaux de gris; 3. 2¹² niveaux de gris;

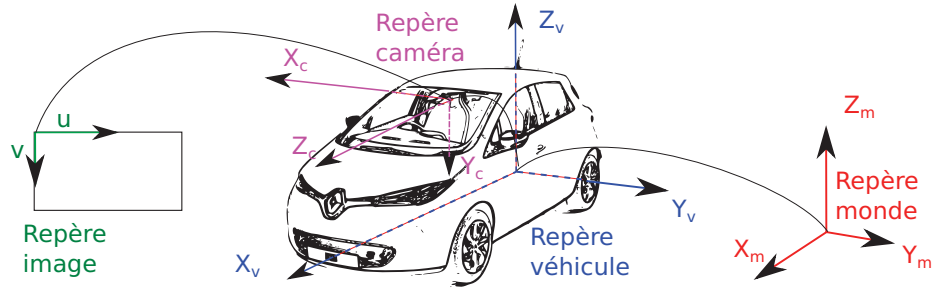


FIGURE II.51 – Les repères utilisés : repère monde, repère véhicule, repère caméra et repère image

Modèles de paramétrisation intrinsèque des caméras

Calibrer intrinsèquement une caméra ou étalonner une caméra signifie de modéliser le processus de formation des images dans le sens de trouver la relation entre les coordonnées métriques spatiales d'un point et sa position en pixels dans l'image. Il existe nombreux modèles de paramétrisation intrinsèque de la caméra [Sturm et al., 2011; Lébraly, 2012], cependant les plus utilisés sont : le modèle sténopé, le modèle direct, le modèle indirect et le modèle unifié [Lébraly, 2012]. Le modèle sténopé (également appelé modèle perspectif ou pin-hole ou modèle central) où tous les rayons optiques s'intersectent en un même point appelé centre de projection (ou centre optique) (figure II.52(a)). Par conséquent, le point image $m = [x, y]^T$ correspondant à un point $P = [X_c, Y_c, Z_c]^T$ de l'espace 3D est situé à l'intersection du plan image (situé à une distance focale f) avec le rayon le relie le point P avec le centre optique de la caméra. On note :

- $\bar{P}_c = [X_c, Y_c, Z_c, 1]^T$ les coordonnées homogènes du point 3D P dans le repère de la caméra ;
- $\bar{m} = [x, y, 1]^T$ les coordonnées homogènes du point 2D dans le repère de la caméra ;
- $\bar{p} = [u, v, 1]^T$ les coordonnées homogènes du point dans le repère de l'image.

Pour le modèle sténopé, le passage du point 3D en coordonnées homogènes au point 2D du plan normalisé se réalise à travers l'équation (II.1). Le passage du repère caméra au repère image se réalise utilisant la transformation affine de l'équation (II.2).

$$\bar{m} = [\mathbf{I}_3 \mid 0_{3 \times 1}] \bar{P}_c \quad (\text{II.1})$$

$$\bar{p} = \mathbf{K} \bar{m} \quad (\text{II.2})$$

$$\mathbf{K} = \begin{bmatrix} \frac{f}{d_x} & 0 & u_0 \\ 0 & \frac{f}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.3})$$

avec \mathbf{K} la matrice des paramètres intrinsèques (cf. équation (II.3)) où (u_0, v_0) sont les coordonnées en pixels du point principal dans l'image, d_x et d_y sont les dimensions élémentaires d'un pixel de la matrice et f est la distance focale.

En pratique l'objectif induit aberrations¹ chromatiques et géométriques [Welford, 1986; Mahajan, 1998; Kingslake and Johnson, 2009]. Parmi les aberrations géométriques, les plus importantes sont les déformations radiales. Dans la littérature cette composante radiale peut être modélisée à travers d'un polynôme (généralement de degré 5). Le passage des coordonnées non-distordues $m_u = [x_u, y_u]^T$

1. Dans l'optique, on appelle aberrations les imperfections des images autres que celles dues à la diffraction ;

aux coordonnées distordues $m_d = [x_d, y_d]^T$ (figure II.52(b)) se réalise à travers un formalisme mathématique nommé modèle direct. Le passage des coordonnées distordues aux coordonnées non-distordues, sera appelé indirect (figure II.52(c)).

Pour le modèle direct, m_d s'écrit sous la forme :

$$m_d = (1 + D(r_u)) m_u \quad (\text{II.4})$$

$$D(r_u) = \sum_{k=1}^n a_k r_u^{2k} \quad (\text{II.5})$$

où D est le polynôme de degré n en r_u^2 avec $r_u = \sqrt{x_u^2 + y_u^2} = \|m_u\|$ la distance radiale entre l'axe optique et le point m_u dans le plan normalisé.

Pour le modèle indirect, m_u s'écrit sous la forme :

$$m_u = (1 + I(r_d)) m_d \quad (\text{II.6})$$

$$I(r_d) = \sum_{k=1}^n b_k r_d^{2k} \quad (\text{II.7})$$

où I est le polynôme de degré n en r_d^2 avec $r_d = \sqrt{x_d^2 + y_d^2} = \|m_d\|$ la distance radiale entre l'axe optique et le point m_d dans le plan normalisé.

Calibrer une caméra revient à retrouver les paramètres intrinsèques qui la définissent : la matrice \mathbf{K} et les coefficients du polynôme de distorsion. Une hypothèse fréquemment utilisée est que la détection des points dans les images distordues est affectée par un bruit gaussien. Sous cette hypothèse, un estimateur de maximum vraisemblance est utilisé pour construire une fonction de coût à minimiser. Ainsi, pour le modèle direct et le modèle indirect les fonctions de coût sont présentées par les équations (II.8) et (II.9).

$$\sum_{i=1}^N \|D(\pi(\mathbf{K} [\mathbf{I}_3 \mid 0_{3 \times 1}] \bar{P}_{c_i})) - \hat{p}_i\|^2 \quad (\text{II.8})$$

$$\sum_{i=1}^N \|\pi(\mathbf{K} [\mathbf{I}_3 \mid 0_{3 \times 1}] \bar{P}_{c_i}) - I(\hat{p}_i)\|^2 \quad (\text{II.9})$$

où π représente la fonction qui permet de passer des coordonnées homogènes aux coordonnées cartésiennes d'un point 2D, \hat{p}_i la détection d'un point dans le plan image et N le nombre des points utilisés.

Le dernier modèle qui nous intéresse, est le modèle unifié proposé par C. Geyer et K. Daniilidis [Geyer and Daniilidis, 2000]. Celui-ci à l'avantage de mieux modéliser les caméras grand angle, fish-eye ou encore catadioptriques [Ying and Hu, 2004; Barreto, 2006]. Dans la figure II.52(d) on note $[X_c, Y_c, Z_c]^T$ et $[X_s, Y_s, Z_s]^T$ les coordonnées du point 3D P dans le repère de la caméra respectivement de la sphère unitaire de centre $[0, 0, Z_c]^T$, avec $Z_c = \xi \geq 0$. Soit Q_s la projection du point P sur la sphère unitaire. Alors,

$$Q_s = \frac{P_s}{\rho} = \frac{1}{\rho} [X_s, Y_s, Z_s]^T \quad (\text{II.10})$$

$$\rho = \sqrt{X_s^2 + Y_s^2 + Z_s^2} \quad (\text{II.11})$$

Soit m la projection perspective de Q_s sur le plan normalisé $Z_c = 1$. Alors les coordonnées homogènes \bar{m}_c de m sont données par l'équation (II.12). Enfin, on utilise la transformation affine

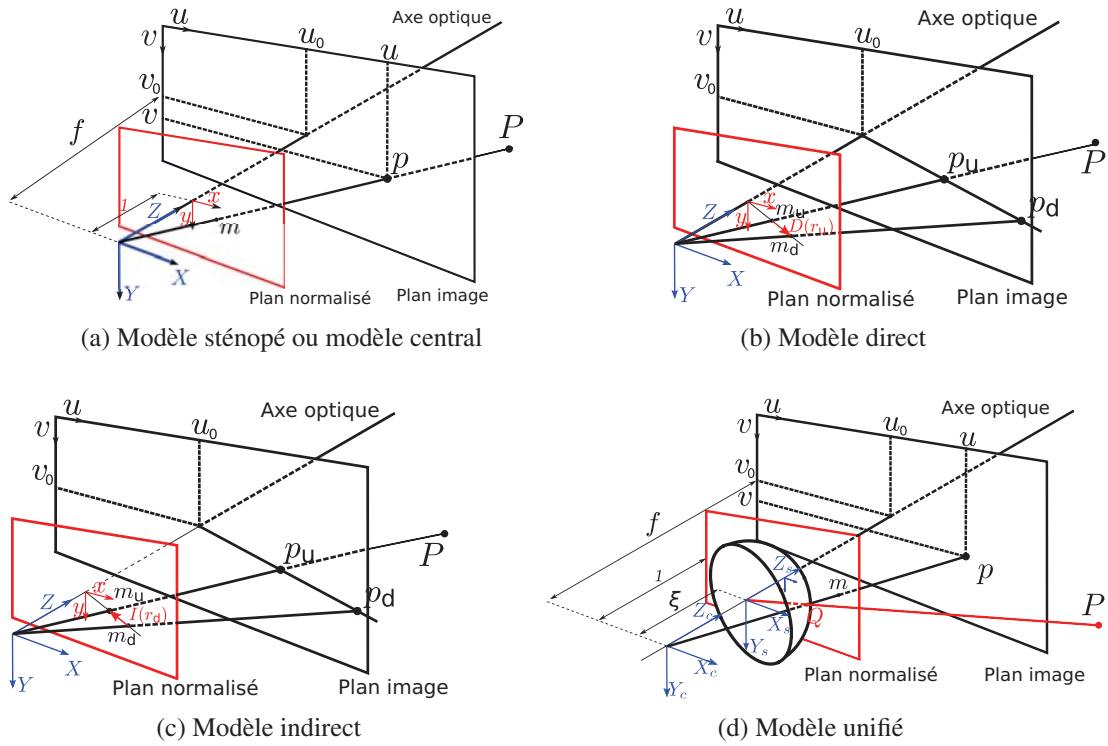


FIGURE II.52 – Quatre modèles de paramétrisation intrinsèque les plus utilisés : modèle sténopé, le modèle direct, le modèle indirect et le modèle unifié [Lébraly, 2012]

pour obtenir le point p (cf. équation (II.13)).

$$\bar{m}_c = \left[\frac{X_s}{Z_s + \rho\xi}, \frac{Y_s}{Z_s + \rho\xi}, 1 \right]^T \quad (\text{II.12})$$

$$\bar{p}_c = \mathbf{K} \bar{m}_c = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \bar{m}_c \quad (\text{II.13})$$

P. Lébraly présente également la démonstration que le modèle unifié permet de retrouver analytiquement le point sur la sphère à partir d'un point image à travers un procédé appelé *lancer de rayon* [Barreto, 2004; Ying and Hu, 2004; Courbon et al., 2007; Courbon, 2009; Lébraly, 2012].

$$Q_s = \eta \left[x, y, 1 - \frac{\xi}{\eta} \right]^T \quad (\text{II.14})$$

$$\text{avec } \eta = \frac{\xi + \sqrt{1 + (x^2 + y^2)(1 - \xi^2)}}{x^2 + y^2 + 1} \quad (\text{II.15})$$

Étalonnage intrinsèque d'une caméra

Pour le calcul des intrinsèques d'une caméra, des images d'une mire d'étalonnage sont prises. Dans ce mémoire nous présentons deux modèles à travers les figures II.53(a) à (d).

Utilisant ces mires, une détection automatique est réalisée du damier ou des motifs encodés. P. Lébraly propose l'utilisation d'un code-barres circulaire composé d'un entête, d'un label

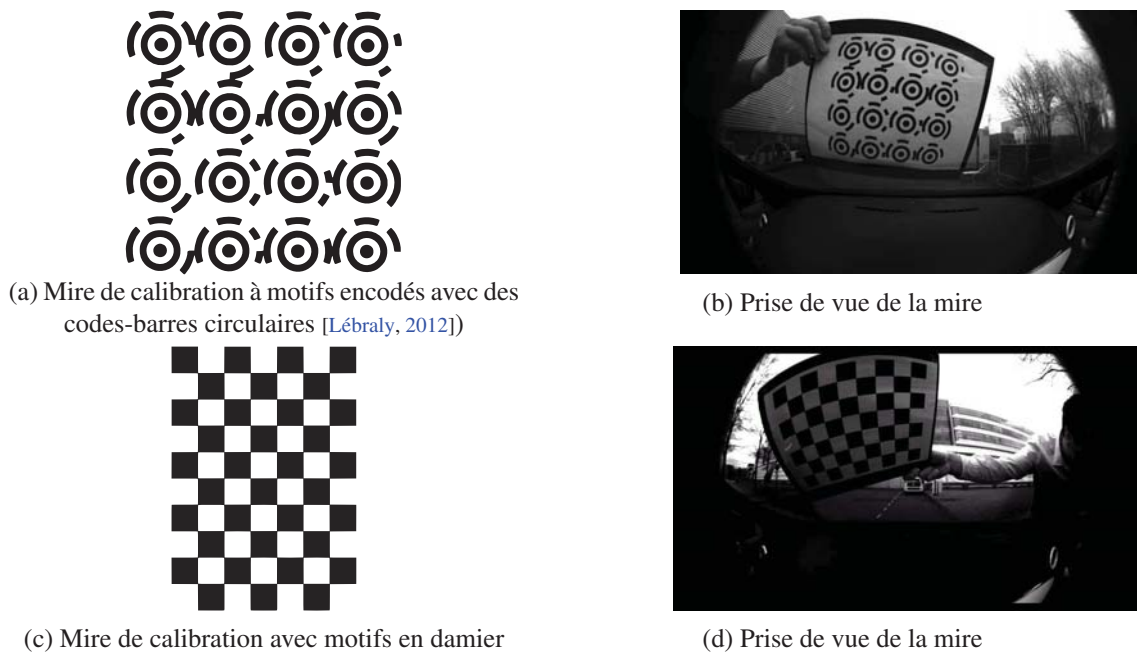


FIGURE II.53 – Mires de calibration et prises de vue réalisées lors de l'étape de calibration de la caméra *Baumer VisiLine VLG-22M* équipée d'un objectif grand angle

et d'un bit de parité [Lébraly, 2012]. La mise en correspondance des points situés à des distances connues sur le plan de la mire avec ceux détectés dans les images permet de trouver les paramètres recherchés [Tarel, 1994; Brand, 1995; Gurdjos et al., 2006; Bradski and Kaehler, 2008; Soille, 2013].

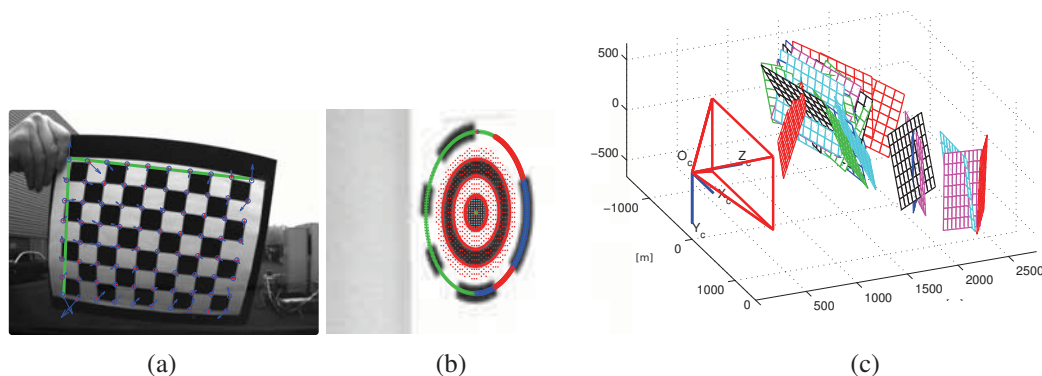


FIGURE II.54 – Résultat de la détection des points d'intérêt sur la mire damier (figure (a)) [Bouguet, 2004] ou sur la mire composée des motifs de code-barres circulaires (figure (b)) [Lébraly, 2012] et exemple projection des plans des mires dans un repère 3D centré sur la caméra (figure (c))

Grâce aux paramètres intrinsèques retrouvés suite à l'étalonnage de la caméra, partant d'une image distordue (figure II.55(a)) nous sommes capables de la corriger et de la tronquer en fonction du champ de vue souhaité. La figure II.55(b) présente le résultat de la correction. Distordre une image complète est un processus coûteux en temps de calcul est pas forcément nécessaire à réaliser. Dans ce mémoire nous reviendrons sur le calcul des distances aux obstacles à travers un procédé nommé *Structure from Motion*. Le suivi des points d'intérêts dans plusieurs images successives permettra de récupérer leur position 3D dans le repère de la caméra. Par conséquent, le processus de correction de la distorsion sera appliqué à un ensemble limité des points d'intérêt [§IV.4].



(a) Image complète de la caméra grand angle 2 044 px × 1 084 px



(b) Image corrigée et tronquée 1 024 px × 288 px

FIGURE II.55 – Exemple de transformation d'une image brute de la caméra grande angle et dans une image tronquée et corrigée

Paramètres extrinsèques

Calibrer une caméra extrinsèquement signifie de trouver les paramètres qui déterminent la position de la caméra dans un repère connu. Dans cette situation, nous sommes intéressés à retrouver la position de la caméra, dans le repère véhicule \mathfrak{R}_v situé au milieu de l'essieu des roues arrières, voir figure II.51. Cela revient à retrouver la matrice de rotation $R_{(v \rightarrow c)}$ et le vecteur de translation $\vec{T}_{(v \rightarrow c)}$ caractéristiques au changement du repère véhicule \mathfrak{R}_v au repère caméra \mathfrak{R}_c .



FIGURE II.56 – Emplacement et fixation de la caméra dans l'habitacle (derrière le pare-brise)

$$\vec{T}_{(v \rightarrow c)} = [X_c, Y_c, Z_c]^T \quad (\text{II.16})$$

$$\mathbf{R}_{(v \rightarrow c)} = \text{Rot}_z(\alpha_z) \text{Rot}_y(\alpha_y) \text{Rot}_x(\alpha_x) \quad (\text{II.17})$$

$$= \begin{bmatrix} \cos \alpha_z & -\sin \alpha_z & 0 \\ \sin \alpha_z & \cos \alpha_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha_y & 0 & \sin \alpha_y \\ 0 & 1 & 0 \\ -\sin \alpha_y & 0 & \cos \alpha_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_x & -\sin \alpha_x \\ 0 & \sin \alpha_x & \cos \alpha_x \end{bmatrix} \quad (\text{II.18})$$

où X_c, Y_c, Z_c sont les coordonnées de la caméra dans le repère véhicule, $\text{Rot}_x(\alpha_x)$, $\text{Rot}_y(\alpha_y)$, $\text{Rot}_z(\alpha_z)$ sont les trois matrices de rotation élémentaires respectivement suivant les axes x, y, z avec $\alpha_x, \alpha_y, \alpha_z$ les angles de roulis, tangage et lacet associés au véhicule¹.

La fixation de la caméra dans le véhicule et les mesures de sa position et orientation dans l'habitacle du véhicule ont été réalisées par un spécialiste dans le domaine. Grâce à des instruments de mesure de précision, les paramètres extrinsèques qui relient la position de la caméra au repère véhicule, ont été mesurés. Ce que nous cherchons lors de la fixation de la caméra dans l'habitacle est de la positionner dans l'axe de véhicule (la composante Y du vecteur de translation $\vec{T}_{(v \rightarrow c)}$ est nulle) le plus près du pare-brise. Généralement les caméras sont légèrement inclinées pour ne pas voir trop de ciel mais, dans notre cas, l'ouverture de 185° nous permet de la positionner horizontalement (angle de tangage nul). Nous cherchons également que la caméra ne soit pas

1. Également appelé angles de Cardan ;

ournée dans son axe (angle de roulis nul) et ni dans le plan horizontal (angle de lacet nul). Dans la figure II.54(c) nous avons montré que lors de l'étape d'étalonnage de la caméra, en plus des paramètres intrinsèques il est possible de calculer les positions des mires dans le repère de la caméra. Dans la figure II.57 nous montrons qu'il est possible de retrouver également les positions des caméras par rapport à une mire.

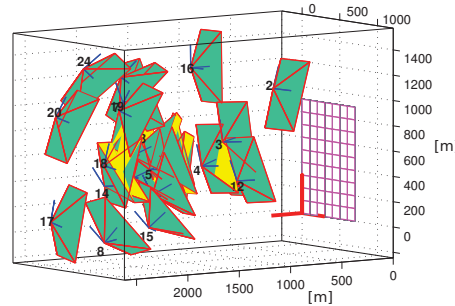


FIGURE II.57 – Positions de la caméra par rapport à la mire pour chaque prise de vue

En pratique, pour la simplicité du calcul la mire est fixée sur le sol ou sur un mur. Déterminer la position de la caméra par rapport à la mire et mesurer la position du véhicule par rapport à la même mire permet de retrouver précisément la matrice de rotation $\vec{R}_{(v \rightarrow c)}$ et le vecteur de translation $\vec{T}_{(v \rightarrow c)}$ de la caméra dans le repère du véhicule.



(a) Caméra située au niveau de la planche de bord



(b) Caméra située à mi-distance entre la planche de bord et le rétroviseur intérieur



(c) Caméra située au niveau du rétroviseur intérieur

FIGURE II.58 – Variation de la position de la caméra à l'intérieur du véhicule en vue du choix du meilleur champ de vue de la scène et du meilleur emplacement de fixation

Les figures II.58(a) à (c) illustrent différentes prises de vue réalisées lors de la phase de réflexion sur le choix de la position de fixation de la caméra. La position de fixation de la caméra a été choisie au niveau du rétroviseur intérieur (figure II.58(c)). Ce qu'on peut facilement constater est qu'un tiers de l'image est occupée par la capture de la planche de bord, donc de l'information inutile. Toutefois, la grande ouverture angulaire nous permet d'apercevoir les obstacles proches situés des deux côtés du véhicule. Nous reviendrons sur cette propriété de large champ de vision dans la section §IV.4 dédiée à la détection d'obstacles par approche *Structure from Motion*.

Par la suite, nous présentons des capteurs proprioceptifs installés sur le véhicule et susceptible d'être utilisés dans l'application de voiturier autonome.

II.3.3 Capteurs proprioceptifs

Encodeurs rotatifs (incrémentaux et absolus)

En plus de la caméra et des capteurs ultrason, nous disposons de l'information odométrique mesurée par les capteurs intégrés dans le système ESP¹. Chaque roue est équipée d'un encodeur avec une résolution angulaire de $2\pi/48$ rad (48 tops par tour de roue). Ce qui revient à une résolution linéaire d'approximativement 4 cm pour un rayon de roue de 0,31 m.

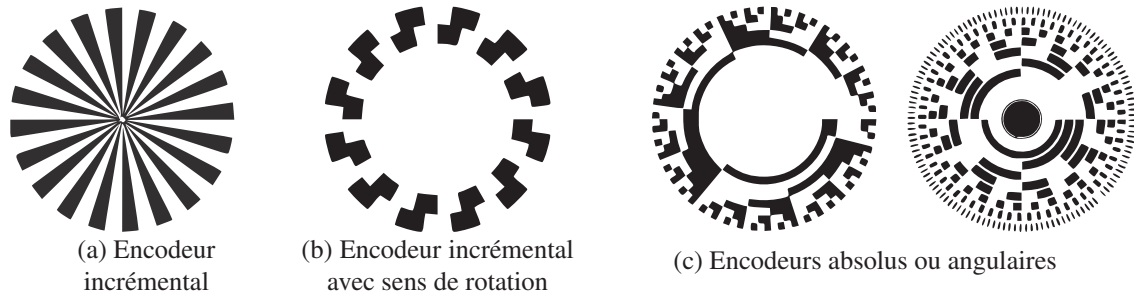


FIGURE II.59 – Disques d'encodage relatif et absolu

L'information odométrique est considérée sécuritaire uniquement dans le calculateur ESP. L'inconvénient est que nous n'avons pas directement accès à l'information dans le calculateur ESP mais, aux valeurs diffusées par celui-ci sur le réseau CAN. Ces informations transmises sur CAN ne sont pas datées et la fréquence est approximativement de 50 Hz. Nous utilisons la notion de fréquence approximative, car dans la priorisation des informations, l'information odométrique est considérée moins importante. En fonction du taux de charge du réseau CAN, il est permis de ne pas envoyer les paquets correspondants. Pour pallier à ce problème de sauts de paquets, le système ESP utilise quatre compteurs sur 14 bits ($2^{14} = 16\,384$ valeurs), un pour chaque roue, qui sont tenus à jours même s'il ne sont pas forcément envoyés sur le CAN. Ce système permettant ainsi d'avoir de manière robuste la valeur exacte d'avancement de chaque roue. Un autre problème important à signaler vient du type de disque d'encodage utilisé. Dans notre cas, le disque contient une seule piste d'encodage (figure II.59(a)) ce qui ne nous permet pas de savoir le sens de rotation de la roue.

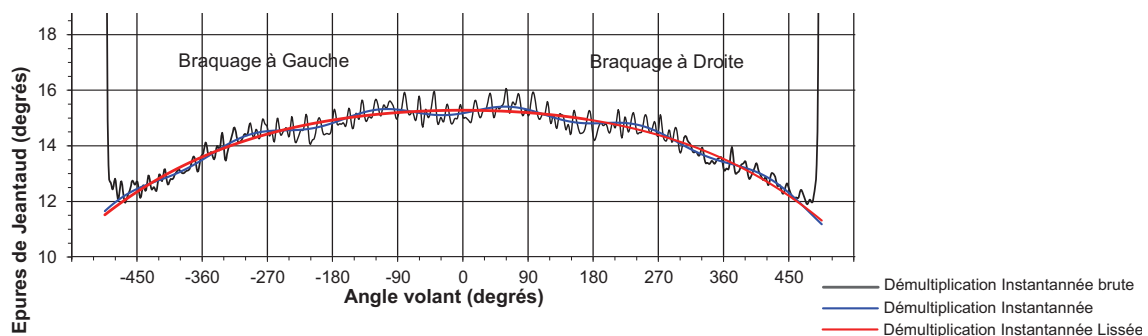


FIGURE II.60 – Mesure de la cinématique de direction sur banc de test : épures de Jeantaud et rapport de démultiplication

Par ailleurs, un encodeur d'angle est fixé sur la colonne de direction et permet de mesurer la position exacte du volant (figure II.59(c)). Du fait qu'une variation petite de l'angle volant a un effet important sur la trajectoire du véhicule à haute vitesse, l'encodeur, cette fois-ci a une

1. Electronic Stability Program;

résolution de 16 bits pour une plage maximale de $[-495^\circ, +495^\circ]$. La relation entre l'angle volant et l'angle des deux roues se fait à travers l'épure de Jeantaud¹. La figure II.60 illustre les mesures d'angle volant et des angles des roues avant, réalisées sur un banc de test pour un véhicule Zoe de Renault, la base mécanique de notre voiture de test. Le tableau II.5 reprend ces résultats et illustre, de manière synthétique les données mesurées.

Moyenne du Jeantaud mesurée en butées :	5,7°	Rapport de démultiplication instantanée à 0° volant :	15,1
Moyenne du Jeantaud calculée en butées :	11,0°	Rapport de démultiplication instan. lissé à 0° volant :	15,3
Moyenne du sous Jeantaud en butées :	5,3°	Moy. des rapports de démultiplication instan. lissés en butées :	11,5
Angle volant de butée à butée :	985,1°	Amplitude crête-crête de la modulation entre $\pm 90^\circ$ volant :	0,3
Angle roues moyen en butée droite :	34,7°	Taux de modulation entre $\pm 90^\circ$ volant :	2,2%
Angle roues moyen en butée gauche :	-34,7°	Angle de calage de la modulation :	-26,0°
		Diamètre de braquage moyen estimé :	10,9 m

TABLE II.5 – Résultats et analyse de la cinématique de direction

Gyromètre, Accéléromètre, Centrale Inertielle

Le gyromètre est un capteur qui mesure une vitesse angulaire par rapport à un référentiel inertiel. Souvent le gyromètre est assimilé au gyroscope² qui est un capteur de position angulaire. L'accéléromètre, comme le nom l'indique, est un capteur capable de mesurer une accélération selon une ou plusieurs directions. En fonction de la technologie utilisée, l'accélération est saisie de manière indirecte soit par détection du déplacement de la masse sismique du capteur soit par détection de la force inertielle induite par de la masse sismique du capteur. Dans l'industrie automobile, ce capteur est utilisé dans les systèmes de contrôle de la stabilité du véhicule dans les virages. En plus, il a repris le rôle de l'inclinomètre et permet également d'intervenir dans les stratégies d'arrêt et de redémarrage en côte avec serrage et desserrage automatique du frein de parking électrique.

La mise en commun de 3 gyromètres et 3 accéléromètres crée un nouveau capteur, nommée centrale inertielle³. Ce capteur est généralement utilisé dans l'industrie automobile uniquement en instrumentation, car le prix important ne permet pas de le déployer en série à ce jour. En revanche le système ESP du véhicule est équipé des accéléromètres et gyromètres à bas coût, sensibles uniquement à de forts mouvements. Le rôle de l'ESP est de mettre en corrélation l'angle volant, la vitesse de lacet et l'accélération latérale du véhicule. En cas de détection de dérapage, il est capable d'actionner indépendamment les freins des quatre roues pour remettre le véhicule sur la trajectoire demandée par le conducteur. Comme l'ESP est fonctionnel dans des situations à forte dynamique, les capteurs dans sa composition sont conçus pour être sensibles sur une plage en corrélation avec l'utilisation. Par conséquent nous ne pouvons pas les utiliser dans notre application de voiturier autonome où les accélérations sont très faibles et la vitesse ne dépasse pas 10 km/h.

Compas électronique

Même si notre véhicule n'en dispose pas, nous avons mené une réflexion pour l'équiper d'un compas électronique⁴. Ce capteur est un instrument de navigation car il est capable d'indiquer le Nord magnétique de la Terre. Même si cet instrument de mesure est largement répandu dans la navigation aérienne et navale, sa présence se fait moins sentir dans l'industrie automobile. Les principales technologies utilisées dans les compas électroniques sont la magnétorésistivité, la

1. Ou épure d'Ackermann; 2. «Appareil inventé en 1852 par Foucault, formé d'un rotor de masse élevée, dont l'axe est suspendu à un support formé de deux cercles articulés entre eux et au rotor par des suspensions à cardan» [Larousse]; 3. IMU - Inertial Measurement Unit, en anglais; 4. Il existe également de compas magnétique, gyroscopique et satellitaire;

magnétoinduction, l'effet Hall et le fluxgate. L'inconvénient majeur, qui nous a convaincu de ne pas continuer à l'exploiter, est que ces techniques de détection du champ magnétique terrestre sont facilement perturbées par le champ magnétique propre au véhicule ou à l'environnement.

Capteur de lumière et capteur de pluie

Un autre capteur qu'on cite ici, est le capteur de lumière utilisé initialement pour l'allumage des feux du véhicule. Son utilité a été étendue également dans les applications de traitement d'images, car il permet d'affiner les réglages des algorithmes de prétraitements embarqués dans la caméra pour assurer un niveau de clarté optimal de l'image.



FIGURE II.61 – Détecteur de pluie : Par temps sec le rayon est réfléchi dans le bloc optique (figure (a)), en revanche par temps de pluie, une partie de la lumière se réfracte (sort du bloc) en passant par la goutte d'eau (figure (b))

Le dernier capteur qui équipe le véhicule et qui nous intéresse, est le détecteur de pluie (figures II.61(a) et (b)). Il utilise un rayon de lumière qui est envoyé dans le pare-brise. L'angle du rayon est calculé pour qu'il se réfléchisse à l'intérieur du verre et finisse dans un détecteur de lumière. Mais, si des gouttes d'eau se déposent sur le pare-brise, une partie du rayon se réfracte et moins de lumière arrive au détecteur. Savoir quand il pleut permet de mettre en marche les balais d'essuie-glace et d'appliquer des stratégies différentes dans les algorithmes de perception par caméra ou par capteurs ultrason et éventuellement de baisser la vitesse du véhicule.

II.4 Bilan sur la perception

Dans ce chapitre nous nous sommes attardés sur la problématique de la perception de l'environnement. Après avoir analysé les notions d'obstacle physique et d'obstacle réglementaires, nous avons réalisé un état de l'art des moyens d'acquisition de l'information adaptés à l'automatisation de la conduite. Nous avons également abordé le sujet de l'utilisation des cartes de navigation ou de la communication V2X car ces outils apportent de l'information permettant de simplifier la tâche de perception.

A la suite de l'état de l'art nous avons détaillé les moyens de perception du véhicule prototype tout en justifiant nos choix par rapport à l'application de voiturier autonome. Nous avons retenu d'utiliser un système de capteurs ultrason de pré-série, une caméra dotée d'un objectif fish-eye orientée vers l'avant du véhicule, des capteurs odométriques et également toutes les informations disponibles sur le réseau de communication CAN relatives à l'état interne de fonctionnement du véhicule.

Dans les chapitres III à V nous détaillons la manière dont ces capteurs sont utilisés pour la localisation et pour l'analyse de l'espace dans lequel le véhicule évolue.

CHAPITRE III

Localisation

Sommaire

II.1	Problématique	26
II.2	Perception : État de l'art	27
II.2.1	Perception de l'environnement et moyens d'acquisition de l'information	27
II.2.2	Cartographie	37
II.2.3	Communication V2X	41
II.2.4	Détection et suivi d'obstacles dans le domaine routier	43
II.2.5	Détection d'obstacles pour une situation de parking	44
II.3	Moyens de perception du véhicule prototype	47
II.3.1	Perception par capteurs ultrason	48
II.3.2	Perception par caméra	55
II.3.3	Capteurs proprioceptifs	69
II.4	Bilan sur la perception	71

III.1 Problématique

Ce chapitre est dédié à la fonction de localisation du système de voiturier autonome. Dans la section §I.6 nous avons exposé l'enchaînement typique des événements en concluant qu'avant l'utilisation courante du système, il faudrait passer par une phase d'apprentissage. Durant cette phase, le système collecte des informations liées au trajet et à l'environnement. À la fin de ce processus, le système de voiturier autonome devra être exploitable.

Dans ce chapitre nous nous proposons d'approfondir la problématique de localisation du véhicule pour le scénario typique énoncé. Lors de l'utilisation courante, le système devra être suffisamment flexible pour permettre au conducteur de déposer le véhicule dans une zone assez large autour du point initial d'apprentissage. En plus, le véhicule devra être en mesure de contourner un obstacle qui ne se trouve pas sur la trajectoire initiale. Cela implique le fait que le véhicule ne suivra pas nécessairement la trajectoire d'apprentissage. Par conséquent, il n'est pas suffisant de développer une approche de localisation et de suivi précis d'une trajectoire identique. Il faudra donc tenir compte dans la construction de la fonction de localisation, des implications de changement d'environnement dues aux changements subis par la trajectoire.

Nous utilisons dans ce chapitre la notion de grille d'occupation que nous avons brièvement évoqué dans la section §II.2.2.3 et que nous détaillerons dans le chapitre IV. Nous supposons pour l'instant disposer d'une carte contenant à la fois l'espace libre et ainsi que les obstacles détectés le long de la trajectoire. Nous utilisons l'information de couloir disponible et nous nous donnons comme objectif de créer un module de localisation en charge de surveiller la position du véhicule. L'étude approfondie du cas d'utilisation du système de voiturier autonome a relevé le fait qu'il est souhaitable d'avoir une localisation précise dans le couloir de navigation, mais ce n'est pas une contrainte imposée. Plus exactement, on peut tolérer des déviations de trajectoire, mais à l'approche de la destination, le véhicule devra se localiser précisément.

Dans la section §III.4 nous présentons la stratégie de localisation qui permettra d'assurer la disponibilité de la fonction de localisation même avant la demande expresse du conducteur de démarrage en autonome. Nous étudions également la possibilité de sortie du couloir et de localisation même dans les zones non cartographiées. Dans la section §I.6 nous avons exposé la possibilité de se confronter à une configuration multi-trajets jusqu'à la place de parking. Dans le cas où le conducteur engage le système de voiturier autonome depuis une position non apprise ou incertaine, le système ne devra pas permettre le démarrage en automatique. Cependant, le système peut suggérer à l'utilisateur de continuer en mode manuel jusqu'à ce que sa position soit détectée. Par la suite, nous dédions une section à l'état de l'art en essayant de donner une image globale des travaux entrepris dans la communauté de la robotique sur la localisation.

III.2 Localisation : État de l'art

Où suis-je ? est une des questions à laquelle notre cerveau répond presque en continu sans que nous nous en rendions compte. Connaître son emplacement et être capable de naviguer vers d'autres endroits est extrêmement important. L'acte de trouver son emplacement est connu sous le nom de *localisation*. La complexité de la tâche de localisation pose essentiellement trois problèmes [Singhal, 1997; Negenborn, 2003; Filliat, 2011] :

- le problème du suivi continu de position (*position tracking*) : partant d'une position connue, le système doit continuer de se localiser durant son déplacement ;
- le problème de la localisation globale (*wake-up robot* ou *global positioning*) : dès la mise en service, le véhicule doit se localiser sans qu'aucune estimation initiale ne soit fournie ;
- le problème du robot kidnappé (*kidnapped robot problem*) : le robot sait exactement où il se trouve mais soudainement il est transféré (kidnappé) à un autre endroit ; même suite à ce changement il est censé se localiser.

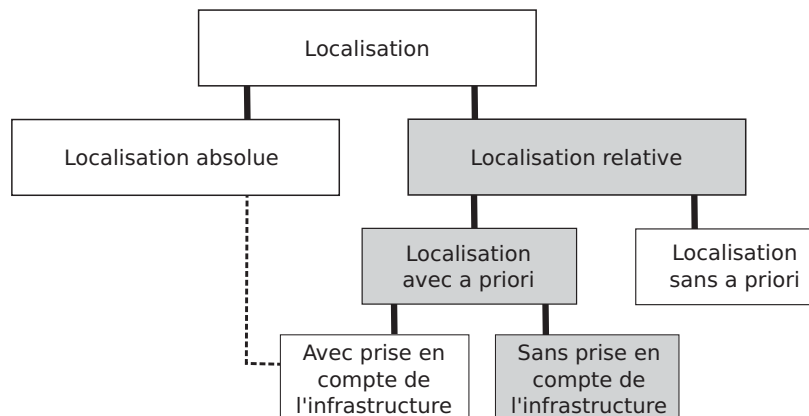


FIGURE III.1 – Techniques de localisation (la ligne pointillée désigne le fait que certaines techniques de localisation relative deviennent des techniques de localisation globale comme par exemple la localisation dans une carte à l'échelle d'une ville voire un pays)

Pour répondre à ces défis, la robotique a développé plusieurs approches suivant l'accès à un système de localisation absolue ou à l'utilisation ou non d'un système fournissant des informations a priori. En principe tout capteur que nous avons présenté dans la section §II.2.1 peut être utilisé pour la localisation. Nous nous concentrons uniquement sur les techniques en cohérence avec nos choix de capteurs : GPS, odomètres, caméra, système de capteurs ultrason (voir section §II.3).

Dans la littérature scientifique deux grandes classes de localisation sont présentées : la localisation absolue et celle relative à un repère ou un amer (figure III.1). Dans la première classe se retrouvent les systèmes du type GNSS¹ qui donnent la position du récepteur en coordonnées terrestres. La deuxième classe, comme le nom l'indique, fournit une position par rapport à une autre, appelée position de référence ou simplement référentiel. Ces techniques seront détaillées dans les sous-sections suivantes.

III.2.1 Techniques de localisation absolue

La localisation absolue était synonyme jusqu'à il y a quelques années de la localisation par GPS. Un autre capteur qui donne uniquement une orientation absolue est le compas électronique [Rahok et al., 2011] que nous avons déjà présenté dans §II.3.3.

Nous verrons par la suite que l'enrichissement continu des cartes géolocalisées de très grandes dimensions (à l'échelle d'une ville voire d'un pays) [§III.2.1.2] ainsi que le développement en parallèle des techniques de localisation dans une carte [§III.2.2.2] ouvrent une nouvelle voie vers la localisation absolue.

1. Global Navigation Satellite System ;

III.2.1.1 Global Positioning System (GPS)

Le GPS¹, littéralement *système mondial de positionnement* ou géo-positionnement par satellite, est un système de navigation et de localisation (figure III.2(a)) développé par le département de défense des États-Unis. Le projet débute en 1973 et cinq ans plus tard, le 22 février 1978, le premier satellite GPS, nommé OPS-5 111, est envoyé sur orbite. Le 5 février 2016 on comptait l'arrivée sur orbite du 73^{ème} satellite GPS, mais parmi ceux-ci uniquement 30 sont fonctionnels et un 31^{ème} est en phase de mise en service. Même si couramment on parle de GPS, nous rappelons qu'il y existe plusieurs systèmes de navigation par satellite comme : GLONASS² [Jokinen et al., 2013], Galileo³ [Paziewski and Wielgosz, 2015], BeiDou NSS⁴, COMPASS⁵, QZSS⁶ ou bien IRNSS⁷.

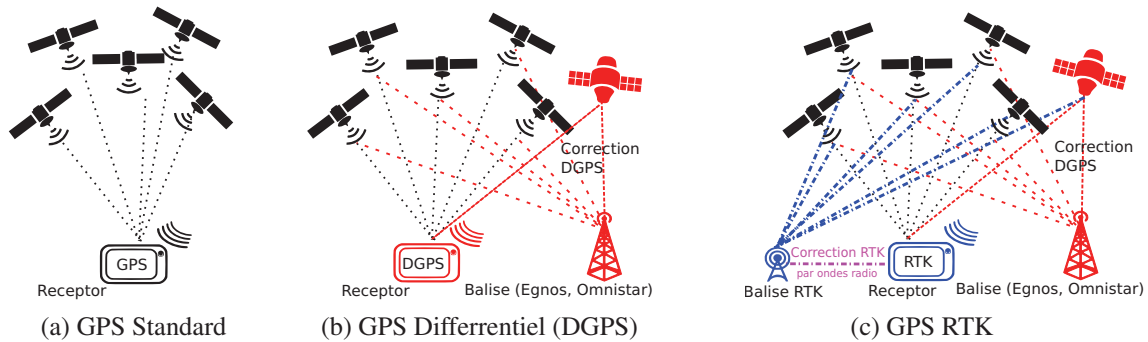


FIGURE III.2 – Les techniques de localisation GPS

Un des inconvénients de ces systèmes est leur niveau de précision qui varie entre 1 et 30 mètres en fonction du niveau de couverture (nombre de satellites), de l'environnement (zones dégagée, urbaines, etc.), de la qualité du récepteur et également du prix du service. Dans le domaine automobile, une fusion entre le GPS et l'odométrie du véhicule est habituellement utilisée pour une amélioration de la position [Bonnifait et al., 2001, 2003] et pour pallier le problème de non-détection du signal sur certains secteurs de route ou dans les tunnels. Afin d'améliorer la précision de positionnement, il existe systèmes comme le DGPS⁸ [Specht, 2015] ou le GPS-RTK⁹ [Odolinski et al., 2015] qui utilisent des relais fixes ou mobiles qui transmettent des corrections (voir figures III.2(b) et (c)). Le tableau III.1 fait un comparatif sur la précision de localisation en fonction du choix de la technique de correction (DGPS bas/haut de gamme ou GPS-RTK) mais également sur le prix des services.

Type de connexion	GPS	DGPS EGNOS	DGPS payante Bas de gamme	DGPS payante haut de gamme	RTK
Frais Annuel	Aucun	Aucun	≈1 200€	≈1 600€	Entretien de l'antenne
Investissement	dès 1€	dès 1€	dès 5€	dès 25€	> 5 000 €
Précision	± [1, 30] m	± [15, 20] cm	±[8, 12] cm	±[6, 10] cm	±[1, 10] cm

TABLE III.1 – Frais annuels et investissement pour différentes technologies GPS

Même s'il ne fait pas l'objet de notre champ d'application, il est à préciser que, en plus du calcul de position, les systèmes de navigation par satellite sont également utilisés dans le domaine

1. Global Positioning System;
2. GLObal NAVigation Satellite System - système russe opérationnel par tout dans le monde;
3. Système européen non fonctionnel à cause du mauvais placement sur l'orbite de deux sur les huit satellites envoyés à ce jour;
4. Navigation Satellite System - système chinois utilisable en Asie et au Pacifique de l'Ouest;
5. Système de navigation global chinois qui sera opérationnel avant 2020;
6. Quasi-Zenith Satellite System - système de navigation régional japonais qui couvre l'Asie et l'Océanie;
7. Indian Regional Navigation Satellite System - système de navigation régional mise en place par l'Inde;
8. Differential GPS;
9. GPS-Real Time Kinematic;

de la météorologie [Bevis et al., 1992] ou dans la géodésie [Davis et al., 1985].

Dans l'application de voiturier autonome, nous utilisons un GPS classique bas coût qui permet une estimation métrique globale de la position. Nous souhaitons savoir uniquement que le véhicule s'approche du point de départ et que le système de voiturier pourrait potentiellement être mis en route. Les détails sur la manière d'utilisation des techniques de localisation seront exposés dans la section §III.4. Pour la vérité terrain des jeux de données enregistrés utilisant un GPS-RTK et un *Correxit*¹ sont utilisés pour ensuite comparer les résultats des algorithmes de localisation.

III.2.1.2 Système d'Information Géographique (SIG)

L'acronyme SIG est couramment associé au Système d'Information Géographique qui est un «ensemble d'outils destinés à collecter, archiver et analyser des données géoréférencées» [Le Berre and Tissot, 1996]. Dans la section §II.2.2.2 nous avons montré que les tendances sont à la construction des cartes très riches en information du réseau routier et envisagé de les utiliser pour remonter à la localisation absolue du véhicule [Aronoff, 1989; Foresman, 1998; Jensfelt and Kristensen, 2001; Jakubiec and Reinhart, 2013; Pornon, 2015]. Dans les travaux de F. Malartre nous trouvons une comparaison des avantages et des inconvénients de ces bases de connaissances dans la localisation [Malartre, 2011].

Dans la section §III.2.1.1 nous avons décrit le fonctionnement du DGPS / GPS-RTK et nous avons montré que l'ajout d'une balise fixe augmente la précision de localisation. Sans utiliser forcément un système satellite, mais uniquement des balises radio terrestres, il est possible de réaliser une estimation de la position utilisant la triangulation [Esteves et al., 2003; Tekdas and Isler, 2010]. Sur le même principe, les émetteurs Wi-Fi peuvent être utilisés comme balises pour localiser un récepteur à l'intérieur ou à l'extérieur des bâtiments [Osa et al., 2013; Garcia-Valverde et al., 2013]. D'énormes bases de données SIG avec des bornes Wi-Fi géolocalisées ont été créées. Par exemple la base en libre accès Wigle annonce à ce jour de contenir 254 millions des bornes Wi-Fi et 6,3 millions de tours d'émission de réseau téléphonique [Wigle, 2016]. La géolocalisation de ceux-ci s'est réalisée utilisant plus de 3 552 millions d'observations acquises avec nos smartphones. Aujourd'hui, chaque fois que nos téléphones mobiles perçoivent du signal réseau ou Wi-Fi (sans qu'il soit nécessaire de s'y connecter), ils sont capables de se localiser de manière absolue. La construction de ce type de bases de données utilisant nos smartphones sans accord préalable pose un problème de légitimité et d'éthique mais ce sont des sujets que nous n'aborderons pas dans ce mémoire.

Une localisation précise utilisant le SIG n'est pas une fonctionnalité acquise. Même si les avancées en matière de précision et de richesse des bases de données sont remarquables, à ce jour il y a encore beaucoup de travail à réaliser pour assurer la fiabilité des données stockées. Par conséquent, même si les SIG présentent un grand avantage de localisation globale, les changements dans l'environnement non traduits dans les cartes rendent difficile leur utilisation dans toute situation.

Il se trouve que les moyens de localisation absolue ne sont pas disponibles dans toutes les situations ou que la précision fournie ne soit pas suffisante pour certaines applications. Pour notre système de voiturier autonome, l'utilisation des SIG n'est pas une voie d'exploration car nous savons déjà que l'environnement où le système est susceptible d'évoluer ne sera probablement pas cartographié. La manière dont les données GPS seront utilisées sera détaillée dans la section §III.4.

1. Capteur optique pour la mesure à haute fréquence et à haute précision de la distance, de la vitesse et l'angle. Il est très souvent utilisé dans les applications sur la dynamique du véhicule ;

III.2.2 Techniques de localisation relative

En ce qui concerne la localisation relative, comme le nom l'indique, le but est de calculer de combien s'est déplacé l'observateur (dans notre cas le véhicule) relativement à une position locale de référence. Ainsi nous réalisons la classification suivante :

- **localisation relative sans utilisation des a priori** : odométrie, odométrie visuelle ou localisation inertielle ;
- **localisation relative avec utilisation des a priori** :
 - localisation avec prise en compte de l'infrastructure ;
 - localisation sans prise en compte de l'infrastructure.
- **localisation et cartographie simultanée (SLAM)** : cette technique peut être considérée à la fois comme localisation relative avec ou sans a priori. Nous préférons de ne pas entrer dans ce débat et traiter l'algorithme de SLAM séparément dans la sous-section III.2.2.3.

III.2.2.1 Localisation relative sans utilisation des a priori

Comme le nom l'indique, l'estimation de la position se réalise par rapport à un repère fixe et sans utiliser des a priori sur l'environnement, comme les cartes. Nous détaillons d'abord les techniques pour ensuite conclure sur les avantages et les limitations de ces méthodes (notamment en termes de dérive). Dans ce cadre, ce type de localisation utilise l'odométrie avec ses déclinaisons : odométrie véhicule, odométrie visuelle ou localisation inertielle.

Localisation par odométrie

Une technique de localisation relative est l'odométrie. Dans la section §II.3.3 nous avons détaillé les encodeurs rotatifs qui équipent le véhicule. Utilisant un seul encodeur il est possible de calculer uniquement une distance parcourue. Par contre, deux encodeurs sont suffisants pour calculer une trajectoire [Chong and Kleeman, 1997; Maaref et al., 1999]. Comme le système ESP du véhicule utilise les informations issues des 4 encodeurs incrémentaux et de l'encodeur absolu fixé sur la colonne de direction, nous les utilisons pour tirer le maximum de profit. Les détails de cette localisation que nous employons également sont donnés dans la section §III.3.1.

Localisation inertielle

La localisation inertielle est une autre méthode de localisation relative. Utilisant des accéléromètres pour mesurer l'accélération et des gyromètres pour les variations de rotations, il est possible de remonter à la trajectoire parcourue [Barshan and Durrant-Whyte, 1995; Grewal et al., 2007; Woodman, 2007]. Ces capteurs sont généralement utilisés conjointement avec les odomètres pour la détection du glissement ou de perte de trajectoire [Lucet et al., 2015; Lucet and Micaelli, 2015].

Localisation par odométrie visuelle

Dans la littérature nous trouvons des techniques appelées *odométrie visuelle* (figure III.3) qui réalisent une estimation du déplacement relatif utilisant des images acquises en continu à partir d'une ou plusieurs caméras [Nistér et al., 2004, 2006; Maimone et al., 2007; Howard, 2008; Konolige et al.,

2010; Scaramuzza and Fraundorfer, 2011; Zhang et al., 2014]. L'idée de base est de détecter les similitudes entre plusieurs images successives et de déterminer les déplacements de la caméra qui permettent de passer d'une image à l'autre. Généralement l'approche est divisée en trois phases : la détection des primitives (ou points d'intérêt), l'appariement des primitives entre deux images et l'estimation de la transformation entre images [Mouaddib et al., 2007].

Il existe également des techniques d'odométrie par LiDAR que nous retrouvons dans la littérature en anglais sous le nom de *visual-LiDAR odometry* [Zhang and Singh, 2015] ou de *LiDAR odometry* [Fallon et al., 2012]. Par ailleurs, l'utilisation des caméras ToF¹ a donné naissance à des techniques d'odométrie visuelle qui utilisent à la fois l'appariement des points d'intérêt dans les images, mais également la profondeur de chaque pixel [Huang et al., 2011; Dong and Barfoot, 2014].

Appréciations sur la localisation relative sans utilisation d'a priori

L'avantage de ces techniques vient du fait qu'il n'y a pas besoin des informations a priori dans l'estimation du déplacement. En plus, ces techniques sont en principe pas très onéreuses en termes de temps de calcul. Par contre, l'inconvénient de la localisation relative (inertielle, odométrique, visuelle) est que l'erreur de calcul s'accumule dans le temps et crée une dérive de trajectoire. Comme dans ce type de localisation il n'y a pas d'élément de référence à part la position initiale, il n'y a aucun moyen de constater la dérive. Des techniques d'optimisation locale par fenêtre glissante ou d'optimisation globale par ajustement de faisceaux (*Bundle Adjustment*) ont été proposées pour la réduction de l'erreur globale de localisation pour l'odométrie visuelle [Sünderhauf et al., 2006; Fraundorfer et al., 2010; Konolige et al., 2010]. De plus, la combinaison avec d'autres capteurs, tels que GPS, le LiDAR, ou même avec une IMU apporte des améliorations importantes, bien évidemment en fonction de la précision des capteurs [Mourikis and Roumeliotis, 2007; Tardif et al., 2010a,b; Jones and Soatto, 2011; Ben-Afia et al., 2014].

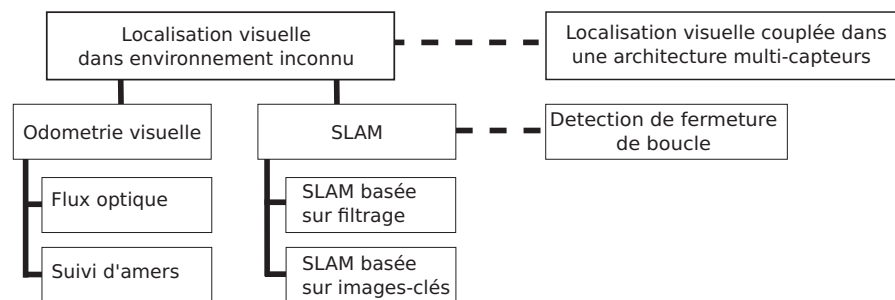


FIGURE III.3 – Classification des techniques de localisation visuelle

Dans la figure III.3 on peut observer en parallèle de l'odométrie visuelle, la localisation du type SLAM. Cette méthode, que nous détaillerons plus loin dans ce chapitre, suppose de localiser le véhicule en même temps que la création de la carte qui a vocation à être utilisée dans une étape ultérieure. Par la suite nous nous attardons sur les approches qui permettent de pallier le problème de dérive par l'utilisation des a priori.

III.2.2.2 Localisation relative avec utilisation a priori

Une autre manière courante de réaliser la localisation est d'utiliser des a priori. Dans la littérature deux types d'information a priori se distinguent : l'information cause-effet (*cause-effet*

1. Time of Flight;

relationships [Singhal, 1997]) et les cartes.

Utilisation des informations de type cause-effet

Concernant ce type de localisation, pour un jeu d'observations, la mise en relation cause-effet permet de préciser au robot où il se trouve. Pour donner un exemple, on affirme que le véhicule voit la Tour Eiffel. En sachant qu'il y a plusieurs répliques partout dans le monde, la position du véhicule est très incertaine. Si on ajoute que la tour a plus de 100 m, alors on restreint à cinq positions possibles : Paris (324 m), Las Vegas - États-Unis (165 m), Hangzhou - Chine (108 m), Shenshen-Chine (108 m), Ohio - États-Unis (101 m). En ajoutant l'information que la tour se trouve en France, alors on retrouve directement qu'il s'agit de l'original et sa position exacte est bien connue. Ce type de raisonnement cause-effet a donné naissance également à une famille de techniques probabilistes de localisation utilisant des cartes.

Utilisation des cartes

La localisation dans une carte est souvent considérée comme une localisation relative car la position est donnée dans le repère de la carte et celui-ci n'est pas toujours géolocalisé. Conformément à la classification présentée (voir figure III.1), un des critères de classification de la localisation dans une carte est fonction de la présence ou non d'infrastructure supplémentaire. Dans la littérature la division entre localisation absolue et locale peut différer quand il s'agit de l'utilisation des balises actives. La localisation par triangulation est un cas particulier de localisation dans une carte. Ce n'est pas la technique de triangulation qui transforme la localisation locale en une localisation absolue, mais c'est l'existence ou pas de l'information de géolocalisation liée à la balise. A titre d'exemple, dans les travaux de [Ait Aider, 2002] sur la localisation par balises actives, ou dans les travaux de [Talluri and Aggarwal, 1992] pour la localisation par repères passifs ou bien dans les travaux de [Ait Aider, 2002] sur la localisation à l'aide d'une carte du type plan CAD, nous retrouvons que toutes ces méthodes de localisation relèvent de la localisation absolue. Nous n'avons pas vocation à entrer dans ce débat mais, nous souhaitons uniquement donner les grandes lignes de recherche de la localisation utilisant des a priori.

Dans le chapitre II nous avons dédié la section §II.2.2 à la cartographie. Nous avons expliqué la nécessité d'utiliser les cartes pour combler le manque de performance et fiabilité des algorithmes d'interprétation de l'environnement en temps réel. Nous avons détaillé la problématique de la cartographie dans le domaine automobile et, enfin, nous avons donné les grandes lignes de la représentation des cartes avec les trois types de cartes : métriques, topologiques et mixtes. Pour la localisation dans une carte nous avons comme objectif d'assurer un suivi de position et, en dualité, de trouver la position globale dans la carte.

Concernant le suivi de position, les approches de localisation dans une carte basées sur le filtre de Kalman sont les plus fréquemment utilisées. Des variations de la technique ont été développées, comme celles basées sur le filtre de Kalman Étendu (EKF¹) [Ashokaraj et al., 2004]. Celui-ci peut être vu comme un estimateur bayésien itératif qui concentre l'ensemble des informations des états précédents dans son estimation courante. Par conséquent, il représente un moyen d'estimation à la fois de la position du véhicule, de son orientation. De plus, le choix et la quantité de données à fournir à cet estimateur peuvent être contrôlés. Toutes ces particularités font que le filtre de Kalman est très régulièrement employé dans les techniques de localisation mais également dans celles de cartographie dans le cas d'un SLAM² visuel [§III.3.2], par exemple.

1. Extended Kalman Filter; 2. Simultaneous Localisation And Mapping;

Si le suivi de position se limite à une estimation dans le voisinage de la position précédente, la localisation globale dans la carte dispose davantage de techniques d'optimisation globale. Dans les travaux de [Filliat, 2011] deux classes de techniques de localisation globale sont présentées : *inférence directe de position* et *le suivi de plusieurs hypothèses*. Les techniques utilisent les informations proprioceptives et extéroceptives pour trouver la meilleure localisation dans l'environnement, mais la première est limitée dans les environnements où il n'y a pas de *perceptual aliasing*. En échange, la deuxième technique explore en parallèle toutes les positions possibles. Dans ce sens, les travaux de localisation probabiliste hiérarchique de [Diard, 2003] sont cités ou également les travaux de localisation par approche *Top-Down* de perception multisensorielle développées par [Aynaud et al., 2014; Aynaud, 2015] et [Bernay-Angeletti et al., 2015]. Ces travaux ont à la base une modélisation probabiliste bayésienne [Fox et al., 1999; Bruyninckx, 2002; Burgard et al., 2011].

Une manière de réaliser la localisation globale dans la carte est issue de la technique d'appariement des cartes (*map matching*, en anglais) [Blanco et al., 2007; Hunter et al., 2014; Pinto et al., 2015; Furukawa et al., 2015]. Dans le chapitre IV nous donnerons les détails de la création d'une carte de traversabilité utilisant la perception par capteurs ultrason et caméra. La carte a été créée lors de l'étape d'apprentissage et sera appelée carte de référence. Un deuxième passage au même endroit permet de réaliser une deuxième carte. La mise en correspondance de ces cartes permet de réaliser une localisation par une technique appelée appariement des cartes (*map matching* en anglais). Nous utilisons également une stratégie de ce type dans notre approche et nous lui dédions la section §III.3.3.

Une autre technique de fusion des cartes ou également de localisation d'une carte dans une autre est par la technique I-SLSJF¹ qui utilise un filtre d'information et une optimisation globale par moindre carré [Huang et al., 2008; Zhao et al., 2013]. Sur le même principe de minimisation globale, la technique de fusion de cartes locales dans une carte globale est présentée par [Zhao et al., 2013] dans le Linear SLAM.

Si on prend le cas de la localisation par vision, dans un cadre idéal, si les coordonnées 3D des points de la carte sont parfaitement connues, un groupe de correspondances 2D/3D est alors nécessaire et suffisant pour estimer précisément la pose de la caméra. En pratique nous disposons rarement des cartes contenant ce type d'information. Pour éliminer ce problème, nous aimerions que le véhicule soit en mesure de construire sa propre carte. En effet, utiliser une carte fournie par des tiers est source d'erreurs. Le domaine de la robotique qui étudie la localisation et la cartographie est généralement appelé SLAM ou localisation et cartographie simultanées. Nous verrons par la suite que la construction de la carte par des moyens propres ouvre la voie vers l'utilisation d'une plus large gamme de techniques de localisation. Cela est possible car les informations nécessaires sont préalablement créées et stockées.

III.2.2.3 Localisation et cartographie simultanée - SLAM

Le SLAM n'est pas une technique en elle-même mais un principe. Les débuts de la localisation et cartographie se font dans les années 1986 avec les travaux de R. C. Smith et P. Cheeseman sur la représentation et l'estimation de l'incertitude spatiale durant la localisation [Smith and Cheeseman, 1986; Smith et al., 1987] ou dans le brevet déposé par [Kamejima et al., 1986] qui fait référence à la mise en correspondance d'une image perçue en temps réel avec une base d'images. L'acronyme de SLAM dans le sens de localisation et cartographie simultanée est employé pour la première fois par J. J. Leonard et H. F. Durrant-Whyte [Leonard and Durrant-Whyte, 1991a,b].

1. Information Sparse Local Submap Joining Filter;

Concernant les techniques de SLAM, nous retrouvons plusieurs critères de classification. Par exemple, certaines techniques sont temps réel, d'autres techniques ont besoin d'une étape appelée *post-traitement* (qui se réalise hors-ligne), généralement coûteuse en temps de calcul. D'autres classifications réduisent les approches de SLAM à deux autres classes : SLAM basé sur des techniques de filtrage et SLAM basé sur des techniques d'images clés.

SLAM basée sur filtrage

Le **KF-SLAM**¹ est la référence parmi les algorithmes de SLAM temps-réel. Ce SLAM utilise comme hypothèse que les transitions entre les états suivent une loi linéaire, que les transitions entre les mesures acquises suivent également une loi linéaire et de plus que les mesures sont exposées à un bruit blanc gaussien. Le plus souvent l'algorithme utilise l'information odométrique et des mesures issues des capteurs extéroceptifs pour accomplir sa tâche. Le succès de cette approche se voit à travers les multiples déclinaisons de l'algorithme dont les plus importantes sont : EKF-SLAM², CEKF-SLAM³, UKF-SLAM⁴ [Tuna et al., 2012; Kurt-Yavuz and Yavuz, 2012].

Le filtre de Kalman étendu (EKF) est un algorithme très répandu en estimation séquentielle. Celui-ci dépasse la contrainte de linéarité et réalise une linéarisation entre les transitions des états et de mesures via un développement de Taylor au premier ordre. Après ce processus de linéarisation, le KF classique est utilisé pour estimer les états. Le premier inconvénient est dû aux mauvais appariements qui font diverger le filtre. Le deuxième inconvénient de l'EKF-SLAM est l'augmentation du temps de calcul avec l'ajout de chaque nouvel amer dans le vecteur d'état. Si on considère N le nombre d'amers dans l'environnement, alors la complexité algorithmique de l'EKF-SLAM sera de $O(N^2)$ à chaque étape, avec une complexité totale de $O(N^3)$.

[Pérez, 2008] et [Paz et al., 2008] proposent des stratégies de couplage hiérarchique des cartes locales avec la mise à jour uniquement d'une seule carte pour réduire la complexité calculatoire globale à $O(N)$. Avec les mêmes objectifs, de réduire le temps de calcul et l'espace mémoire, des approches d'EKF-SLAM avec des cartes peu denses ont été développées et adaptées à l'environnement extérieur riche en obstacles [Julier, 2001; Guivant and Nebot, 2002; Civera et al., 2008, 2010; Féraud, 2011; Bresson et al., 2015].

Le CEKF-SLAM a été initialement proposé par J.E. Givrant et E.M. Nebot. Cet algorithme permet de réduire la complexité de calcul en divisant le vecteur d'état du système en deux parties appelées : partie active et partie passive. Seul le vecteur d'état des amers locaux est mis à jour à chaque étape par l'EKF. Les informations nécessaires à la mise à jour des autres états sont comprimées dans des structures matricielles appelées *matrices de coefficients auxiliaires* et mis à jour moins souvent. L'algorithme réalise des mises à jour globales uniquement quand le robot change entre deux cartes locales. Ainsi la complexité de calcul initialement requise de $O(N^2)$ pour la mise à jour à chaque étape devient $O(2L^2)$, où L représente le nombre d'états des amers locaux [Guivant and Nebot, 2001, 2002, 2003].

Une autre version de SLAM est l'UKF-SLAM, une technique basée sur la *Unscented Transformation (UT)* qui permet d'estimer plus facilement une fonction de probabilité qu'une fonction non linéaire. Par conséquent, l'UKF permet de réduire l'erreur d'estimation et le temps de calcul efficacement car l'UKF n'a pas besoin de dériver les matrices jacobiniennes [Huang et al., 2009].

1. **Kalman Filter-based SLAM**. Le filtre de Kalman est la variante la plus employée du filtre de Bayes; 2. **Extended Kalman Filter-based SLAM**; 3. **Compressed Extended Kalman Filter-based SLAM**; 4. **Unscented Kalman Filter-based SLAM**;

Le **SLAM à base de filtre à particules** est une autre solution de SLAM qui se base sur le principe de tester une multitude d'hypothèses par des tirages aléatoires du type Monte-Carlo sur un ensemble d'états. Partant de l'idée que la position des amers perçus est dépendante de la position du véhicule dans la carte, l'algorithme FastSLAM 1.0 [Montemerlo et al., 2002; Fox et al., 2003] a été créé. Le FastSLAM a divisé le SLAM en un problème de localisation et un problème d'estimation de la position des amers. Il utilise un échantillonnage de la trajectoire du véhicule et calcule les positions des amers conditionnellement à ces trajectoires. Il a été démontré que si N est le nombre d'amers dans l'environnement et M le nombre de particules. Alors chaque particule possède N filtres de Kalman indépendants qui estiment les positions des amers sous la contrainte de trajectoire. Dans ce cas l'espace mémoire de FastSLAM évolue sur une loi $O(M N)$ et le temps de calcul sur une loi $O(M \log N)$. Une deuxième version FastSLAM 2.0 a été proposée. L'innovation réside dans l'amélioration à la résistance au bruit et dans l'amélioration de la qualité des échantillons générés par la diminution du nombre des particules du filtre [Montemerlo and Thrun, 2007].

Il existe d'autres SLAM basés sur le filtre à particules comme : DP-Mapping¹ [Eliazar and Parr, 2003] ou également GMapping basé sur le filtre à particule Rao-Blackwell² [Grisetti et al., 2005, 2007].

Le **SLAM à base de filtre d'information**³ a été développé car ce type de filtre a l'avantage que plusieurs mesures sont prises en compte indépendamment les unes des autres. Ces mesures apportent leurs contributions au vecteur d'information et à la matrice d'information. Parmi ces informations se trouvent celles de localisation et des amers 3D. L'inconvénient majeur de ce type d'approche est la nécessité d'effectuer la conversion du vecteur d'information et de la matrice d'information à chaque étape. Cela demande d'inverser une matrice et implique donc une consommation des ressources supplémentaires [Thrun and Liu, 2005; Walter et al., 2007; Webster et al., 2013].

SLAM basée sur images-clés

Le problème de SLAM par filtrage consiste à construire de manière incrémentale une carte à partir d'une séquence d'images et d'utiliser cette carte pour localiser la caméra. Dans les méthodes de SLAM basé sur images-clés (*keyframes*), le nuage des points 3D et les poses de la caméra sont simultanément calculées par un processus d'optimisation nommé ajustement de faisceaux (BA : Bundle Adjustment en anglais) [Melbouci et al., 2015]. Le problème revient alors à trouver les poses des caméras C_i et les points 3D P_j qui minimisent les erreurs de réprojection $\min(f(C_1 \dots C_i, P_1 \dots P_i))$. Les deux techniques de minimisation les plus utilisées dans le BA sont : *Schur Complement* [Jeong et al., 2012; Wu, 2013; Carlone et al., 2014] ou Levenberg-Marquardt [Lourakis and Argyros, 2004, 2005; Agarwal et al., 2010; Dellaert, 2014].

Nous citons ici les travaux de M. Tamaazousti sur l'ajustement de faisceaux contraint comme cadre d'unification des méthodes de localisation [Tamaazousti, 2013] ou bien le C-KLAM⁴ [Nerurkar et al., 2014], le DCSLAM⁵ [Ramadasan et al., 2015] ou le MCSLAM⁶ [Ramadasan, 2015].

Détection de fermeture de boucle

Il a été constaté que pour toute technique de localisation et cartographie incrémentale, les erreurs d'estimation des positions des caméras et des points 3D s'accumulent et donnent lieu à des dérives (figure III.4(a)). Même après des processus d'optimisation globaux, les erreurs de dérive ne

1. **D**istributed **P**article; 2. Filtre basé sur le théorème Rao-Blackwell-Kolmogorov; 3. Appelé également la forme inverse du filtre de Kalman car il utilise une matrice d'information définie comme l'inverse de la matrice de covariance; 4. **C**onstrained **K**eyframe-based **L**ocalization **A**nd **M**apping; 5. **D**ynamically **C**onstrained real-time **S**LAM; 6. **M**ultiple **C**onstrained **S**LAM;

peuvent pas être réduites à zéro et un exemple de résultat est présenté dans la figure III.4(b). Il est facilement observable que les points sont mieux répartis au long de la trajectoire mais on observe que la boucle ne se ferme pas, même si en réalité elle devrait. Ces corrections de distorsions des cartes par fermeture de boucle sont possibles uniquement si les endroits qui se recouvrent sont connus. Le problème revient alors à détecter quelle est l'image au long de la trajectoire par où le véhicule est déjà passé qui correspond à une image donnée. La figure III.4(c) indique les résultats de fermeture de boucle présentés par [Williams et al., 2008] dans ses travaux.

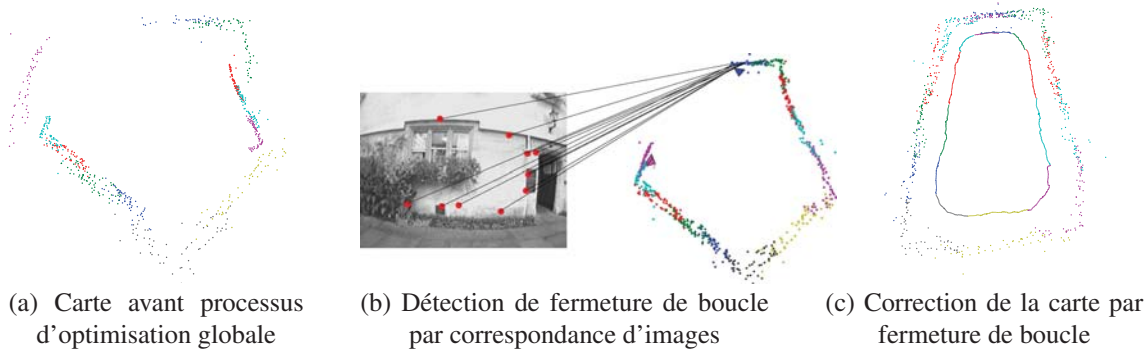


FIGURE III.4 – Détection de fermeture de boucle et correction de la carte au long d'une trajectoire approximativement de 70 m [Williams et al., 2008]

Le même principe de reconnaissance d'une image courante dans une base d'images de référence (à la suite d'une étape d'apprentissage) est également utilisé par [Blanc et al., 2004, 2005] dans la localisation d'un robot mobile à l'intérieur d'un bâtiment. La particularité de ce robot est qu'il se localise utilisant une caméra qui perçoit le plafond.

La même problématique de fermeture de boucle est retrouvée aussi dans les travaux de [Estrada et al., 2005] sur le Hierarchical-SLAM. Dans cette approche la carte globale est découpée dans des cartes locales connectées hiérarchiquement. Les appariements des cartes locales permettent de réaliser des correspondances et, par conséquent, de trouver le début et la fin d'une boucle. Nous citons deux autres approches développées toujours dans l'équipe de J. D. Tardós : l'ORB-SLAM [Mur-Artal et al., 2015] et le Real-time Monocular Object SLAM [Gálvez-López et al., 2016]. Ces deux dernières approches sont basées sur le principe de *Bag of Words* (BoW) ou de *sac de mots* en Français. Apparier ou retrouver la ressemblance entre deux images est une fonction coûteuse en temps de calcul. Apparier une image avec une base d'images est d'autant plus coûteux. L'idée de l'approche BoW est d'extraire des caractéristiques de chaque image et de les transformer dans une structure de données compacte appelée génériquement *Bag of Words*. Ensuite une indexation de ces données est réalisée pour faciliter une recherche ultérieure. Ainsi quand une nouvelle image recherche sa correspondance dans la base de données, le coût calculatoire sera amélioré car la recherche effective est faite à travers de petites structures de données indexées. Sur ce principe, nous pouvons noter les travaux de [Milford and Wyeth, 2012] sur l'OpenSeqSLAM ou bien les travaux de Mobile Robotics Group de Oxford University sur les techniques de FAB-MAP [Cummins and Newman, 2008; Paul and Newman, 2010; Cummins and Newman, 2011; Glover et al., 2012; Milford and Wyeth, 2012].

Nous rappelons le contexte : lors de la création d'une carte, l'imprécision de la technique de localisation fait que même si le véhicule revient au même endroit après avoir réalisé une trajectoire en forme de boucle, dans la trajectoire enregistrée il y a un écart. Pour pallier à ce problème il est nécessaire de détecter que le véhicule y est déjà passé. Connaissant cette information, il est possible alors de corriger la trajectoire par des techniques de *fermeture de boucle*.

Un deuxième problème lié à la reconnaissance des lieux est le *problème de la localisation initiale* ou *wake-up robot problem* en anglais (sujet étroitement lié au *kidnapped robot problem* [§III.2.2]).

Un troisième problème est la cartographie multisession. Ceci représente la tâche d'aligner deux cartes partielles de l'environnement recueillies par le robot au cours des différentes périodes de fonctionnement. La rotation, translation entre les deux cartes est inconnue et doit être déterminée automatiquement.

Ces trois problèmes sont des éléments communs et importants d'un système autonome qui doit être capable de fonctionner à long terme. Une solution à ces problèmes est d'utiliser une technique de localisation par comparaison d'images. Cette technique ne propose pas de calculer les poses de caméras en relatif par rapport aux autres images, mais simplement de quantifier que le contenu de la scène illustrée dans une image, ressemble à un autre suivant un critère de ressemblance. Si à chaque image on associe la pose de la caméra lors de la prise de vues utilisant par exemple une localisation relative par odométrie, alors retrouver deux images qui se ressemblent revient à mettre en correspondance deux poses, par conséquent de réaliser une technique de localisation. Les étapes génériques dans la composition d'une technique de localisation par dictionnaire de mots sont :

- **étape de construction du dictionnaire de mots-visuels** : cette étape est réalisée hors ligne car généralement très coûteuse en temps de calcul. Elle consiste à utiliser une base d'images contenant une grande variété de scènes pour extraire des caractéristiques uniques nommées *mots-visuels* qui seront triées et stockées dans une structure nommée dictionnaire ;
- **étape d'entraînement ou de cartographie d'une série d'images** avec ses deux sous étapes :
 - décomposition de chaque image dans des mots-visuels répertoriés dans le dictionnaire ;
 - création d'une structure indexée facilitant la recherche ultérieure ;
- **étape de test ou de localisation** : étant donné une image courante, il est nécessaire de réaliser un calcul de ressemblance entre les mots-visuels extraits de celle-ci avec ceux de la structure indexée créée dans l'étape de cartographie.

Nous reviendrons sur ses aspects dans la section §III.3.4 dédiée aux techniques de localisation par comparaison d'images.

III.3 Évaluation des techniques de localisation proposées

Dans la section précédente nous avons présenté l'état de l'art des techniques de localisation. Nous avons détaillé la localisation absolue et la localisation relative basée ou non sur des a priori, enfin la localisation et la cartographie simultanée. Dans cette section nous réalisons une évaluation de certaines techniques de localisation que nous considérons susceptibles d'être intégrées dans le système de voiturier autonome :

- localisation par odométrie [§III.3.1];
- localisation par techniques de SLAM [§III.3.2];
- localisation par appariement des grilles d'occupation [§III.3.3];
- localisation par techniques de comparaison d'images [§III.3.4].

III.3.1 Localisation par odométrie

Dans cette section nous étudions le mouvement du véhicule à travers l'information reçue des 4 capteurs odométriques et du capteur d'angle volant. Nous nous intéressons avant à la mécanique du véhicule car celle-ci conditionne le mouvement et donne du sens aux informations acquises. Nous faisons référence aux travaux de J. P. Brossard et dans les figures III.5(a) et (b) nous présentons la schématisation de la cinématique externe [Jean-Pierre Brossard, 2006].

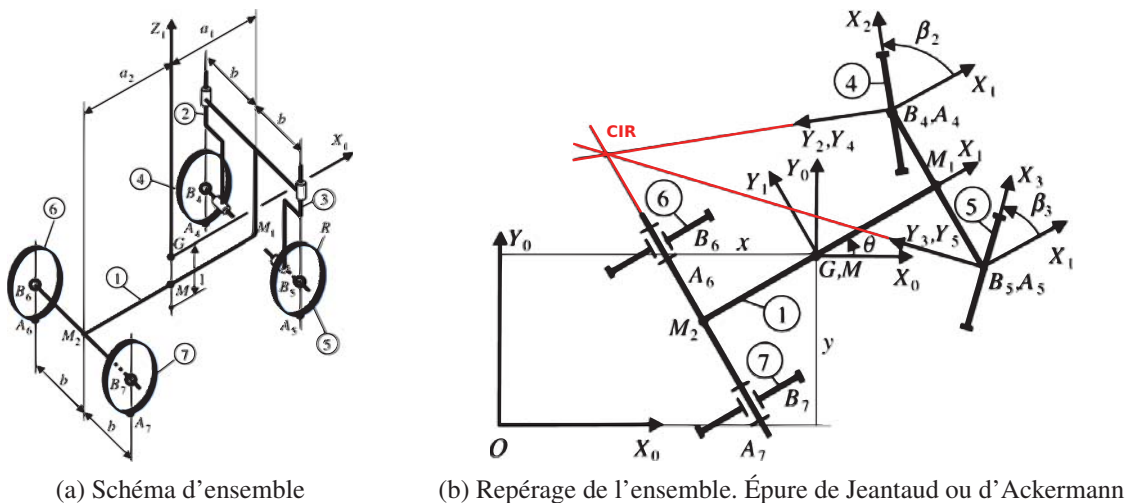


FIGURE III.5 – Schématisation pour la cinématique externe [Jean-Pierre Brossard, 2006]

La *cinématique externe* représente l'étude du mouvement global du véhicule. Par construction, le véhicule respecte l'épure de Jeantaud ou d'Ackermann : en virage, les axes des roues directrices se coupent au *Centre Instantané de Rotation (CIR)* situé sur le prolongement de l'essieu arrière. Pour la simplification, une hypothèse forte est prise : tous les corps en composition du véhicule sont rigides, y compris les roues. Dans les deux schémas des figures III.5(a) et (b), nous retrouvons 7 solides en composition du système, numérotés de (S_1) à (S_7) . Le véhicule est en mouvement sur une route horizontale. On note le solide (S_0) la route et \mathfrak{R}_0 (équivalent à \mathfrak{R}_m) le repère $(O, \vec{X}_0, \vec{Y}_0, \vec{Z}_0)$ associé, alors les points de contact entre le véhicule et la route sont :

$$A_i \leftarrow (S_0)/(S_i) \quad \text{avec } i \in \{4, 5, 6, 7\} \quad (\text{III.1})$$

Une autre hypothèse faite est le roulement sans glissement. Cela signifie vitesse nulle par rapport au sol du point de contact appartenant à la roue. Pour la roue S_4 l'équation s'écrit :

$$\vec{V}_4^0(A_4) = \vec{V}_4^0(B_4) + \vec{\Omega}_4^0 \wedge \overrightarrow{B_4A_4} = 0 \quad (\text{III.2})$$

Comme le point B_4 appartient aussi à (S_1) alors

$$\vec{V}_4^0(B_4) = \vec{V}_1^0(B_4) \quad (\text{III.3})$$

$$\vec{V}_1^0(B_4) = \vec{V}_1^0(G) + \vec{\Omega}_1^0 \wedge \overrightarrow{GB_4} \quad (\text{III.4})$$

$$\vec{V}_1^0(G) = [x', y', 0]_{\mathfrak{R}_0}^T \quad (\text{III.5})$$

$\vec{V}_1^0(G)$ exprimé dans le repère (\mathfrak{R}_1) devient :

$$\vec{V}_1^0(G) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 0 \end{bmatrix} = \begin{bmatrix} x' \cos \theta + y' \sin \theta \\ -x' \sin \theta + y' \cos \theta \\ 0 \end{bmatrix}_{\mathfrak{R}_1} = \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix}_{\mathfrak{R}_1} \quad (\text{III.6})$$

$$\begin{aligned} \vec{\Omega}_1^0 \wedge \overrightarrow{GB_4} &= \theta' \vec{Z}_1 \wedge \overrightarrow{GB_4} = \theta' \vec{Z}_1 \wedge (\overrightarrow{GM} + \overrightarrow{MB_4}) = \\ &= \theta' \vec{Z}_1 \wedge (-(l - R_{roue}) \vec{Z}_1 + a_1 \vec{X}_1 + b \vec{Y}_1) = \\ &= \begin{bmatrix} 0 \\ 0 \\ \theta' \end{bmatrix}_{\mathfrak{R}_1} \wedge \begin{bmatrix} a_1 \\ b \\ -(l - R_{roue}) \end{bmatrix}_{\mathfrak{R}_1} = \begin{bmatrix} -b\theta' \\ a_1\theta' \\ 0 \end{bmatrix}_{\mathfrak{R}_1} \end{aligned} \quad (\text{III.7})$$

La vitesse du centre de la roue est donc

$$\vec{V}_4^0(B_4) = \begin{bmatrix} x' \cos \theta + y' \sin \theta - b\theta' \\ -x' \sin \theta + y' \cos \theta + a_1\theta' \\ 0 \end{bmatrix}_{\mathfrak{R}_1} \quad (\text{III.8})$$

En revenant à l'équation (III.2), on retrouve :

$$\vec{V}_4^0(A_4) = \vec{V}_4^0(B_4) + \vec{\Omega}_4^0 \wedge \overrightarrow{B_4A_4} \quad (\text{III.9})$$

$$\begin{aligned} \text{ou } \vec{\Omega}_4^0 \wedge \overrightarrow{B_4A_4} &= (\vec{\Omega}_4^2 + \vec{\Omega}_2^1 + \vec{\Omega}_1^0) \wedge \overrightarrow{B_4A_4} = (\phi'_4 \vec{Y}_4 + \beta'_2 \vec{Z}_2 + \theta' \vec{Z}_2) \wedge \overrightarrow{B_4A_4} = \\ &= \begin{bmatrix} 0 \\ \phi'_4 \\ \beta'_2 + \theta' \end{bmatrix}_{\mathfrak{R}_2} \wedge \begin{bmatrix} 0 \\ 0 \\ -R_{roue} \end{bmatrix}_{\mathfrak{R}_2} = \begin{bmatrix} -R_{roue} \phi'_4 \\ 0 \\ 0 \end{bmatrix}_{\mathfrak{R}_2} = \\ &= \begin{bmatrix} \cos \beta_2 & \sin \beta_2 & 0 \\ -\sin \beta_2 & \cos \beta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -R_{roue} \phi'_4 \\ 0 \\ 0 \end{bmatrix}_{\mathfrak{R}_1} = \\ &= \begin{bmatrix} -R_{roue} \phi'_4 \cos \beta_2 \\ -R_{roue} \phi'_4 \sin \beta_2 \\ 0 \end{bmatrix}_{\mathfrak{R}_1} \end{aligned} \quad (\text{III.10})$$

où ϕ'_4 représente la vitesse de rotation de la roue avant gauche. On retrouve donc la vitesse du

point A_4 pour un observateur lié à \mathfrak{R}_0 :

$$\vec{V}_4^0(B_4) = \begin{bmatrix} x' \cos \theta + y' \sin \theta - b\theta' - R_{roue}\phi'_4 \cos \beta_2 \\ -x' \sin \theta + y' \cos \theta + a_1\theta' - R_{roue}\phi'_4 \sin \beta_2 \\ 0 \end{bmatrix}_{\mathfrak{R}_1} \quad (\text{III.11})$$

Par conséquent, les relations de roulement sans glissement pour la roue (S_4) sont :

$$(S_4) : \begin{cases} x' \cos \theta + y' \sin \theta - b\theta' - R_{roue}\phi'_4 \cos \beta_2 = 0 \\ -x' \sin \theta + y' \cos \theta + a_1\theta' - R_{roue}\phi'_4 \sin \beta_2 = 0 \end{cases} \quad (\text{III.12})$$

En répétant le raisonnement pour les autres trois roues (S_5), (S_6) et (S_7), on retrouve :

$$(S_5) : \begin{cases} x' \cos \theta + y' \sin \theta + b\theta' - R_{roue}\phi'_5 \cos \beta_3 = 0 \\ -x' \sin \theta + y' \cos \theta + a_1\theta' - R_{roue}\phi'_5 \sin \beta_3 = 0 \end{cases} \quad (\text{III.13})$$

$$(S_6) : \begin{cases} x' \cos \theta + y' \sin \theta - b\theta' - R_{roue}\phi'_6 = 0 \\ -x' \sin \theta + y' \cos \theta - a_2\theta' = 0 \end{cases} \quad (\text{III.14})$$

$$(S_7) : \begin{cases} x' \cos \theta + y' \sin \theta + b\theta' - R_{roue}\phi'_7 = 0 \\ -x' \sin \theta + y' \cos \theta - a_2\theta' = 0 \end{cases} \quad (\text{III.15})$$

Partant des équations (III.12) à (III.15), on exprime les vitesses de rotation des roues en fonction de la vitesse de rotation θ' . Si le véhicule est uniquement en translation, alors $\theta' = 0$ et $\phi'_4 = \phi'_5 = \phi'_6 = \phi'_7 = 0$

$$\text{sinon,} \quad \begin{cases} \phi'_4 = \frac{a_1 + a_2}{R_{roue}} \frac{\theta'}{\sin \beta_2} = \frac{a_1 + a_2}{R_{roue}} \frac{1}{(a_1 + a_2) \cos \beta_2 + b \sin \beta_2} V_x \\ \phi'_5 = \frac{a_1 + a_2}{R_{roue}} \frac{\theta'}{\sin \beta_3} = \frac{a_1 + a_2}{R_{roue}} \frac{1}{(a_1 + a_2) \cos \beta_2 - b \sin \beta_3} V_x \\ \phi'_6 = \frac{a_1 + a_2}{R_{roue}} \coth \beta_2 \phi' = \frac{a_1 + a_2}{R_{roue}} \frac{\coth \beta_2}{(a_1 + a_2) \coth \beta_2 + b} V_x \\ \phi'_7 = \frac{a_1 + a_2}{R_{roue}} \coth \beta_3 \phi' = \frac{a_1 + a_2}{R_{roue}} \frac{\coth \beta_3}{(a_1 + a_2) \coth \beta_3 - b} V_x \end{cases} \quad (\text{III.16})$$

Ainsi nous retrouvons la relation entre β_2 et β_3 :

$$\begin{cases} \text{Si } \theta' = 0 \text{ alors } & \beta_2 = \beta_3 = 0 \\ \text{Sinon} & \coth \beta_3 - \coth \beta_2 = \frac{2b}{a_1 + a_2} = \frac{2b}{l} \end{cases} \quad (\text{III.17})$$

Dans ses travaux, J.P. Brossard présente également la démonstration que les axes des roues avant se coupent sur l'axe de l'essieu arrière dans un point nommé *centre instantané de rotation* (CIR) [Jean-Pierre Brossard, 2006]. Les coordonnées de ce point dans le repère \mathfrak{R}_1 et \mathfrak{R}_0 sont :

$$(X_{CIR}, Y_{CIR})_{\mathfrak{R}_1} = (-a_2, \quad b + (a_1 + a_2) \coth \beta_2) \quad (\text{III.18})$$

$$(X_{CIR}, Y_{CIR})_{\mathfrak{R}_0} = \left(x - \frac{y'}{\theta'}, \quad y + \frac{x'}{\theta'} \right) \quad (\text{III.19})$$

Le rayon de virage R_G est donné par l'équation :

$$R_G = \|\vec{M}_{CIR}\| = \sqrt{X_{CIR}^2 + Y_{CIR}^2} = \sqrt{a_2^2 + [b + (a_1 + a_2) \coth \beta_2]^2} \quad (\text{III.20})$$

Pour notre application, il est important d'estimer le déplacement du véhicule grâce aux capteurs odométriques. Pour cela, il est possible d'utiliser la distance parcourue par chaque roue

mais également avec la vitesse de chaque roue. L'équation de la vitesse du centre d'une roue étant :

$$v = \frac{2\pi R_{roue}}{N} \cdot \delta_{tops} \cdot \frac{1}{\Delta t} \quad (\text{III.21})$$

avec $\frac{2\pi R_{roue}}{N}$ la résolution angulaire, δ_{tops} représentant le nombre de fronts montants détectés par l'encodeur que nous appelons *tops roue* ($N = 48$ tops par tour de roue pour notre encodeur) et Δt représentant l'intervalle de temps écoulé entre deux mesures. Certains capteurs fournissent directement le déplacement angulaire, d'autres fournissent la vitesse angulaire. Nous avons accès aux deux informations envoyées par le calculateur ESP. Nous choisissons de travailler directement en distance car l'utilisation de la vitesse, donc de la division de la distance au laps de temps, est une source d'erreur de calcul. Pour la simplicité des équations nous considérons dans le reste du mémoire (x, y, θ) la position et l'orientation du centre de l'essieu arrière véhicule dans un repère absolu \mathfrak{R}_0 . Cela revient à utiliser les équations proposées par J. P. Brossard dans le cas particulier $a_2 = 0, l = 0$. Nous nous intéressons à calculer successivement les positions atteintes par le véhicule au long de son déplacement, ce qui nous permettra de reconstruire sa trajectoire. A la suite nous présentons plusieurs algorithmes de calcul de trajectoire en utilisant l'information odométrique d'une seule roue directrice [§III.3.1.1], des deux roues arrière [§III.3.1.2] ou des quatre roues et de l'angle volant [§III.3.1.3]. Nous finalisons cette section par une comparaison des résultats de techniques de localisation par odométrie [§III.3.1.4].

III.3.1.1 Utilisation de l'information odométrique d'une seule roue directrice

Une méthode de calcul de la trajectoire est d'utiliser une seule roue directrice ainsi que son angle qui est déduit de l'angle volant. Prenons par exemple la roue avant gauche. Par le réseau CAN nous recevons systématiquement l'information de l'encodeur incrémental de la roue et l'information d'angle volant. Calculer la position suivante atteinte par le véhicule revient à calculer la nouvelle position du centre de l'essieu arrière du véhicule sachant que la roue avant gauche a réalisé un mouvement sous la contrainte d'Ackermann précédemment exposée (équation (III.16)).

Algorithme 1 Odométrie utilisant l'information d'une roue directrice

nécessite : $R_{roue}, a_1, a_2, b, (x_{in}, y_{in}, \theta_{in}), \delta_{topsAvG}, \beta_{topsAvG}, sens$

$(x, y, \theta) \leftarrow (x_{in}, y_{in}, \theta_{in})$

pour tout $i \in [1, \delta_{topsAvG}]$ **faire**

$$x \leftarrow x + sens \frac{2\pi R_{roue}}{N} [(a_1 + a_2) \cos \beta_{AvG} + b \sin \beta_{AvG}] \frac{\cos \theta}{(a_1 + a_2)}$$

$$y \leftarrow y + sens \frac{2\pi R_{roue}}{N} [(a_1 + a_2) \cos \beta_{AvG} + b \sin \beta_{AvG}] \frac{\sin \theta}{(a_1 + a_2)}$$

$$\theta \leftarrow \theta + sens \frac{2\pi R_{roue}}{N} \frac{\sin \beta_{AvG}}{(a_1 + a_2)}$$

fin pour

retourne $(x_{out}, y_{out}, \theta_{out}) \leftarrow (x, y, \theta)$

Notons $\delta_{topsAvG}$ un nombre entier représentant le nombre d'incrémentes mesurés par l'encodeur incrémental par unité de temps. Sachant que par tour de roue il y a N incrémentes (avec $N = 48$ dans notre cas), $\delta_{topsAvG}$ revient à mesurer une variation angulaire. Notons β_{AvG} l'angle fait par la roue avant gauche. Alors l'algorithme 1 nous permet de calculer la nouvelle position atteinte par le véhicule sous l'hypothèse que l'angle de la roue reste inchangé entre deux acquisitions. Dans nos algorithmes, le point de référence sera le milieu de l'essieu arrière.

III.3.1.2 Utilisation de l'information odométrique des roues arrière

Dans la littérature nous trouvons deux modèles odométriques qui n'utilisent que l'information issue des odomètres des roues arrière. Le premier suppose que la trajectoire du robot est un cercle de rayon R_{CIR} autour de CIR, alors la vitesse du centre de l'essieu arrière est égale à :

$$v = \frac{v_{ArrD} + v_{ArrG}}{2} \quad \text{et} \quad R_{CIR} = b \frac{v_{ArrD} + v_{ArrG}}{v_{ArrD} - v_{ArrG}} \quad (\text{III.22})$$

Le changement d'orientation ω du robot et les coordonnées du CIR sont donnés par :

$$\omega = d\theta = \frac{d}{R_{CIR}} = \frac{v}{dt} \cdot \frac{1}{R_{CIR}} \quad (\text{III.23})$$

$$\begin{cases} x_{CIR} = x - R_{CIR} \cos \theta \\ y_{CIR} = y - R_{CIR} \sin \theta \end{cases} \quad (\text{III.24})$$

Ce qui conduit à trouver la nouvelle position du robot ainsi :

$$\begin{cases} \theta \leftarrow \theta + \omega \\ x \leftarrow x_{CIR} + R_{CIR} \cos \theta \\ y \leftarrow y_{CIR} + R_{CIR} \sin \theta \end{cases} \quad (\text{III.25})$$

Toujours en utilisant l'information d'uniquement deux roues arrière, il est possible de faire abstraction des contraintes de cercle de rotation. Par conséquent il n'est pas nécessaire de calculer le CIR, mais uniquement du déplacement élémentaire Δ et de la rotation élémentaire ω .

$$\Delta = \frac{\Delta_{ArrD} + \Delta_{ArrG}}{2} \quad \text{et} \quad \omega = \frac{\Delta_{ArrD} - \Delta_{ArrG}}{2b} \quad (\text{III.26})$$

La position du robot sera mise à jour comme suit :

$$\begin{cases} \theta \leftarrow \theta + \omega \\ x \leftarrow x + \Delta \cos \theta \\ y \leftarrow y + \Delta \sin \theta \end{cases} \quad (\text{III.27})$$

Nous verrons dans le bilan [§III.3.1.4] que, de par leur simplicité, les équations (III.26) et (III.27) transposées dans l'algorithme 2 mèneront aux meilleurs résultats de calcul de trajectoire.

Algorithme 2 Odométrie : Utilisation uniquement de l'information des roues arrière

nécessite : R_{roue} , b , $(x_{in}, y_{in}, \theta_{in})$, $\delta_{topsArrG}$, $\delta_{topsArrD}$, $sens$

$$\Delta \leftarrow \frac{2\pi R_{roue} \delta_{topsArrD} + \delta_{topsArrG}}{N} \quad // \text{Déplacement élémentaire}$$

$$\omega \leftarrow \frac{2\pi R_{roue} \delta_{topsArrD} - \delta_{topsArrG}}{2b} \quad // \text{Rotation élémentaire}$$

$$\theta_{out} \leftarrow \theta_{in} + \omega$$

$$x_{out} \leftarrow x_{in} + \Delta \cos(\theta_{out} + \omega)$$

$$y_{out} \leftarrow y_{in} + \Delta \sin(\theta_{out} + \omega)$$

retourne $(x_{out}, y_{out}, \theta_{out})$

III.3.1.3 Utilisation de l'information odométrique des quatre roues et de l'encodeur du volant

Une dernière approche de calcul de la trajectoire que nous présentons est issue des travaux de P. Bonnifait sur la fusion de données odométriques avec des données GPS [Bonnifait et al., 2001].

Ils partent du principe que la redondance des mesures des quatre roues permettra de calculer une trajectoire d'une meilleure précision que celle issue du modèle différentiel des roues arrière. Cela permet aussi d'évaluer s'il y a du glissement et donc d'évaluer la dérive de localisation.

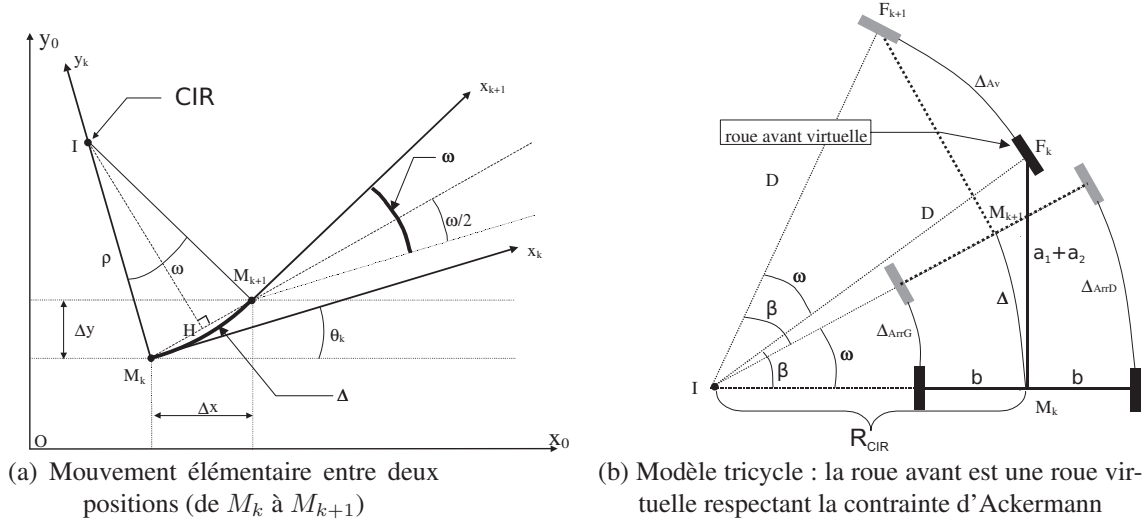


FIGURE III.6 – Calcul de trajectoire utilisant les informations des quatre roues du véhicule et de l'angle volant [Bonnifait et al., 2001]

Algorithme 3 Odométrie : Utilisation de l'information des 4 roues plus l'angle volant. Estimation du déplacement utilisant un filtre de Kalman étendu

nécessite : R_{roue} , l , b , $(x_{in}, y_{in}, \theta_{in})$, $\delta_{topsArrG}$, $\delta_{topsArrD}$, $\delta_{topsAvG}$, $\delta_{topsAvD}$, β_{moy} , $sens$
nécessite : $\mathbf{Q}_{5 \times 5}$ (bruit de l'entrée), $\mathbf{P}_{5 \times 5}$ (matrice d'incertitude associée au vecteur d'état), $\mathbf{R}_{\mathbf{O}_{5 \times 5}}$ (matrice de bruit de mesure)

$$\vec{X}_k \leftarrow \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \Delta_k \\ \omega_k \end{bmatrix} \quad \text{et} \quad \vec{Z} \leftarrow \begin{bmatrix} \tan(\beta_{moy}) \\ sens \frac{2\pi R_{roue}}{N} \delta_{topsArrG} \\ sens \frac{2\pi R_{roue}}{N} \delta_{topsArrD} \\ sens \frac{2\pi R_{roue}}{N} \delta_{topsAvG} \cos(\beta_{AvG}) \\ sens \frac{2\pi R_{roue}}{N} \delta_{topsAvD} \cos(\beta_{AvD}) \end{bmatrix} \quad // \text{Vecteurs d'état de mesure}$$

$$\hat{X}_{k/k-1} \leftarrow f(\hat{X}_{k-1/k-1}) \dots \dots \dots // \text{Phase de prédiction}$$

$$\mathbf{P}_{k/k-1} \leftarrow \mathbf{P}_{k-1/k-1} + \mathbf{Q}_k$$

$$\tilde{Y}_k \leftarrow Z_k - h(\hat{X}_{k/k-1}) \dots \dots \dots // \text{Phase de mise à jour}$$

$$\mathbf{S}_k \leftarrow \mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^T + \mathbf{R}_{\mathbf{O}_k}$$

$$\mathbf{G}_k \leftarrow \mathbf{P}_{k/k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$$

$$\hat{X}_{k/k} \leftarrow \hat{X}_{k/k-1} + \mathbf{G}_k \tilde{Y}_k$$

$$\mathbf{P}_{k/k} \leftarrow (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k) \mathbf{P}_{k/k-1}$$

retourne (x_k, y_k, θ_k)

En faisant l'hypothèse que $\Delta \approx \|M_k M_{k+1}\|$, alors le modèle d'évolution s'écrit :

$$\begin{cases} x_{k+1} = x_k + \Delta_k \cos(\theta_k + \frac{\omega_k}{2}) \\ y_{k+1} = y_k + \Delta_k \sin(\theta_k + \frac{\omega_k}{2}) \\ \theta_{k+1} = \theta_k + \omega_k \\ \Delta_{k+1} = \Delta_k + \epsilon_\Delta \\ \omega_{k+1} = \omega_k + \epsilon_\omega \end{cases} \quad (III.28)$$

où les valeurs de Δ_k et ω_k représentent le déplacement respectivement la rotation élémentaires du centre de l'essieu arrière. L'algorithme 3 basé sur le filtre de Kalman étendu est proposé pour réaliser l'estimation du vecteur d'état $\vec{X}_k = [x_k, y_k, \theta_k, \Delta_k, \omega_k]^T$. Ce filtre EKF peut être interprété comme un estimateur de moindres carrés pondérés (Weighted Least Squares Estimator, en anglais). Du bruit est ajouté sur les paramètres Δ_k et ω_k afin qu'ils puissent évoluer, sinon leurs valeurs seront considérées étant sûres. Le modèle est donné par l'équation (III.28) peut être réécrit sous la forme d'une fonction d'évolution et une fonction de mesure du type :

$$\vec{X}_k = f\left(\vec{X}_{k-1}, \vec{e}_{\vec{X},k}\right) \quad (\text{III.29})$$

$$\vec{Z}_k = h\left(\vec{X}_k, \vec{e}_{\vec{Z},k}\right) \quad (\text{III.30})$$

avec $\vec{e}_{\vec{X},k}$ et $\vec{e}_{\vec{Z},k}$ représentant les bruits de processus respectivement de mesures à l'instant k . Il est possible d'écrire un système d'équations redondantes et non linéaires qui relie les inconnues du modèle d'évolution des variables mesurées ($\beta, \delta_{ArrG}, \delta_{ArrD}, \delta_{AvG}, \delta_{AvD}$) ainsi :

$$\begin{cases} \tan(\beta) & = (a_1 + a_2) \frac{\omega}{\Delta} \\ \Delta_{ArrG} & = \Delta - b\omega \\ \Delta_{ArrD} & = \Delta + b\omega \\ \Delta_{AvG} \cos(\beta_{AvG}) & = \Delta - b\omega \\ \Delta_{AvD} \cos(\beta_{AvD}) & = \Delta + b\omega \end{cases} \quad (\text{III.31})$$

Dans l'algorithme 3 $\vec{Z}_k = [\tan(\beta_k), \Delta_{ArrG,k}, \Delta_{ArrD,k}, \Delta_{AvG,k}, \beta_{AvG,k}, \Delta_{AvD,k}, \beta_{AvD,k}]^T$ est utilisé comme vecteur des mesures bruitées à l'instant k .

III.3.1.4 Comparaison des résultats de techniques de localisation par odométrie

Nous nous intéressons à évaluer la précision des différentes techniques précédemment présentées. Pour cela nous disposons des jeux de données acquises avec le véhicule équipé des capteurs de réalité terrain de type *Correvit*¹ et GPS-RTK. Les trajectoires dont nous disposons, qui présentent des longueurs assez courtes (20 à 30 m) ont été acquises par H. Vorobieva. Son travail a été réalisé dans le même département d'*ADAS et Véhicule Autonome* de Renault et visait à étudier et à automatiser les manœuvres du véhicule lors d'une situation de parking latéral de type créneau [Vorobieva, 2014].

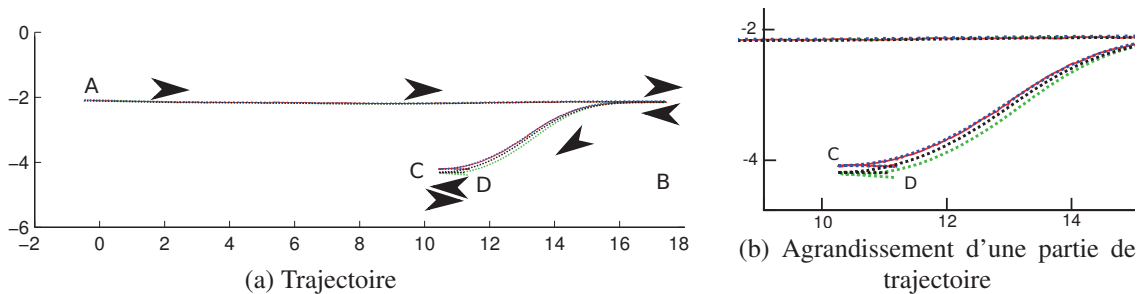
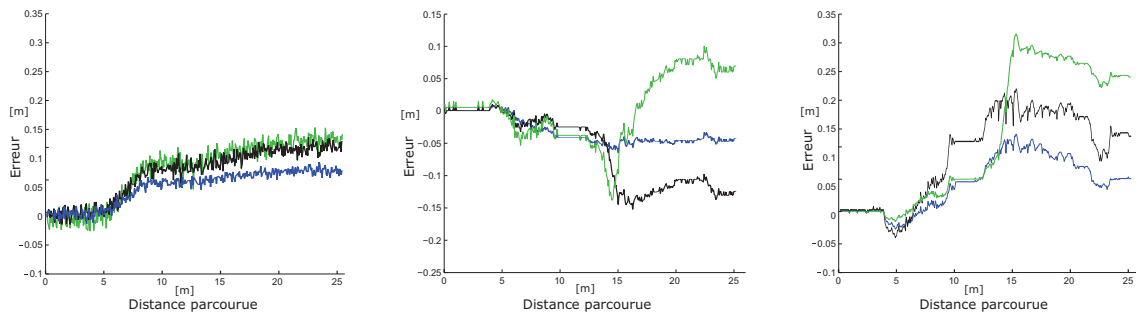


FIGURE III.7 – Trajectoire de type créneau : en rouge trajectoire GPS-RTK, en noir trajectoire estimée en utilisant l'information odométrique d'une seule roue directrice [§III.3.1.1], en bleu des deux roues arrière [§III.3.1.2] et en vert des quatre roues et de l'angle volant [§III.3.1.3]

1. Capteur optique pour la mesure de la vitesse et du déplacement relatif;

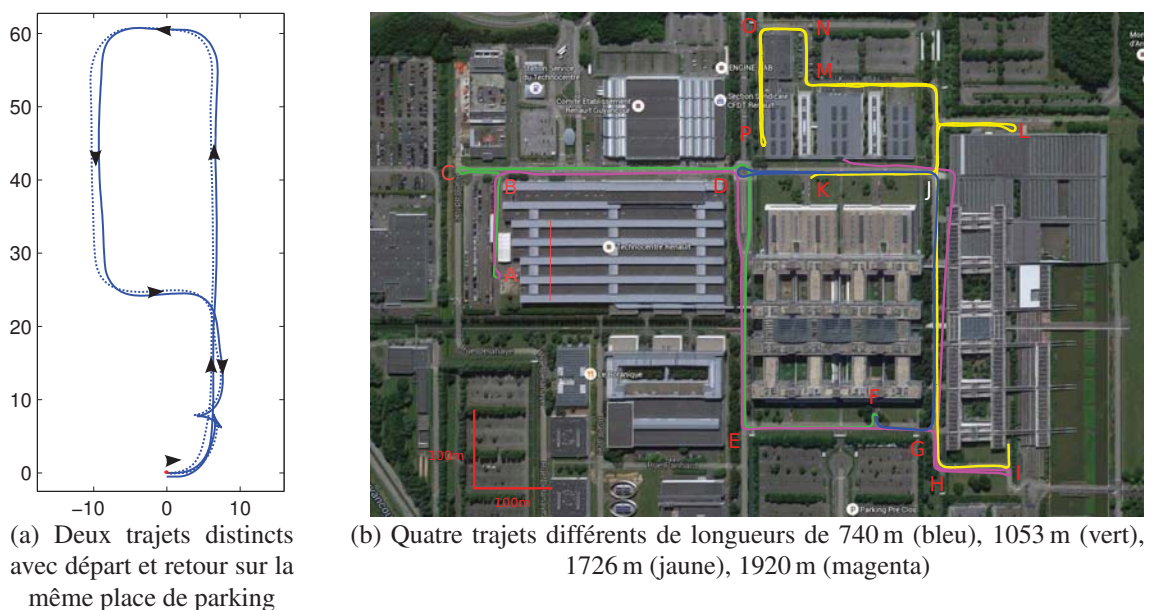
Les figures III.7(a) et (b) illustrent les trajectoires mesurées par GPS-RTK (en rouge) ou estimées par les algorithmes testés. Les figures III.8(a) à (c) indiquent les erreurs sur l'estimation de distance (par rapport au Correvit) et les erreurs de pose selon l'axe des abscisses (erreur longitudinales) et des données (erreurs latérales) dans un repère fixé en début de la trajectoire.



(a) Erreur en distance parcourue (b) Erreur latérale (c) Erreur longitudinale

FIGURE III.8 – Estimation des performances des trois algorithmes : en noir résultats de l’algorithme utilisant l’information odométrique d’une seule roue directrice [§III.3.1.1], en bleu des deux roues arrière [§III.3.1.2] et en vert des quatre roues et de l’angle volant [§III.3.1.3]

Les résultats issus des essais réalisés sur plusieurs trajectoires nous ont montré que les meilleures précisions sont obtenues grâce au modèle différentiel utilisant uniquement les deux roues arrière. Dans la section §II.3.3 du chapitre précédent, nous avons détaillé la mise en correspondance entre des données issues de l’encodeur du volant (fixé sur la colonne de direction) et les angles effectifs des roues directrices mesurés sur un banc de tests (figure II.60). Ces erreurs de mesure des angles des roues en plus de l’hypothèse que le déplacement se réalise sans glissement (ce qui est une contrainte forte pour les roues motrices) et l’utilisation d’un modèle simplifié de la mécanique d’actionnement du braquage de la roue directrice (figure III.5), font que l’utilisation des informations issues des roues avant introduit trop d’erreurs. Par conséquent, ce ne sont que les informations des deux roues arrière qui seront utilisées par la suite.



(a) Deux trajets distincts avec départ et retour sur la même place de parking (b) Quatre trajets différents de longueurs de 740 m (bleu), 1053 m (vert), 1726 m (jaune), 1920 m (magenta)

FIGURE III.9 – Différents trajets réalisés par le véhicule prototype et calcul de trajectoire utilisant le modèle odométrique différentiel des deux roues arrière [§III.3.1.2]

Comme nous ne disposons pas de GPS-RTK pour réaliser des mesures précises pour des trajectoires plus longues, nous réalisons des trajectoires avec fermetures de boucle roulant à faible vitesse (1-2 m/s) et nous mesurons l'écart réel à la main (figure III.9(a)) ou bien nous superposons les résultats sur une carte Google (figure III.9(b)). L'estimation de la distance parcourue est très précise pour notre application. Ce qui nous intéresse pour ces longues trajectoires est la déviation due à une mauvaise estimation de l'angle. La figure III.9(a) présente deux trajectoires distinctes de longueur approximativement 150 m, composées de 10 virages et réalisées dans un parking souterrain avec le départ et l'arrivée sur la même place de parking. Nous avons pu constater à la fin de ces trajectoires des erreurs de positionnement de 0.09 m et respectivement 0.12 m et des erreurs d'orientation inférieures à 2° par rapport à la position réelle. La figure III.9(b) illustre quatre trajectoires de longueurs supérieures à la limite de 500 m imposée dans notre application. Pour les trois premières trajectoires (bleue, verte, jaune) les erreurs au point final de la trajectoire sont inférieures à 0.3 m d'après une estimation réalisée utilisant la carte Google. Pour la trajectoire la plus longue (1920 m), en magenta, une erreur importante est observée à la fin du parcours. En pratique, nous avons réalisé un demi-tour sur du gravier à une vitesse approximative de 10 km/h au niveau du point noté avec «I» dans la carte. En sortie du demi-tour nous avons accumulé une erreur angulaire de 1.5° qui s'est transformée en une erreur de position d'approximativement 6,5 m après 250 m parcourus.

Avant de tirer les conclusions, nous attirons l'attention sur un phénomène observé lors de la création de la trajectoire utilisant le modèle différentiel. L'angle θ du véhicule dans le repère global \mathfrak{R}_m présente une oscillation (figure III.11(b)). Nous rappelons que la résolution angulaire du capteur ESP est de $\frac{2\pi R_{roue}}{N}$ avec $N = 48$. Ce qui revient à une résolution linéaire par roue d'approximativement 4 cm pour un rayon de roue de $R_{roue} = 0,31$ m. Les figures III.10(a) à (c) illustrent une situation de déplacement rectiligne à très basse vitesse ($\delta_{topsArrG} = \delta_{topsArrD}$) où la réception de l'information se réalise décalée dans le temps. Dans ce cas, lors du calcul de la nouvelle variation angulaire ω pour un décalage d'un seul top roue, ω sera égale à 1.546° (cf. équation (III.32)).

$$\omega = \frac{2\pi R_{roue}}{48} \cdot \frac{\delta_{topsArrD} - \delta_{topsArrG}}{2b} = \frac{2\pi \cdot 0,310}{48} \cdot \frac{1}{2 \cdot 0,753} = 0,027 \text{ rad} = 1.546^\circ \quad (\text{III.32})$$

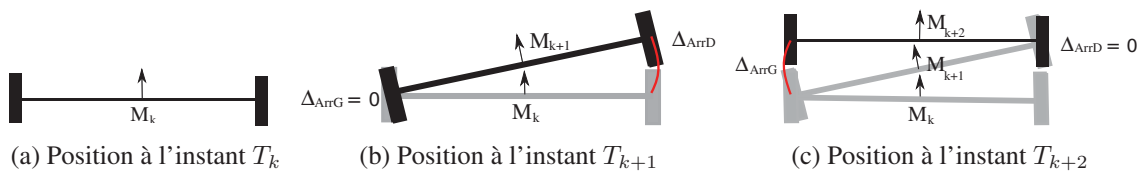


FIGURE III.10 – Interprétation du mouvement rectiligne du véhicule à basse vitesse dû à la réception décalée des mesures Δ_{ArrG} et Δ_{ArrD}

Ce problème peut être résolu de deux manières. La première approche consiste à réécrire les équations de Δ et ω en fonction de la vitesse de chaque roue. Si le calcul des vitesses se réalise sur une fenêtre glissante alors l'estimation du cap ne présentera plus d'oscillations. Mais nous ne disposons pas des temps d'acquisitions des données et nous savons également que le bus CAN ne garantit pas une fréquence constante entre les envois des données. Par conséquent nous ne choisissons pas cette approche car la division de la distance au laps de temps, est source d'erreurs d'estimation à la fois sur le cap mais également sur la pose du véhicule.

La solution choisie est de continuer d'estimer la trajectoire en utilisant les déplacements et les rotations élémentaires et de calculer un nouveau cap par une méthode de filtrage. Comme dans

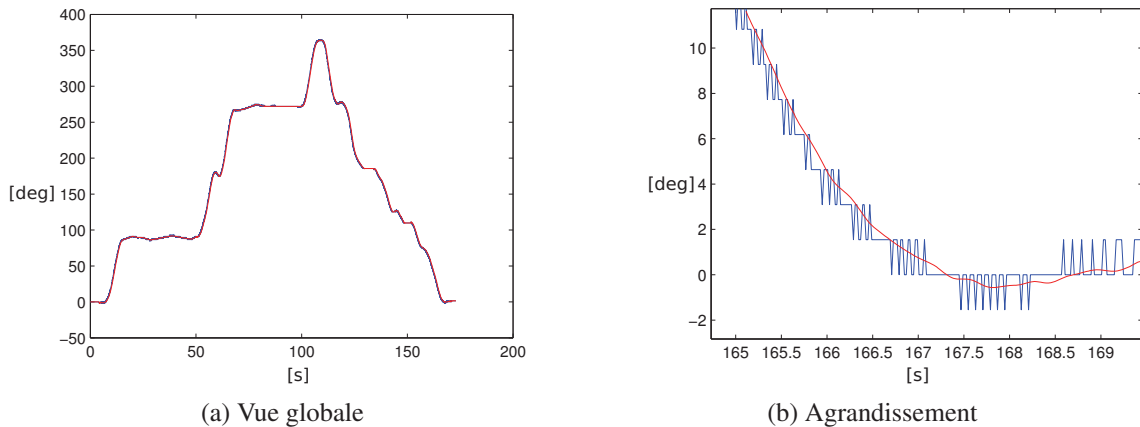


FIGURE III.11 – Évolution dans le temps du cap du véhicule dans le repère initial (\mathcal{R}_0) pour une des trajectoires de la figure III.9(a). La couleur bleu désigne l'angle estimé par la méthode différentielle (algorithme 2), le rouge l'angle estimé après le filtrage

la méthode précédente, l'angle filtré présentera un décalage temporel qui peut être observé dans la figure III.11(b). Ce décalage peut être maîtrisé en fonction du filtre et des paramètres choisis et compensé en commande par l'utilisation de principes prédictifs.

La conclusion que nous pouvons tirer en fin de cette section est que l'estimation odométrique est très adaptée à des trajectoires d'une longueur inférieure à 500 m acquises à des basses vitesses (2 m/s pour notre application) sur du sol à grand coefficient d'adhérence. Par contre, cette approche ne permet que de se localiser relativement à un repère établi en début de la trajectoire. Dans les sections suivantes nous étudions également d'autres approches de localisation du véhicule utilisant la vision ou les capteurs ultrason pour réaliser une estimation de la pose du véhicule dans le même repère pour plusieurs passages du véhicule dans le même couloir de circulation.

III.3.2 Localisation par techniques de SLAM

Comme nous l'avons déjà précisé, il est peu probable que l'environnement où le système de voiturier autonome évoluera soit cartographié. C'est pour cela que nous envisageons de faire appel à des techniques de SLAM qui ont la capacité de créer une carte de l'environnement et de localiser le véhicule dans cette carte. Nous nous attardons d'abord sur les techniques de SLAM dites locales qui approximent généralement les équations d'état ou d'observation par un développement de Taylor. Ensuite nous reviendrons sur les méthodes globales qui, par leur construction, utilisent l'intégrité de l'espace d'état pour fournir le résultat. Entre les deux approches il y a une différence fondamentale de temps de calcul, la première méthode étant plus légère donc plus adaptée aux traitements en temps réel. En revanche la deuxième méthode, pour un coût calculatoire plus important, donne des résultats plus précis.

L'utilisation d'une technique de SLAM basé sur le filtre de Kalman (KF-SLAM ou EKF-SLAM) semble un bon choix du point de vue temps réel. Il reste à évaluer les résultats obtenus et à décider si cette approche basée sur le filtrage est adaptée à notre situation de parking. Par la suite nous faisons appel aux travaux de T. Féraud sur le rejeu de chemin et la localisation monoculaire développés à l'Institut Pascal sous la direction de Roland Chapuis [Féraud, 2011]. Le projet développé avait pour but de permettre à un véhicule de se localiser en utilisant des cartes légères, d'une taille inférieure à 10 MB par kilomètre de trajectoire. Des études ont été menées pour réaliser un format

de carte facilement transportable et adapté à la transmission et au partage entre plusieurs véhicules se déplaçant dans le même environnement [Féraud et al., 2010a]. Pour notre application ces contraintes de partage de données par la communication n'existent pas, ce qui devrait se traduire par des cartes plus riches, donc une possible amélioration de la précision de localisation.

Comme le nom l'indique l'EFK-SLAM est basé sur le filtre de Kalman dont nous rappelons en préambule ses équations et leurs significations. Les trois étapes qui composent le filtre EKF sont : la prédiction (équations (III.33)), le calcul du gain (équations (III.34)) et la mise à jour (équations (III.35)).

$$\text{Prédiction : } \begin{cases} \vec{X}_{k+1/k} & = \mathbf{f}(\vec{X}_{k/k}, u_{k+1}) \\ \mathbf{P}_{k+1/k} & = \mathbf{F}_{k+1} \mathbf{P}_{k/k} \mathbf{F}_{k+1}^T + \mathbf{Q}_{k+1} \end{cases} \quad (\text{III.33})$$

$$\text{Gain : } \begin{cases} \mathbf{G}_{k+1/k} & = \mathbf{P}_{k+1/k} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1/k} \mathbf{H}_{k+1}^T + \mathbf{R}_{\mathbf{o}_{k+1}})^{-1} \end{cases} \quad (\text{III.34})$$

$$\text{Mise à jour : } \begin{cases} \vec{X}_{k+1/k+1} & = \vec{X}_{k+1/k} + \mathbf{G} (z_{k+1} - \mathbf{h}(\vec{X}_{k+1/k})) \\ \mathbf{P}_{k+1/k+1} & = \mathbf{P}_{k+1/k} - \mathbf{G}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_{k+1/k} \end{cases} \quad (\text{III.35})$$

où $\vec{X}_{k/k}$ représente le vecteur d'état, $\mathbf{P}_{k/k}$ la matrice d'incertitude associée au vecteur d'état, \mathbf{Q}_k la matrice de bruit de processus, $\mathbf{R}_{\mathbf{o}_k}$ la matrice de bruit de mesure, \mathbf{f} et \mathbf{h} les fonctions d'évolution et d'observation et \mathbf{F}_k et \mathbf{H}_k leur jacobienne par rapport au vecteur d'état.

III.3.2.1 Localisation dans une carte

Nous supposons d'abord que la carte est connue. Nous détaillons d'abord la localisation du véhicule, ce qui revient à estimer sa pose à l'instant présent utilisant la pose de l'instant précédent et les dernières informations envoyées par les capteurs odométriques et la caméra.

La première étape de prédiction du filtre est divisée en deux : la prédiction de l'état du véhicule (utilisant un modèle d'évolution par odométrie) et la prédiction des points d'intérêt dans l'image (utilisant une approche de recherche par focalisation).

Soit le vecteur des variables d'état $[X_{v_k}, Y_{v_k}, Z_{v_k}, \alpha_{x_k}, \alpha_{y_k}, \alpha_{z_k}]$ composé des trois coordonnées et des trois angles de rotation du centre de l'essieu des roues arrière du véhicule exprimées dans le repère \mathfrak{R}_m . En reprenant le même modèle d'évolution odométrique présenté dans la section précédente l'équation (III.28) d'évolution du véhicule se transcrit comme suit :

$$\begin{cases} X_{v_{k+1}} = X_k + \Delta \cos\left(\alpha_{z_k} + \frac{\omega}{2}\right) \cos(\alpha_{y_k}) \\ Y_{v_{k+1}} = Y_k + \Delta \sin\left(\alpha_{z_k} + \frac{\omega}{2}\right) \cos(\alpha_{y_k}) \\ Z_{v_{k+1}} = Z_k - \Delta \sin(\alpha_{y_k}) \end{cases} \quad \text{et} \quad \begin{cases} \alpha_{x_{k+1}} = \alpha_{x_k} \\ \alpha_{y_{k+1}} = \alpha_{y_k} \\ \alpha_{z_{k+1}} = \alpha_{z_k} + \omega \end{cases} \quad (\text{III.36})$$

avec Δ et ω le déplacement et la rotation élémentaires qui peuvent être exprimés par un modèle odométrique différentiel, voir section §III.3.1.2, ou bien par un modèle bicyclette, comme développé par T. Féraud. Ce qui est important à signaler est le fait que le filtre de Kalman permet d'estimer l'incertitude au travers de la matrice variance-covariance, ou bien de recalculer la covariance suite à l'évolution de l'état du véhicule. Pour ces calculs, la jacobienne du modèle d'évolution selon les variables d'état et selon les variables d'entrée doit être explicitée. En analysant le système d'équations (III.36), nous observons que les informations proprioceptives Δ et ω permettent la mise à jour uniquement de quatre des six composants du vecteur d'état. Par conséquent un bruit est ajouté sur les paramètres afin qu'ils puissent évoluer, sinon le filtre de Kalman considèrera leurs valeurs comme étant sûres et il ne les remettra pas en cause. Dans leurs travaux, P. Delmas et

T. Féraud tiennent compte pour la définition du bruit de la capacité du véhicule à franchir des obstacles, ou bien que la pente de l'environnement ne dépasse pas une certaine inclinaison maximale (α_y^{max}) [Delmas, 2011; Féraud, 2011]. Ces contraintes sont transposées à travers les équations (III.37).

$$\begin{cases} \sigma_x = \Delta \cos(\alpha_{z_k}) \left(1 - \cos(\alpha_y^{max})\right) \\ \sigma_y = \Delta \sin(\alpha_{z_k}) \left(1 - \cos(\alpha_y^{max})\right) \\ \sigma_z = \Delta \sin(\alpha_y^{max}) \end{cases} \quad \text{et} \quad \begin{cases} \sigma_{\alpha_x} \ll \sigma_{\alpha_y} \\ \sigma_{\alpha_y} = \alpha_y^{max} \\ \sigma_{\alpha_z} \ll \sigma_{\alpha_y} \end{cases} \quad (\text{III.37})$$

Le modèle d'évolution odométrique du véhicule (équation III.36) sera complété par de l'information visuelle. En reprenant le formalisme défini dans la section §II.3.2.4 sur la calibration de la caméra, un point 3D \vec{P}_m exprimé dans le repère \mathfrak{R}_m est transcrit en coordonnées caméra ainsi :

$$\vec{P}_c = \mathbf{R}_{(v \rightarrow c)}^T \left(\mathbf{R}_{(m \rightarrow v)}^T \left(\vec{P}_m - \vec{T}_{(m \rightarrow v)} \right) - \vec{T}_{(v \rightarrow c)} \right) \quad (\text{III.38})$$

ou bien en coordonnées image (u, v) , pour un modèle de projection centrale, sans distorsion :

$$\begin{cases} u = \frac{\vec{K}_1 \mathbf{R}_{(v \rightarrow c)}^T \left(\mathbf{R}_{(m \rightarrow v)}^T \left(\vec{P}_m - \vec{T}_{(m \rightarrow v)} \right) - \vec{T}_{(v \rightarrow c)} \right)}{\vec{K}_3 \mathbf{R}_{(v \rightarrow c)}^T \left(\mathbf{R}_{(m \rightarrow v)}^T \left(\vec{P}_m - \vec{T}_{(m \rightarrow v)} \right) - \vec{T}_{(v \rightarrow c)} \right)} = \frac{N_u}{D_{uv}} \\ v = \frac{\vec{K}_2 \mathbf{R}_{(v \rightarrow c)}^T \left(\mathbf{R}_{(m \rightarrow v)}^T \left(\vec{P}_m - \vec{T}_{(m \rightarrow v)} \right) - \vec{T}_{(v \rightarrow c)} \right)}{\vec{K}_3 \mathbf{R}_{(v \rightarrow c)}^T \left(\mathbf{R}_{(m \rightarrow v)}^T \left(\vec{P}_m - \vec{T}_{(m \rightarrow v)} \right) - \vec{T}_{(v \rightarrow c)} \right)} = \frac{N_v}{D_{uv}} \end{cases} \quad (\text{III.39})$$

où $\vec{T}_{(m \rightarrow v)}$ et $\mathbf{R}_{(m \rightarrow v)}^T = \mathbf{Rot}_z(\alpha_{z_c}) \mathbf{Rot}_y(\alpha_{y_c}) \mathbf{Rot}_x(\alpha_{x_c})$ représentent le vecteur de translation et la matrice de rotation du repère monde \mathfrak{R}_m vers le repère véhicule \mathfrak{R}_v , de même que $\vec{T}_{(v \rightarrow c)}$ et $\mathbf{R}_{(v \rightarrow c)}$ représentent le vecteur de translation et la matrice de rotation du repère véhicule vers le repère caméra \mathfrak{R}_c et \vec{K}_i désignant le $i^{\text{ème}}$ ligne de la matrice des paramètres intrinsèques \mathbf{K} .

Nous disposons donc des relations qui relient les prédictions $\vec{y}^i = [\hat{u}_i, \hat{v}_i]^T$ des points \vec{P}_i (exprimés en repère monde \mathfrak{R}_m) à travers le vecteur d'état $X_{k/k} = [X_{v_k}, Y_{v_k}, Z_{v_k}, \alpha_{x_k}, \alpha_{y_k}, \alpha_{z_k}]^T$ composé des six paramètres définissant une position et orientation uniques dans le repère monde \mathfrak{R}_m . La liaison entre les mesures $[u_i, v_i]^T$ dans l'image et leurs prédictions $[\hat{u}_i, \hat{v}_i]^T$ se réalise à travers l'équation (III.40), où $\vec{\epsilon}_i = [\epsilon_{ui}, \epsilon_{vi}]^T$ est un bruit de mesure blanc et gaussien.

$$\vec{y}_i = \mathbf{h}(\vec{X}) + \vec{\epsilon}_i \quad (\text{III.40})$$

Il est à remarquer que l'observation d'un point est une fonction qui dépend uniquement du point 3D en question et de la position du véhicule. La caméra étant fixe sur le véhicule, ses paramètres n'apparaissent pas dans le vecteur d'état. Disposant en plus des jacobiennes, il est possible de définir une zone de recherche d'un point 3D dans l'espace image donc de réaliser une approche de recherche par focalisation et donc potentiellement d'optimiser le temps de calcul pour respecter la contrainte temps réel.

Le problème qui reste à traiter est l'association du point 3D de la carte avec l'observation dans l'image. L'observation dans l'image fait référence à l'identification de certains points qui disposent des propriétés les différenciant des autres points dans son voisinage. Un point 3D tel quel ne peut pas être reconnu dans une image. C'est l'environnement autour de ce point qui le rend reconnaissable. A titre d'exemple les détecteurs les plus utilisés sont :

- les détecteur de coins comme : Harris [Harris and Stephens, 1988; Zhao et al., 2008], Shi-Tomasi [Tomasi and Kanade, 1991; Shi and Tomasi, 1994], Harris-Laplace, Hessian-Laplace, Harris-Affine, Hessian-Affine [Mikolajczyk and Schmid, 2004], FAST (Feature from Accelerated Segment Test) [Viswanathan,

2009], SUSAN (Smallest Univalued Segment Assimilating Nucleus) [Smith and Brady, 1997];

- les détecteurs de zones comme : LOG (Laplacian Of Gaussian), DOG (Difference Of Gaussian) utilisés dans SIFT [Lowe, 2004], MSERs (Maximally Stable Extremal Regions) [Matas et al., 2004] ou Fast-Hessian utilisé dans SURF [Bay et al., 2006, 2008].

Leur fiabilité est définie par la capacité à extraire les mêmes points (ou zone) pour une image qui a subi une rotation, un changement d'illumination ou un changement d'échelle. Pour plus de détails sur leurs propriétés et performances nous citons les travaux de [Schmid et al., 2000] et de [Mikolajczyk and Schmid, 2005].

Dans le cas qui nous intéresse un descripteur image lui est associé ainsi que la position de la caméra lors de la prise de vue. Bien entendu cette imagerie change en fonction du point de prise de vue et des conditions d'éclairages, d'où la complexité de l'association 2D - 3D. L'approche classique consiste à ne pas rechercher une imagerie dans toute une image mais de se restreindre à une zone précise afin de limiter les mauvais appariements et de limiter aussi le temps de calcul. La zone de recherche peut être déduite par propagation des incertitudes à l'aide de l'équation (III.41) :

$$\mathbf{P}_{2D} = \mathbf{H}_{\hat{X}} \mathbf{P} \mathbf{H}_{\hat{X}}^T + \mathbf{H}_{\vec{P}_i} \mathbf{P}_{\vec{P}_i} \mathbf{H}_{\vec{P}_i}^T \quad (\text{III.41})$$

où \vec{P}_i est un point 3D, $\mathbf{P}_{\vec{P}_i}$ est la matrice de variance-covariance associée, \hat{X} est le vecteur d'état estimé, \mathbf{P} est sa matrice d'incertitude associée, $\mathbf{H}_{\hat{X}}$ et $\mathbf{H}_{\vec{P}_i}$ sont respectivement les jacobiennes du modèle d'observation h par rapport au vecteur d'état et par rapport au point courant \vec{P}_i . Cette équation réalise le lien entre l'ellipsoïde d'incertitude exprimé dans l'espace 3D du repère monde à l'ellipse d'incertitude équivalente dans l'espace image. T. Féraud démontre les limites de cette approche en illustrant des cas où l'échec de l'appariement est survenu car une partie de l'espace d'incertitude a été ignoré [Féraud, 2011]. En complément il développe une approche géométrique pour la définition d'une zone de recherche d'un point 3D $\vec{P}_0 = [x_0, y_0, z_0]^T$ dans l'image. Cette zone est définie comme une enveloppe rectangulaire de la projection de l'ellipsoïde dans le plan de l'image. Les quatre points qui définissent la zone sont notés $\vec{P}_{vi} = [x_{vi}, y_{vi}, z_{vi}]^T$ et $\vec{P}_{hi} = [x_{hi}, y_{hi}, z_{hi}]^T$ pour $i \in \{1, 2\}$, les points conduisant à des tangentes verticales respectivement horizontales de l'enveloppe, voir annexe A.4.

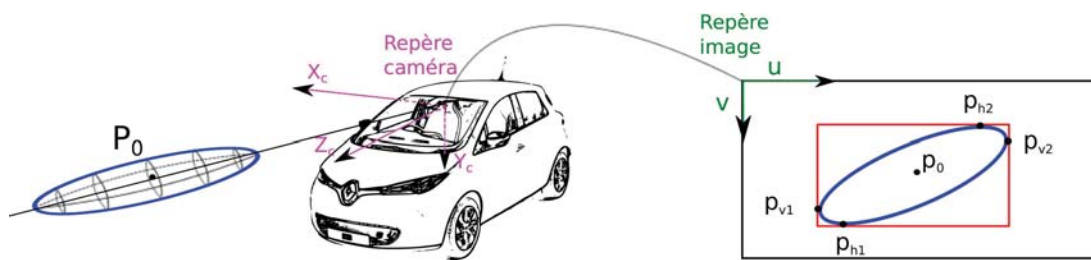


FIGURE III.12 – Projection de l'ellipsoïde d'incertitude du repère caméra \mathcal{R}_c dans une ellipse d'incertitude dans le repère image \mathcal{R}_{Im} et calcul du rectangle englobant (en rouge)

Les points $\vec{P}_{v1}, \vec{P}_{v2}, \vec{P}_{h1}$ et \vec{P}_{h2} ainsi calculés, seront projetés dans l'image et notés p_{v1}, p_{v2}, p_{h1} et p_{h2} . Ces derniers points se situent sur l'ellipse d'incertitude et permettent de retrouver les limites de la zone rectangulaire englobant la projection de l'incertitude (voir figure III.12).

Nous disposons d'outils qui nous permettent de diminuer la zone de recherche d'un point dans l'image. L'étape suivante est de reconnaître le point associé à une prédiction et cela se réalise à travers l'opérateur d'association de descripteurs. Comme nous l'avons précisé, à chaque point nous associons des descripteurs de type imagerie qui nous permettront de réaliser l'association entre les points. Dans la littérature il existe principalement deux classes de méthodes de

recherche d'appariement basées sur la minimisation de la différence ou sur la maximisation de la vraisemblance. Notons \mathbf{I} et \mathbf{J} deux descripteurs à comparer. La première classe peut utiliser des opérateurs comme la norme $L_1 = \sum_i \sum_j |\mathbf{I}_{ij} - \mathbf{J}_{ij}|$ ou la norme $L_2 = \sum_i \sum_j (\mathbf{I}_{ij} - \mathbf{J}_{ij})^2$ pour estimer la différence. Quant à la deuxième classe, la maximisation de la vraisemblance se réalise à travers des opérateurs du type CC (Cross-Corrélation), ZCC (Cross-Corrélation centrée en zéro), NCC (Cross-Corrélation normalisée) ou bien ZNCC (Cross-Corrélation normalisée et centrée en zéro) [Ortega, 2007]. Le choix de l'opérateur se réalise en mettant en balance les performances obtenues et le temps de calcul. Indépendamment de la méthode utilisée, le temps de calcul augmente pour des descripteurs plus grands mais permet simultanément de mieux discriminer les points. Dans les applications du type SLAM, la taille de descripteurs varie entre 10×10 et 20×20 pixels.

Supposons disposer d'une trajectoire de référence et d'une carte de référence composée des positions 3D des points cartographiés (les points en couleur cyan dans les figures III.13(a) à (c)). De plus pour chaque point de la carte nous disposons des éléments suivants :

- la position du véhicule lors de la dernière observation du point dans une image ;
- un ellipsoïde d'incertitude ;
- un descripteur associé au point d'intérêt.

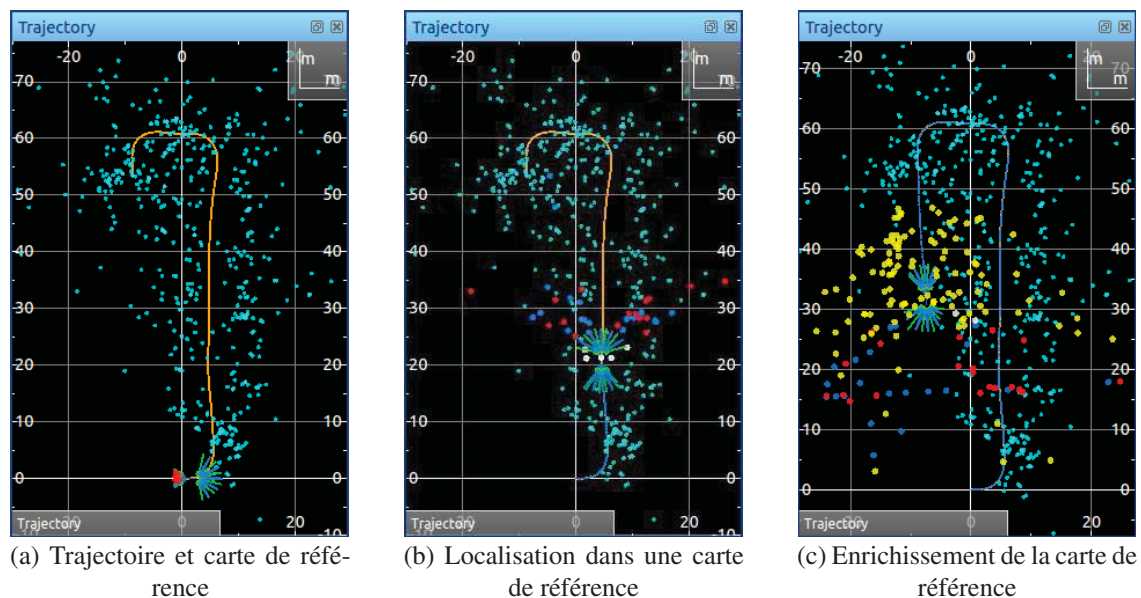


FIGURE III.13 – La figure (a) illustre en couleur cyan les positions des points d'intérêt cartographiés et en orange la trajectoire du véhicule lors de la création de la carte. Dans la figure (b) est présenté un véhicule se localisant dans la carte ; en couleurs bleu et rouge sont indiqués les états d'appariement ou de non-appariement des points de la carte avec des points vus en temps réel. Enfin, la figure (c) illustre par les points jaunes une nouvelle carte créée en temps réel pour compléter les endroits non cartographiés (cette étape sera détaillée dans la section §III.3.2.2)

La figure III.13(b) correspond à une situation de localisation dans une carte de référence. Pour première validation de l'algorithme nous avons utilisé un jeu de données composées des images prises au long de la trajectoire de référence mais qui n'ont pas servi à la création de la carte. Dans ses travaux [Féraud, 2011] annonçait une erreur inférieure à 0,08 m. La figure III.15 indique l'écart latéral calculé dans les mêmes conditions. Les deux couleurs rouge et vert indiquent une information de détection d'un déplacement réalisable ou non par le véhicule. Par conséquent, un filtrage peut être réalisé pour garder uniquement les données cohérentes. Si nous analysons

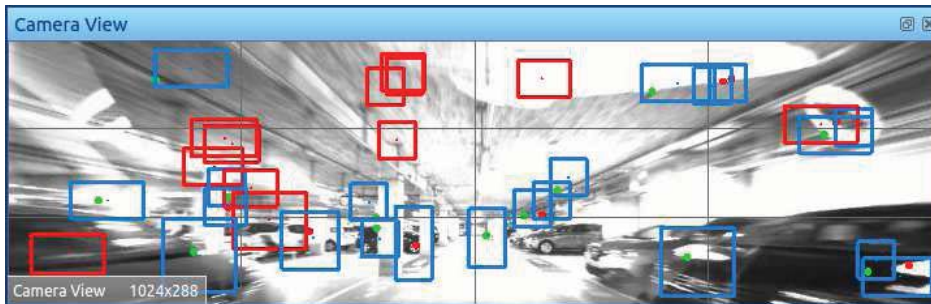


FIGURE III.14 – Les rectangles englobant les ellipses d’incertitudes des projections des points de la carte de référence dans l’image ont une couleur bleue et rouge en fonction de la réussite ou non de l’appariement ; les points verts correspondent aux positions exactes des points mis en correspondance

uniquement les données illustrées en couleur verte, nous constatons que l’erreur de positionnement est inférieure à 0,12 m. L’erreur obtenue est supérieure parce que la résolution du capteur odométrique utilisé est inférieure à celle du capteur utilisé par T. Féraud dans son application. La deuxième source d’augmentation de l’erreur vient du fait que nous utilisons une caméra fish-eye qui a l’inconvénient de distordre beaucoup l’image vers l’extrémité du champ de vue. Même si l’image est corrigée, des erreurs dues à la distorsion sont transposées dans des erreurs d’appariement par le biais des descripteurs des points d’intérêt qui sont construits à partir de l’image corrigée.

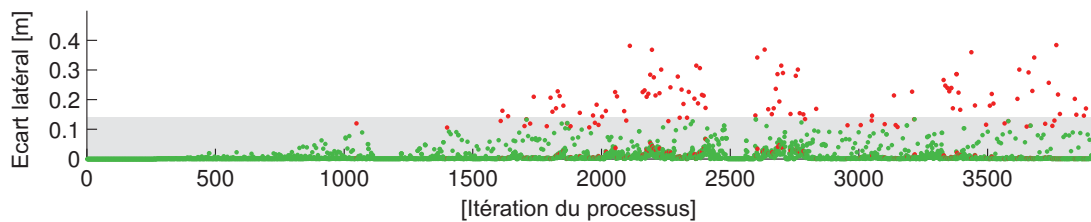


FIGURE III.15 – Écart latéral mesuré entre la trajectoire de référence et le résultat par la méthode de localisation. Les deux couleurs rouge et vert indiquent une information de détection d’un déplacement réalisable ou non par le véhicule.

En réalisant plus d’essais avec d’autres trajectoires nous constatons que cette approche est adaptée pour des situations de suivi de trajectoire. Par exemple pour la navigation autonome des véhicules de type navette [§I.3.4] qui ont des trajectoires fixes ou pour des situations de convois où le premier véhicule partage la carte avec des véhicules suiveurs.

Le principe de la méthode repose sur la projection des points 3D dans le plan image pour les mettre en correspondance avec des points vus en temps réel. Si le véhicule dérive, alors la probabilité de réaliser des mauvais appariements augmente. Une fois que le véhicule réalise plusieurs mauvaises estimations de sa position, comme la localisation est récursive, le véhicule ne pourra plus retrouver sa vraie position. Lors des essais nous avons constaté que cette technique de localisation a l’inconvénient d’être très sensible face à l’incertitude de la position initiale. Pour une grande incertitude initiale du véhicule, la zone de recherche d’un point sera également grande, ce qui peut conduire à des mauvais appariements. Nous avons constaté que la richesse de l’environnement, la ressemblance des objets et des éloignements des objets jouent sur la réussite de la relocalisation dans la carte. Nos constats nous amènent à affirmer que lors de l’étape de localisation initiale l’incertitude sur la position initiale ne doit pas dépasser 0,8 m. Concernant l’erreur angulaire tolérée, elle est dépendante du recouvrement du champ de vue courant avec celui observé

lors de la création de la carte. Toujours de manière expérimentale, nous avons constaté qu'au-delà d'une erreur angulaire initiale de $\pm 20^\circ$ la probabilité que le véhicule se localise est très faible.

La figure III.13(c) nous montre une situation où à la fin du couloir cartographié, le véhicule démarre une étape de création d'une nouvelle carte (en couleur jaune) car il constate que des points 3D ne sont plus disponibles pour réaliser la localisation. Potentiellement notre problème de navigation peut être résolu par la création d'une nouvelle carte dans les situations où le véhicule emprunte une autre trajectoire et enfin de fusionner les cartes. Par la suite nous détaillons cette étape de création de la carte.

III.3.2.2 Création de la carte

Nous avons précédemment montré les résultats obtenus pour la localisation du véhicule dans une carte pour une position initiale connue. Maintenant nous nous préoccupons de la problématique de la création de la carte. Pour cela il faut extraire des points d'intérêt dans l'image et calculer leurs positions 3D au fur et à mesure du déplacement.

Pour assurer une meilleure répartition des points d'intérêt, chaque image est découpée en m lignes et n colonnes. De cette façon nous pouvons contrôler la répartition des points dans chacune des $m \times n$ zones créées. A titre d'exemple, nous savons que dans un environnement extérieur, la partie supérieure de l'image est occupée généralement par le ciel et le nombre de points d'intérêt extraits peuvent être inférieur au reste de l'image. De plus, puisque les détecteurs de coins calculent des scores par pixel compte tenu de son voisinage et indépendamment du calcul du score d'un autre pixel, cette étape est parallélisée permettant ainsi de diminuer le temps total de calcul.

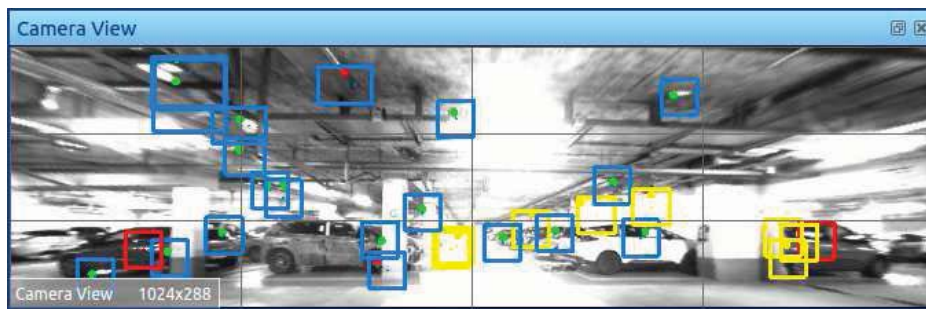


FIGURE III.16 – Détection et suivi des points d'intérêt dans plusieurs images successives. Les rectangles englobants des points d'intérêt ont une couleur en fonction du niveau de la précision de la position 3D du point correspondant : rouge pour les points les plus incertains, puis bleu, terminant par la couleur jaune pour les points ayant atteint la précision requise pour que le point soit stocké dans la carte. Enfin, si un point n'a pas atteint le niveau «jaune» avant de disparaître du champ de vue, il sera rejeté et un autre point d'intérêt coloré en magenta le remplacera

A la suite de l'extraction des points d'intérêt, nous nous intéressons à l'estimation des points 3D correspondants. Donc un point exprimé dans le repère capteur peut être transféré dans le repère monde utilisant l'équation (III.42) :

$$\vec{P}_m = \mathbf{R}_{(m \rightarrow v)} \left(\mathbf{R}_{(v \rightarrow c)} \vec{P}_c + \vec{T}_{(v \rightarrow c)} \right) + \vec{T}_{(m \rightarrow v)} \quad (\text{III.42})$$

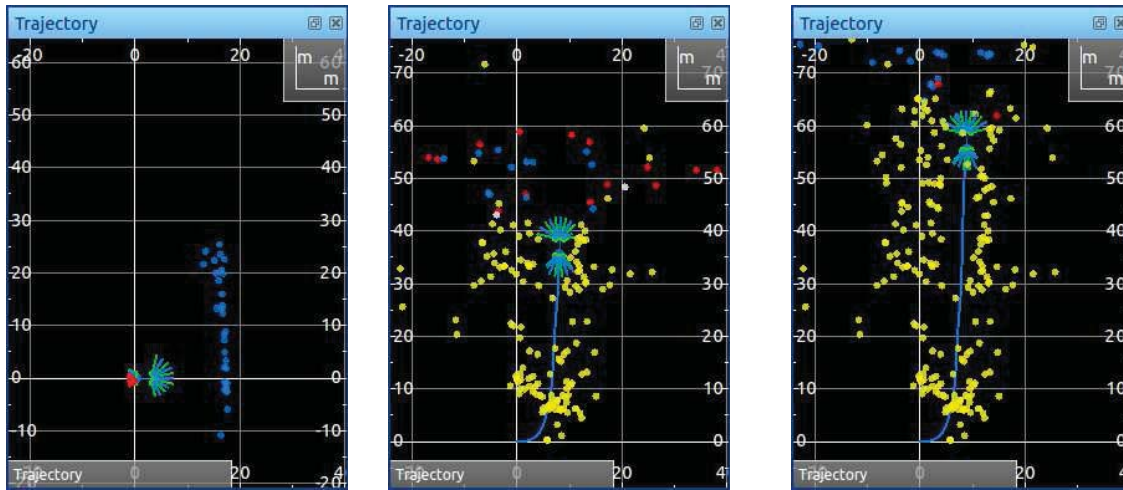
Considérant $\vec{P}_c = [x_c, y_c, z_c]^T$ et connaissant les coordonnées (u, v) du point p_i dans

l'image, alors les coordonnées du point 3D s'écrivent ainsi :

$$\vec{P}_c = \left[\frac{u - u_0}{f_u} z_c, \frac{v - v_0}{f_v} z_c, z_c \right]^T \quad (\text{III.43})$$

La seule inconnue est la profondeur z_c qui ne peut pas être déterminée utilisant une seule image. Une approche est d'initialiser z_c à une valeur constante Z_0 puis d'associer au point \vec{P}_i une incertitude initiale $\mathbf{P}_{\vec{P}_i}$. Cette incertitude peut être déterminée en calculant les paramètres de l'ellipsoïde d'incertitude dont la projection est connue par une incertitude définie dans l'image. L'équation (III.44) montre le résultat démontré dans [Féraud, 2011].

$$\begin{cases} \sigma_x = \frac{\sigma_u}{f} \sqrt{\rho^2 - \sigma_z^2} \\ \sigma_y = \frac{\sigma_v}{f} \sqrt{\rho^2 - \sigma_z^2} \\ \sigma_z = \rho - \delta \end{cases} \quad \text{et} \quad \begin{cases} \rho = \sqrt{x_c^2 + y_c^2 + Z_0^2} \\ \delta = \text{distance minimale d'observation} \end{cases} \quad (\text{III.44})$$



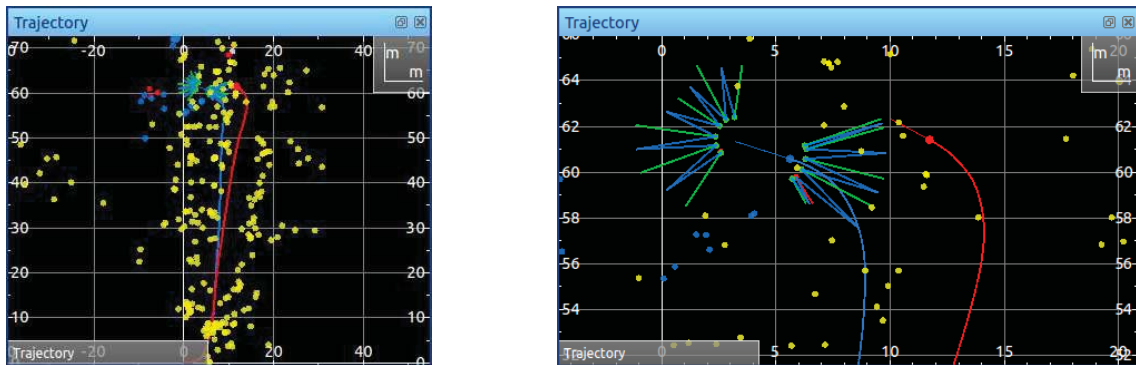
(a) Carte à l'instant t_1

(b) Carte à l'instant t_2

(c) Carte à l'instant t_3

FIGURE III.17 – Processus de SLAM. La figure (a) illustre l'étape d'initialisation des points observés dans l'image à une distance Z_0 donnée dans l'espace 3D. Dans les figures (b) et (c) est présentée la carte à différents instants du processus de création. Les couleurs des points (rouge, bleu, jaune) correspondent aux couleurs des rectangles englobants présentés dans la figure III.16

Par rapport à l'étape de localisation où les appariements entre les observations et la carte sont utilisées pour déterminer la position du véhicule, dans la création de la carte il est nécessaire d'estimer les positions des points 3D dans la même étape d'estimation de la position du véhicule. Au démarrage de l'algorithme SLAM dans la carte il n'y a aucun point. Un nombre N de points p_i , $i \in [1, N]$ sont alors choisis dans l'image et initialisés, comme précédemment précisé, à une distance Z_0 , voir figure III.17(a). Le suivi continu de chaque point dans les images successives permet de réduire l'incertitude de position, voir figure III.16. Un point qui n'a pas encore convergé pose un problème de non linéarité à la fois dans le calcul de la zone de recherche dans l'image mais également dans le calcul du gain de Kalman. Il est connu que le filtre de Kalman est optimal uniquement pour les fonctions linéaires mais il a été démontré par [Ristic et al., 2004] qu'il le reste pour des fonctions non linéaires en cas de linéarisation autour d'un point vrai. Comme ce point vrai n'est pas disponible, la linéarisation se fait autour du point estimé ce qui fait que le filtre EKF est sous-optimal. Lors de la création de la carte uniquement les points qui ont abouti à la convergence seront gardés.



(a) Cartographie de l'environnement

(b) Agrandissement d'une zone de la figure (a)

FIGURE III.18 – Situation de dérive et comparaison de la trajectoire réalisée uniquement par odométrie (en couleur bleu) et la trajectoire résultée lors du processus de SLAM par fusion de l'odométrie et de la vision

Malgré les avantages de la méthode de localisation EKF-SLAM, elle n'est pas complètement adaptée à notre situation de voiturier autonome. En effet, les essais ont montré que l'EKF-SLAM apporte une précision moindre et présente plus de dérive que l'estimateur odométrique. Si lors des réglages nous accordons trop de confiance aux données odométriques, le SLAM se comportera plus comme la localisation odométrique présentée dans la section antérieure. Par contre si nous donnons plus de confiance aux données visuelles, alors nous risquons plus de dérive, voir figures III.18(a) et (b). Par conséquent nous choisissons d'utiliser uniquement l'odométrie tenant compte du fait que pour obtenir des meilleurs résultats il faudrait investir plus de temps.

Tout le développement réalisé dans cette partie de localisation n'a pas été vain. Il a servi pour approfondir la problématique du SLAM visuel, pour comprendre sa complexité et le transformer en un algorithme du type *Structure from Motion* que nous détaillerons dans la section §IV.4 du quatrième chapitre. Dans les sections suivantes nous détaillerons deux autres techniques de localisation par appariement de cartes d'occupation et par appariement d'images. Enfin nous concluons sur la complémentarité des deux techniques et sur la stratégie de localisation utilisée pour le système de voiturier autonome.

III.3.3 Localisation par appariement des grilles d'occupation

III.3.3.1 Problématique

Comme nous l'avons déjà présenté, lors des phases d'apprentissage, le conducteur va réaliser d'abord les manœuvres de conduite et d'évitement d'obstacles pour garer le véhicule. Pendant cette étape, le système se charge d'enregistrer la trajectoire et tous les obstacles qu'il perçoit durant le déplacement. Ceci nous amène à employer une structure de données de type de grille d'occupation pour le type d'information que nous souhaitons utiliser. Dans le chapitre IV nous détaillerons la manière dont la grille d'occupation est initialement remplie par l'information issue des capteurs ultrason. Dans un second temps, nous allons pallier le problème du trop court champ de perception du capteur ultrason, en exploitant l'information vidéo.

Les figures III.19(a) et (b) illustrent deux cartes réalisées lors des deux campagnes d'enregistrements différents. Le résultat de leur superposition est réalisée manuellement et présenté dans

la figure III.19(c). Nous cherchons à réaliser un algorithme pour superposer automatiquement ces deux cartes, ce qui revient à déterminer la position d'une carte dans le repère de l'autre.

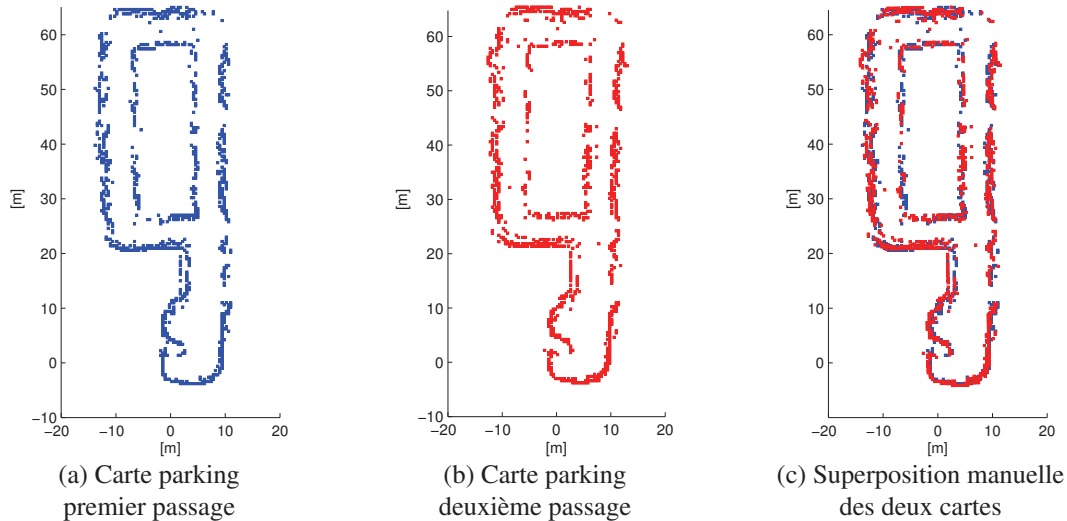


FIGURE III.19 – Comparaison des deux cartes ultrasons

III.3.3.2 Calcul de la translation par la méthode de corrélation

Considérons les cartes présentées dans les figures III.20(a) et (b) étant la *carte globale* respectivement la *carte locale*. Pour la clarté de la représentation nous utilisons une résolution de 0,4 m. Au fur et à mesure de l'avancement du véhicule une carte locale est construite pour l'apparier ultérieurement à la carte globale. Cela revient à déterminer la position du véhicule lors du deuxième passage dans le repère de la carte de référence. La méthode que nous adaptons à notre besoin d'appariement des cartes fait partie d'une grande famille de techniques qui s'appelle *Template Matching* et plus particulièrement *Template Matching using Correlation Coefficients*.

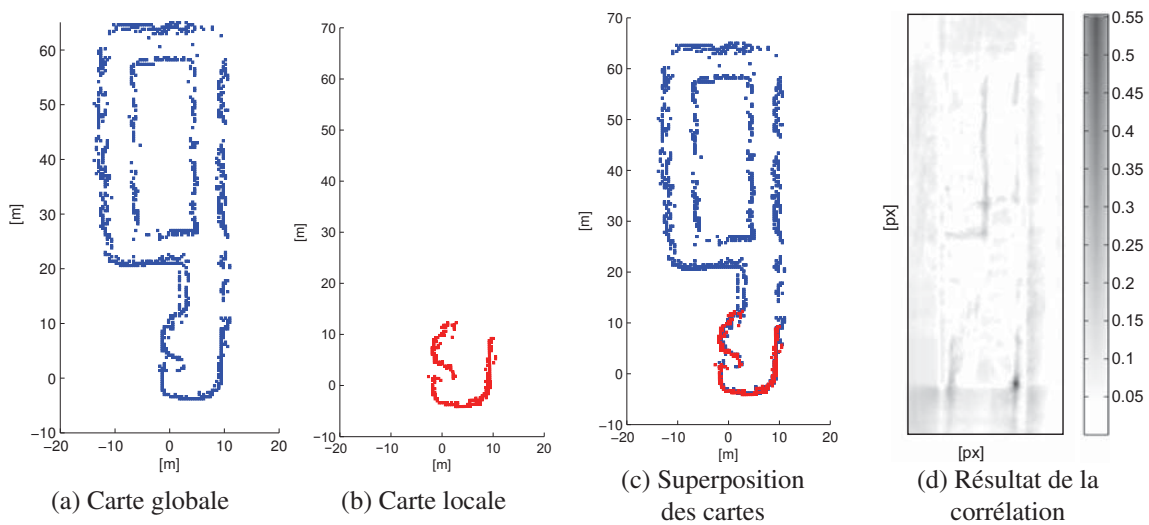


FIGURE III.20 – Exemple de résultat de la méthode d'appariement par corrélation; dans cet exemple la résolution de la grille est de 0,4 m

Supposons que les deux cartes sont alignées et essayons d'abord de retrouver la translation \vec{T} de la sous-carte qui correspond au mieux en matière de corrélation à la carte globale. La figure III.20(d) montre un premier résultat de la corrélation entre les deux cartes utilisant l'algorithme 6 détaillé dans l'annexe A.5. La valeur maximale 0,55 est retrouvée dans un point indiquant la position du barycentre des obstacles de la carte locale dans le repère de la carte globale. Le résultat de la superposition des deux cartes peut être observé dans la figure III.20(c).

A la suite, nous analysons l'évolution du résultat de la méthode de corrélation lors de la construction progressive de la carte locale. Dans les figures III.21(a) à (c) trois cartes locales (en couleur rouge) qui s'enrichissent au fur et à mesure du déplacement du véhicule sont affichées. Les trois résultats, aux trois instants différents peuvent être observés dans les figures III.21(d) à (f).

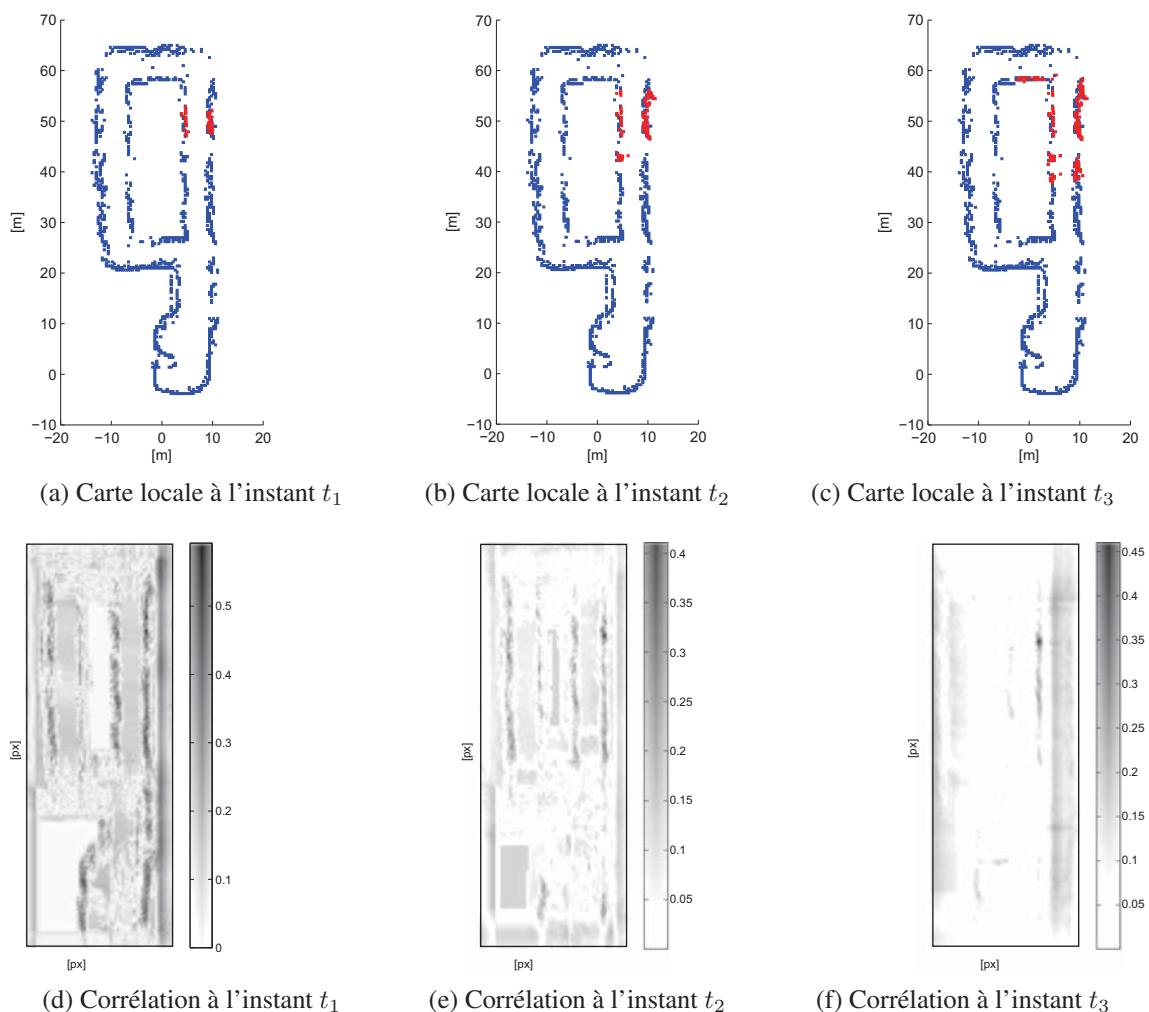


FIGURE III.21 – Évolution de la localisation

La figure III.21(d) montre qu'à l'instant t_1 la méthode ne fournit pas un résultat unique. Cela se justifie par la petite taille de la carte locale (en rouge) et par le fait que l'allée de circulation ressemble à beaucoup d'autres endroits dans le parking. A l'instant t_2 la méthode réussit déjà à fournir le bon résultat mais l'écart entre le maximum global et les autres maximum locaux est très faible, voir figure III.21(e). Par contre à l'instant t_3 le résultat est unique, voir figure III.21(d). Cela

s'explique par le fait que la distance parcourue par le véhicule est suffisamment grande, 16 m dans ce cas, et que le véhicule a réalisé un virage ce qui *personnalise* plus la carte locale.

III.3.3.3 Optimisation du temps de calcul utilisant des a priori sur la position du véhicule

Même si cette méthode donne des résultats robustes, elle présente l'inconvénient d'être très coûteuse en temps de calcul. Pour calculer une seule translation le temps de calcul est estimé à **38 ms** dans cette configuration des cartes (figures III.21(d) et (e)). Dans l'absolu il faudra répéter cette opération pour les 360° avec la résolution souhaitée. Nous avons fait cet exercice pour s'assurer de la robustesse de la méthode. Effectivement, les résultats sont très satisfaisants du point de vue de la qualité de positionnement mais l'approche est inadaptée à l'utilisation en temps réel car le temps du calcul s'élève à **13,6 s**. Supposons disposer d'un algorithme qui nous fournit des a priori de position et d'orientation du véhicule. Utilisant ces a priori, il est possible de réduire la zone de recherche et de cette façon de diminuer le temps de calcul global.

Prenons par exemple un seul angle de position de la carte locale et varions le rayon autour duquel se réalise la recherche. Les résultats sont présentés dans le tableau III.2.

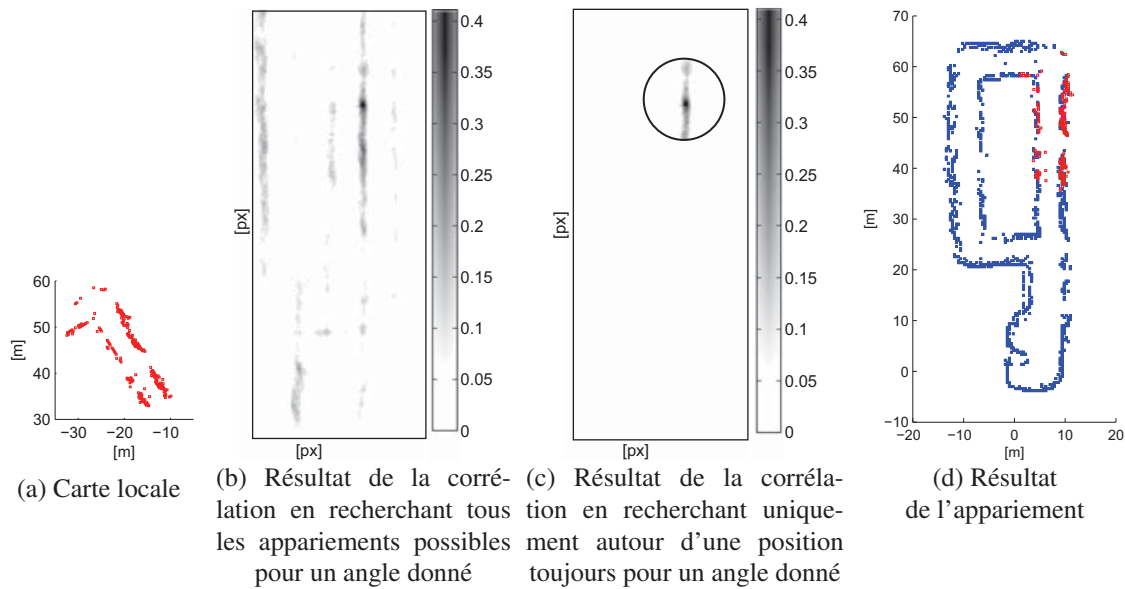


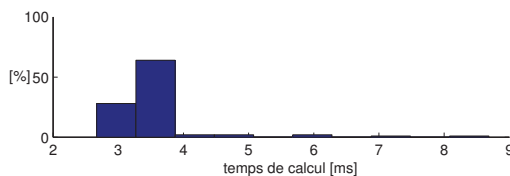
FIGURE III.22 – Exemple d'optimisation du temps de calcul utilisant un a priori sur la position

Rayon de recherche	$R = 16 \text{ m}$	$R = 8 \text{ m}$	$R = 4 \text{ m}$	$R = 2 \text{ m}$	$R = 1,2 \text{ m}$	$R = 0,8 \text{ m}$	$R = 0,4 \text{ m}$
Temps de calcul	10,5 ms	5,7 ms	4,6 ms	3,5 ms	2,15 ms	2,01 ms	1,95 ms

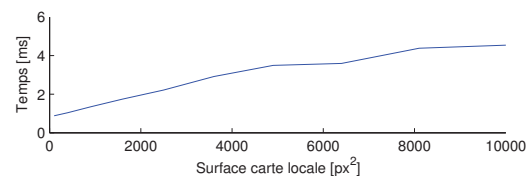
TABLE III.2 – Variation de la vitesse de calcul pour l'algorithme 6 détaillé dans l'annexe A.5 pour différents niveaux d'incertitude de localisation donnés sous forme d'un rayon de recherche R

En imposant un rayon de recherche R de 2 m autour d'une position donnée, nous constatons que le temps de calcul est de **3,5 ms**. Donc pour une plage angulaire $[-40^\circ, 40^\circ]$ le temps de calcul devrait être $\approx 3,5 \text{ ms} \times 81 \approx 280 \text{ ms}$. Mais grâce au calcul parallèle, il est possible de le réduire à **50 - 60 ms**. Le suivi des étapes ainsi que les détails du procédé de calcul de la roto-translation sont présentés dans l'algorithme 6 et une brève analyse du temps de calcul est présentée dans les

figures III.23(a) et (b). Nous pouvons observer des anomalies dans l'histogramme III.23(a) justifiées par le fait que pour paralléliser le calcul, la librairie *OpenMP*¹ est utilisée [Dagum and Enon, 1998; Chapman et al., 2008; Agoiz Badia, 2016]. Elle permet de gérer un programme avec des sections parallélisées et d'autres qui ne le sont pas. Comme nous n'avons pas accès à la gestion des interruptions du système ni les fils de calcul parallèle, il est possible de retrouver des allongements de temps de calcul de 6 à 8 ms comme présenté dans l'histogramme III.23(a).



(a) L'histogramme du temps de calcul de l'algorithme d'appariement par corrélation pour une carte locale de taille 5 000 px² et une carte globale de 10 000 px²



(b) L'influence de la surface de la carte locale (en px²) sur le temps de calcul de l'algorithme d'appariement de cartes

FIGURE III.23 – Analyse du temps de calcul de l'algorithme d'appariement des cartes (détaillé dans l'annexe A.5) pour des a priori sur la position sous forme d'un rayon de recherche $R = 2$ m

III.3.3.4 Optimisation du temps de calcul utilisant la structure géométrique de l'environnement cartographié

Une autre manière d'optimisation du temps de calcul est d'utiliser des propriétés géométriques de la carte. La suite des algorithmes et des résultats ont comme point de départ les travaux de Carpin [2008] où une méthode d'appariement des cartes d'intérieurs des bâtiments est présentée. Celle-ci réalise une détection des lignes droites comme pour retrouver en premier une possible rotation de la carte. Des résultats de précision d'appariement et des temps de calcul de cette méthode sont également présentés. Une analyse de la figure III.19(a) du parking souterrain relève la facilité de détection des couloirs de circulation parallèles ou perpendiculaires. C'est pour cela que nous nous intéressons à l'existence de ces lignes dans la carte.

Analysant mieux notre problème, nous constatons qu'il est possible de réaliser un appariement angulaire avant de calculer la translation. Dans le cas où les rangés sont perpendiculaires, cela suffirait de déterminer une ligne droite et d'appliquer l'algorithme de calcul de la translation uniquement 4 fois (tous les 90°), mais cette hypothèse n'est pas toujours vérifiée et ne sera pas appliquée dans ces travaux.

Alignement de la carte globale

Prenons d'abord la carte globale III.24(a) pour l'aligner sur les axes XY, comme présenté dans la figure III.24(b).

Pour réaliser la superposition nous calculons d'abord la **Transformée de Hough** TH_{CarteG} ensuite le **Spectre de Hough** SH_{CarteG} de la carte initiale, $CarteG$. Les deux résultats sont observés dans la figure III.25. Dans la **Transformée de Hough** chaque ligne est un vecteur de coordonnées paramétriques (θ, ρ) où θ représente l'angle et ρ la norme du vecteur (la longueur du segment perpendiculaire à la droite d'angle θ et passant par l'origine). En transformant toutes les

1. **Open Multi-Platform** shared-memory parallel programming ;

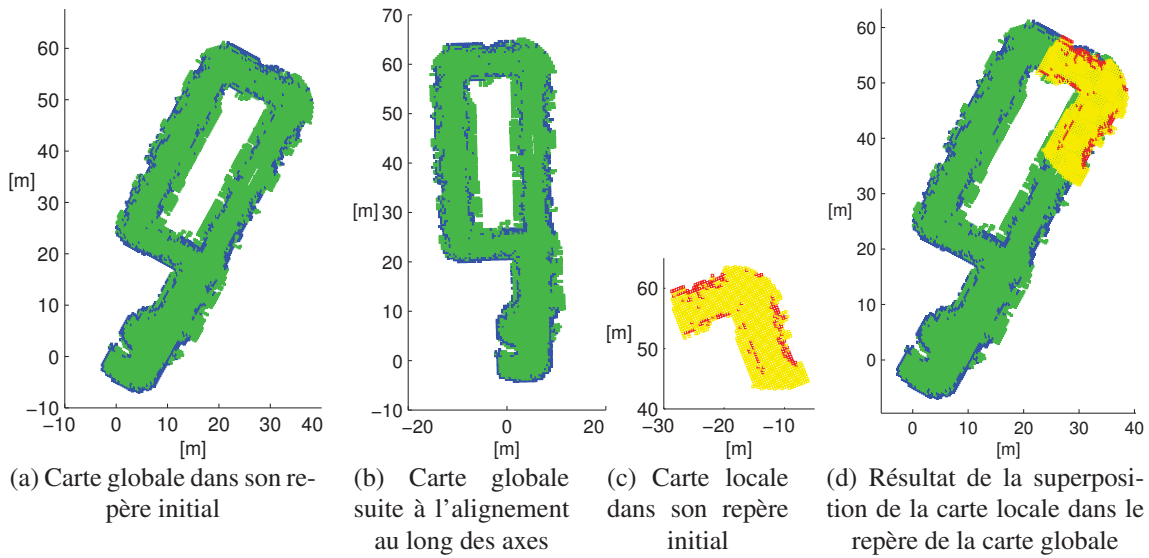


FIGURE III.24 – Appariement des cartes utilisant les propriétés de la *Transformée de Hough*. Les couleurs vert et jaune indiquent l'espace libre exploré, les couleurs bleu et rouge indiquent l'espace occupé pour la carte globale respectivement locale

lignes possibles qui passent par un point, on obtient une sinusoïde dans l'espace de Hough. Si les courbes associées à deux points se coupent, l'endroit où elles se coupent dans l'espace de Hough correspond aux paramètres d'une droite qui relie ces deux points.

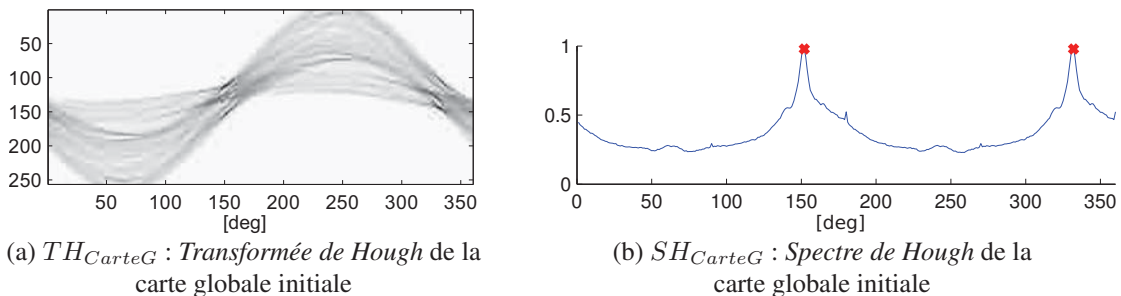


FIGURE III.25 – Calcul de la *Transformation* et du *Spectre de Hough* pour déterminer l'alignement de la carte globale initiale au long des axes X et Y

Regardons d'abord la figure III.25(a). L'abscisse correspond au paramètre θ et l'ordonnée au paramètre ρ . Le résultat du calcul du *Spectre de Hough* est exprimé par une variation de la couleur. Deux pics sont observés pour la valeur de 150° et 330° . Comme nous connaissons l'angle de rotation entre la carte initiale $CarteG$ et la carte initiale alignée $CarteGa$, nous constatons que les deux maximums obtenus pour une rotation de 150° et 330° par rapport à l'axe V sont dus à l'existence d'un long couloir de circulation dans la carte. A la suite les cartes seront alignées au long de l'angle de rotation le plus petit par rapport à un axe, dans ce cas $330^\circ = -30^\circ$.

Cet alignement de la carte globale initiale peut se faire hors ligne et une fois réalisé sera sauvegardé pour la réutilisation.

Calcul des rotations possibles de la carte locale par rapport à la carte globale utilisant la Transformée de Hough

Supposons deux cartes, une dite carte globale qui est donnée en entrée et qui ne change pas et une autre carte locale qui est enrichie en temps réel, voir figure III.24(d). Par la suite nous détaillons la démarche de calcul des angles possibles de rotation de la carte locale qui permettra de diminuer le champ de recherches et donc le temps de calcul global. Les détails de l'algorithme 7 sont donnés dans l'annexe A.5.

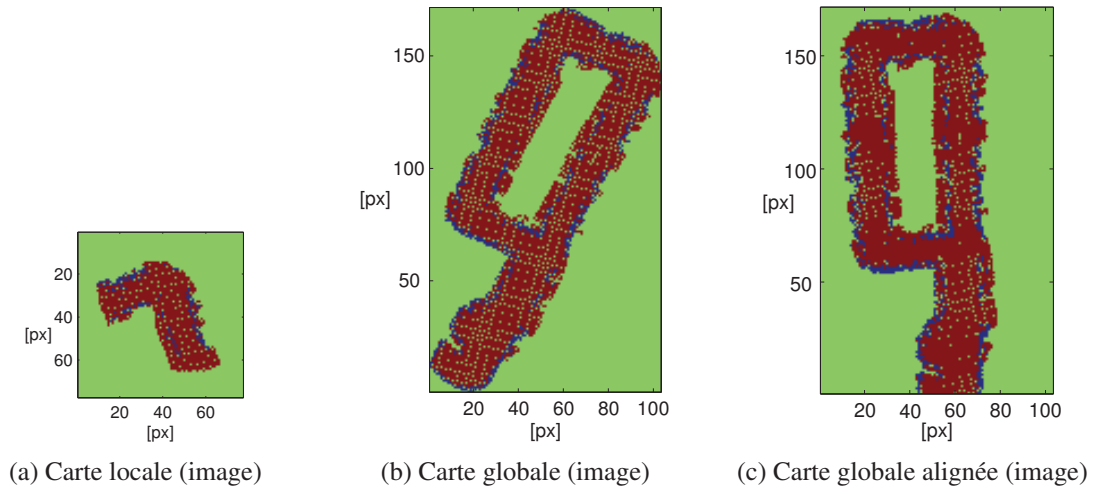


FIGURE III.26 – Les deux cartes (locale et globale) à appairer; les cartes sont deux images ayant la dimension d'un pixel équivalente à $0,4\text{ m} \times 0,4\text{ m}$

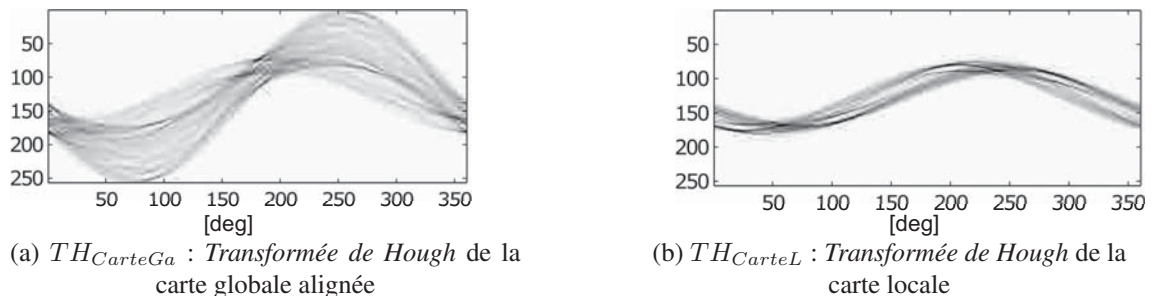


FIGURE III.27 – Calcul de la Transformée de Hough pour la carte globale alignée et la carte locale

En analysant la figure III.28(a), il est possible de valider que les valeurs maximales du $SH_{CarteGa}^{Rot}$ sont atteintes pour les angles de 0° , 180° et 360° , ce qui confirme que la carte est bien alignée au long des axes U et V . Pour le Spectre de Hough SH_{CarteL}^{Rot} les quatre plus grandes valeurs sont atteintes pour les angles multiples de 90° ce qui est bien en concordance avec la forme de la carte locale qui présente un virage à gauche à 90° dans un parking souterrain (voir figure III.28(b)).

Pour déterminer mathématiquement les plus probables quatre rotations, une corrélation croisée entre $SH_{CarteGa}^{Rot}$ et SH_{CarteL}^{Rot} est réalisée, comme présenté dans la figure III.28(c). Les premières quatre valeurs maximales de cette corrélation sont dans l'ordre décroissant : 160° , 340° , 123° et 303° . A la suite nous créons quatre cartes locales nommées $CarteL_A$, $CarteL_B$, $CarteL_C$ et $CarteL_D$ qui seront construites par la simple rotation de la $CarteL$ avec les quatre angles précédemment calculés.

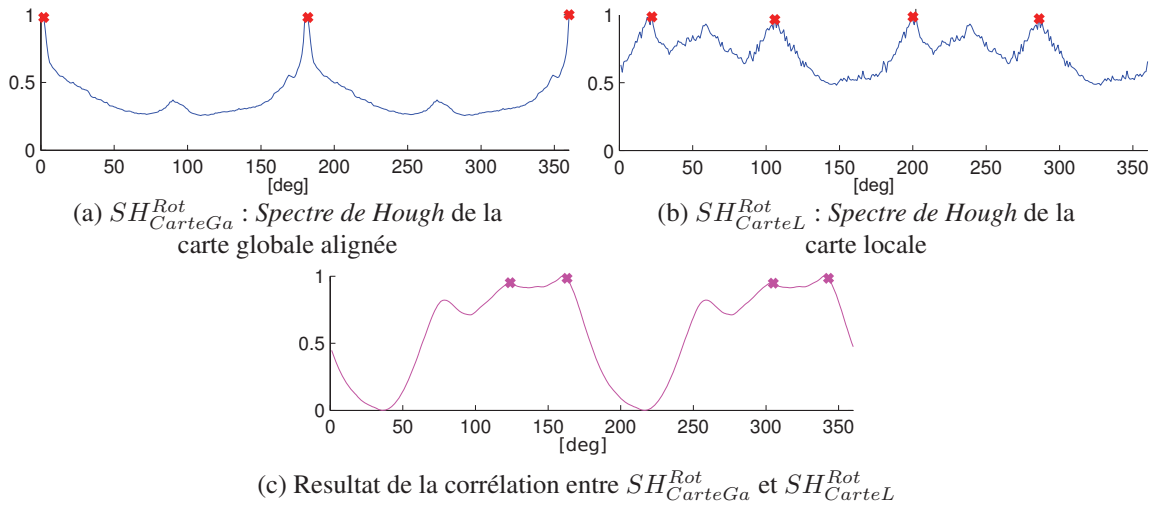
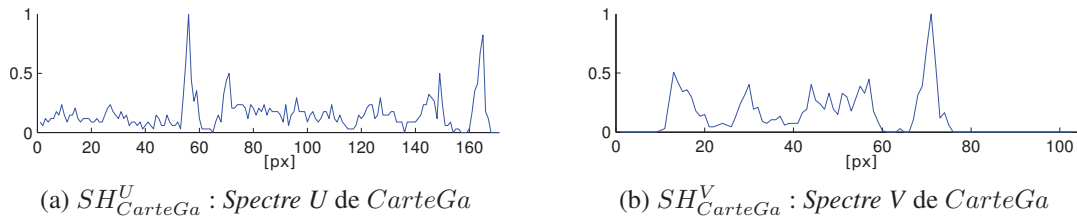


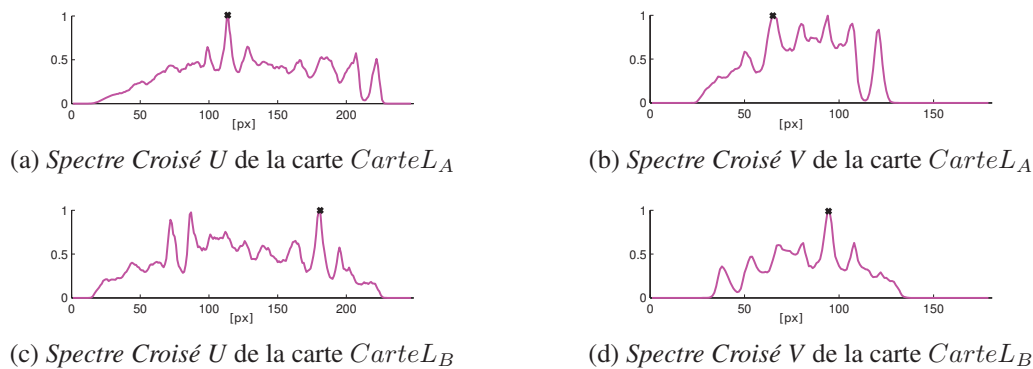
FIGURE III.28 – Recherche de directions dominantes de la carte locale

Calcul des translations possibles de la carte locale par rapport à la carte globale utilisant la Transformée de Hough

A la suite, nous déterminons pour ces nouvelles quatre cartes locales, quelles sont les quatre translations qui réalisent les meilleurs appariements avec la carte globale. Pour ce faire, les notions de *Spectre U* (noté SH_{Carte}^U) et le *Spectre V* (noté SH_{Carte}^V) sont introduites. Par *Spectre U* et *spectre V* sont désignés deux vecteurs de taille c_U respectivement c_V représentant les dimensions la carte. Ils correspondent aux projections de la carte au long de l'axe U et V. Les figures III.29(a) et (b) illustrent les deux spectres pour la carte globale alignée *CarteGa*.

FIGURE III.29 – Spectre U et Spectre V pour la carte globale alignée *CarteGa* de dimensions $103px \times 171px$

Les figures III.30(a) à (h) indiquent les résultats de la corrélation entre les *Spectre U* ou *V* de la *CarteGa* avec les *Spectres U* ou *V* des quatre cartes locales. Cette corrélation est nommée *Spectre Croisé U* ou *V* des cartes A, B, C respectivement D.



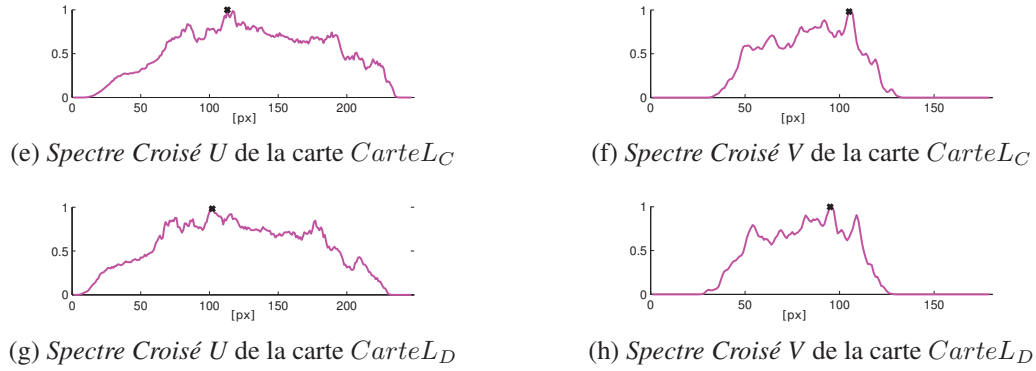


FIGURE III.30 – Corrélation entre le Spectre U de la $CarteGa$ et les quatre Spectres U des quatre cartes locales (figures (a), (c), (e) et (g)). L'opération est répétée également pour l'axe V (figures (b), (d), (f) et (h))

Les déplacements de la carte locale, Δ_U et Δ_V sont calculés suivant l'équation (III.45) :

$$\begin{cases} \Delta_U = [\dim(\text{SpectreCroisé}_U^{CarteL}) + 1 - \dim(SH_{CarteGa}^U)] - \text{argmax}(\text{SpectreCroisé}_U^{CarteL}) \\ \Delta_V = [\dim(\text{SpectreCroisé}_V^{CarteL}) + 1 - \dim(SH_{CarteGa}^V)] - \text{argmax}(\text{SpectreCroisé}_V^{CarteL}) \end{cases} \quad (\text{III.45})$$

$$\begin{cases} \Delta_U = cst_U - \text{argmax}(\text{SpectreCroisé}_U^{CarteL}) \\ \Delta_V = cst_V - \text{argmax}(\text{SpectreCroisé}_V^{CarteL}) \end{cases} \quad (\text{III.46})$$

Pour la situation présentée, les résultats sont détaillés dans l'équation (III.47) :

$$\begin{cases} \Delta_U = [247\text{px} + 1\text{px} - 171\text{px}] - \text{argmax}(\text{SpectreCroisé}_U^{CarteL}) \\ \Delta_V = [179\text{px} + 1\text{px} - 103\text{px}] - \text{argmax}(\text{SpectreCroisé}_V^{CarteL}) \end{cases} \quad (\text{III.47})$$

où $\dim(\text{SpectreCroisé}_U^{CarteL}) = 247$ et $\dim(SH_{CarteGa}^U) = 171$. Cela nous conduit aux résultats présentés dans le tableau III.3 où $\text{argmax}(\text{SpectreCroisé}_{CarteL}^{U/V})$ sont déduits dans les graphiques des spectres croisés présentés dans les figures III.30(a) à (h).

Roto-translation de la carte $CarteL_X$ dans le repère $CarteG$			
	$\Delta_U =$	$\Delta_V =$	Angle rot. =
$CarteL_A$	$cst_U - 114\text{px} = -37\text{px}$	$cst_V - 65\text{px} = 12\text{px}$	$-160^\circ + 30^\circ = -130^\circ$
$CarteL_B$	$cst_U - 180\text{px} = -103\text{px}$	$cst_V - 94\text{px} = -17\text{px}$	$-340^\circ + 30^\circ = 50^\circ$
$CarteL_C$	$cst_U - 113\text{px} = -36\text{px}$	$cst_V - 106\text{px} = -29\text{px}$	$-123^\circ + 30^\circ = -93^\circ$
$CarteL_D$	$cst_U - 101\text{px} = -24\text{px}$	$cst_V - 96\text{px} = -19\text{px}$	$-303^\circ + 30^\circ = 87^\circ$

TABLE III.3 – Résultats de la méthode d'appariement des cartes avec *Transformée de Hough*

Les figures III.31(a) à (d) illustrent les cartes locales après les roto-translations calculées et affichées dans le tableau III.3.

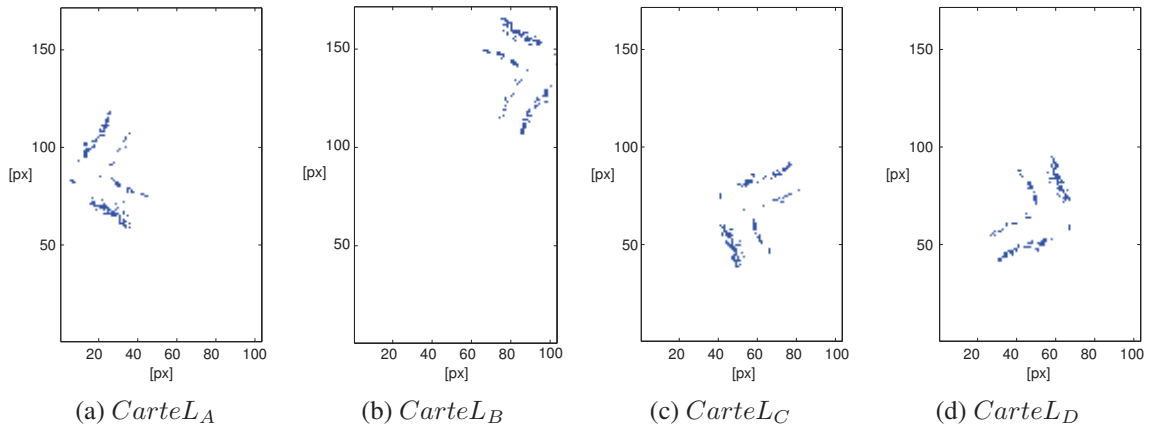


FIGURE III.31 – Cartes locales après les roto-translations illustrés dans le tableau III.3

Enfin, les figures III.32(a) à (d) permettent de visualiser le résultat obtenu par superposition des cartes.

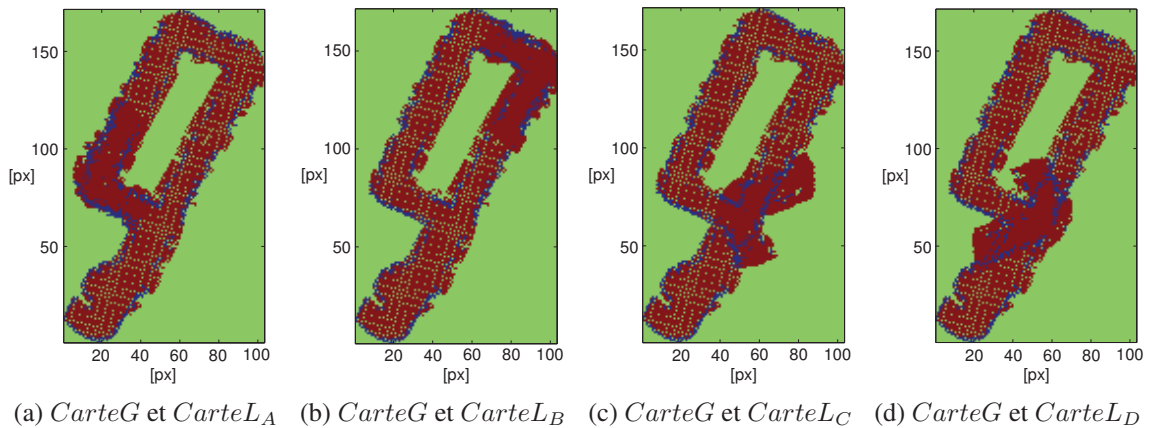


FIGURE III.32 – Résultat de la superposition de la carte globale et des cartes locales estimées

III.3.3.5 Résultats et analyse du temps de calcul

En final, nous pouvons affirmer que nous disposons d'une technique qui nous permet d'apparier des cartes par une technique de corrélation [§III.3.3.2]. Cela revient à déterminer la position du véhicule lors d'un deuxième passage dans la carte enregistrée dans l'étape d'apprentissage.

Dans la figure III.34 nous proposons cinq versions d'algorithme que nous détaillerons par la suite.

Version 1 : 13,6s - suppose de ne pas utiliser des a priori et de tester tout l'espace de recherche. Pour la carte globale de taille $60 \text{ px} \times 160 \text{ px}$, il est nécessaire de réaliser 9 600 corrélations entre la carte locale et celle globale pour une seule rotation. Pour couvrir tout l'espace il faut répéter l'opération pour les 360° à une résolution d'au moins 1° . Dans ce cas le temps de calcul s'élève à plus de 13 secondes ce qui rend l'algorithme inutilisable dans cette forme.

Les versions 2 à 5 supposent d'utiliser des a priori sur la position (sous forme d'un rayon de recherche R) et l'orientation du véhicule fournies par un second algorithme.

Version 2 : 372 ms - pour un rayon de localisation de 4 m et une plage de variation de l'orientation de $[-40^\circ, 40^\circ]$ avec une résolution de 1° , la deuxième version est approximativement 35 fois plus rapide que la première version.

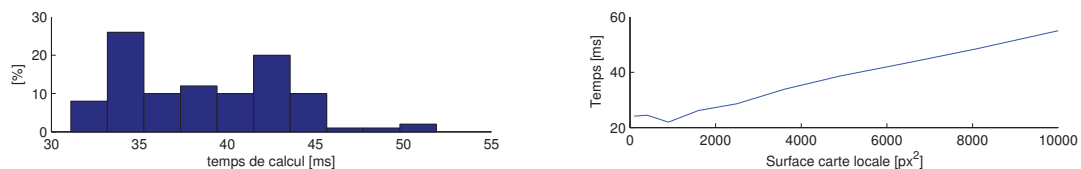
Version 3 : 66,5 ms - nous avons montré que dans des situations d'environnement très structuré, comme l'environnement intérieur, il est possible d'utiliser la méthode de Hough [§III.3.3.4] pour extraire les orientations les plus probables dans lesquelles le véhicule peut se trouver en fonction de l'orientation des murs. Enfin, pour chaque orientation il est possible d'utiliser le spectre de Hough pour retrouver la translation. Comme cette technique est moins précise que celle par corrélation, une étape supplémentaire est ajoutée pour affiner le résultat. L'algorithme 7 est détaillé dans l'annexe A.5.

La première étape a pour but de proposer des roto-translations nécessaires à appliquer à une carte pour la superposer à une deuxième, tout en exploitant la particularité de l'existence des lignes droites dans l'environnement. Son temps de calcul moyen estimé sur le même ordinateur est d'environ **35ms** pour le cas présenté, voir figure III.24(d).

La deuxième étape utilise en entrée la meilleure position précédemment calculée et affine le résultat utilisant l'approche par corrélation. En reprenant les résultats du tableau III.2, pour un rayon de recherche de $R = 2$ m, le temps de calcul moyen est décrit par l'équation (III.48).

$$3,5 \text{ ms} \cdot \frac{\text{nombre de degrés}}{\text{résolution angulaire}} \cdot \text{nombre de cartes locales} \quad (\text{III.48})$$

De manière expérimentale, nous constatons qu'il faudrait tester sur une plage de $[-2^\circ, 2^\circ]$ avec une résolution angulaire imposée à $0,5^\circ$ le calcul revient à : 9 (angles à tester) $\times 3,5$ ms \times nombre des cartes (voir Version 3 dans la figure III.34).



(a) L'histogramme du temps de calcul de l'étape I de l'algorithme 7 pour une carte locale de dimensions $70\text{px} \times 70\text{px}$ et une carte globale de dimensions $60\text{px} \times 160\text{px}$ (b) L'influence de la surface en pixels ($\text{px} \times \text{px}$) de la carte locale sur le temps de calcul de l'algorithme 7

FIGURE III.33 – Analyse du temps de calcul de l'étape I de l'algorithme 7 (voir annexe A.5)

Version 4 et 5 : 18 ms à 31,5 ms - cette stratégie permet à la fois de réaliser une localisation globale avec le coût de calcul associé mais également une localisation continue avec a priori. Une fois le véhicule localisé, il est possible d'utiliser une contrainte forte de localisation et orientation donnée par l'approche odométrique. Sachant que l'odométrie est très précise, nous pouvons limiter l'espace de recherche sur une plage d'orientation de $[-2^\circ, 2^\circ]$ avec une résolution angulaire imposée à $0,5^\circ$. Si le rayon de recherche est de 2 m, alors la version 4 (couleur verte dans la figure III.34) correspond exactement à la deuxième étape de l'algorithme appliqué dans la version 3. Si le rayon de recherche R diminue, il est possible de gagner encore du temps de calcul. Ainsi pour $R = 0,8$ m le temps s'élève à 18 ms.

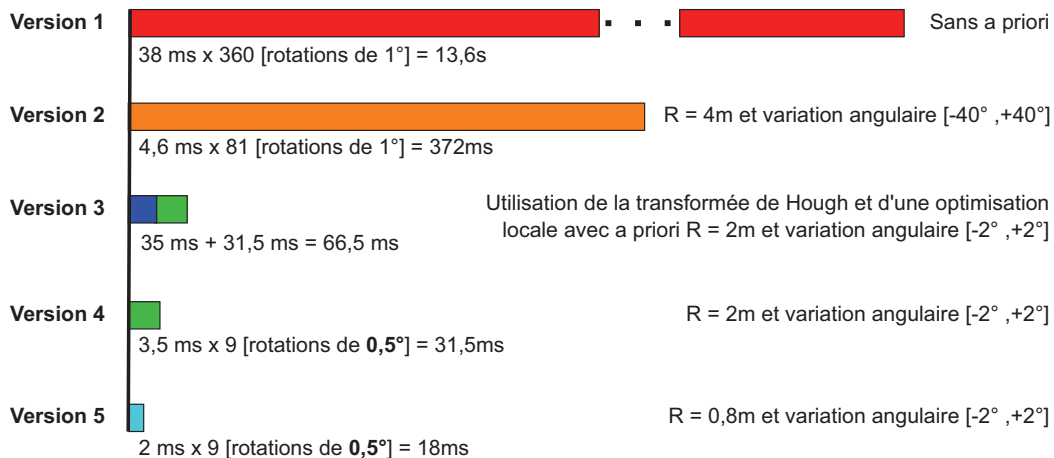


FIGURE III.34 – Comparaison du temps de calcul pour les approches d'appariement des cartes d'obstacles avec ou sans utilisation des a priori sur la position et l'orientation du véhicule ou sur la structure géométrique de l'environnement

En plus de la localisation initiale il est possible d'utiliser la même technique pour une vérification périodique à basse fréquence de la concordance entre la position du véhicule estimée en continu et la position globale.

Par la suite, nous abordons la dernière technique de localisation qui nous intéresse pour notre système de voiturier autonome : la localisation par techniques de comparaison d'images. Il s'avère qu'il est important d'explorer cette piste car ces techniques sont très utilisées dans la robotique pour mettre en corrélation la vue actuelle avec une base d'images déjà apprises. En connaissant les positions des images déjà apprises, retrouver l'image qui ressemble à la vue courante revient à retrouver la plus proche position connue dans le repère de la trajectoire de référence. Par conséquent il est possible d'alimenter l'algorithme d'appariement des cartes par des a priori de position.

III.3.4 Estimation des a priori de localisation et d'orientation par techniques de comparaison d'images

Dans la section précédente nous avons montré qu'il est possible de localiser le véhicule en utilisant une technique d'appariement des cartes. Nous avons montré que la méthode peut être utilisée comme approche de localisation globale ou continue en fonction de l'existence des a priori de localisation. Dans cette section nous nous intéressons à des techniques de localisation par comparaison d'images nous permettant de retrouver les a priori de localisation dont on a besoin.

Dans l'état de l'art des techniques de localisation nous avons introduit la problématique de fermeture de boucle. Ce principe a été illustré dans l'état de l'art dans la figure III.4 la section §III.2.2.3. Ces techniques d'appariement d'images par ressemblance sont essentiellement basées sur l'extraction d'éléments représentatifs dans l'image, procédé génériquement nommés : dictionnaire de mots ou sac-de-mots (BoW : Bag-of-Word ou BoV : Bag-of-Visterms, en anglais) [Quelhas and Odobez, 2007; Yang et al., 2007; Mikulik et al., 2013; Peng et al., 2016] et sont issues des techniques d'apprentissage non supervisées. Dans la vision par ordinateur, le modèle de sac-de-mots a été initialement appliqué à la classification. Cela revient à retrouver des éléments de ressemblances qui permettent de regrouper des images selon certaines caractéristiques nommées mots-visuels.

Pour notre système de voiture autonome, pendant l'étape d'apprentissage nous réalisons une acquisition d'images tout au long de la trajectoire effectuée par le conducteur. A chaque image nous associons une pose de la caméra calculée utilisant l'odométrie du véhicule. Lors d'un deuxième passage nous réalisons un appariement de la vue courante avec la base d'images pour retrouver l'image qui ressemble le plus selon un critère de distance donné. Par conséquent il est possible de retrouver une position approximative du véhicule dans le repère initial.

Par la suite nous détaillerons les trois étapes que nous utilisons également dans notre application de voiture autonome. Nous verrons dans la section §III.3.4.4 qu'il est possible de fusionner l'étape de la construction du dictionnaire des mots-visuels et l'étape d'enseignement dans une seule et même étape.

III.3.4.1 Étape de construction du dictionnaire de mots-visuels

Pour mettre en place cette technique, il est nécessaire de revenir aux structures élémentaires utilisées dans les techniques de traitement d'images : les points d'intérêt et leurs descripteurs. Nous rappelons qu'il y a une très grande variété de détecteurs de points ou de zone d'intérêt dans les images. Conjointement à l'utilisation d'un détecteur de point d'intérêt, il faut utiliser des descripteurs comme : SIFT, SURF ou bien d'autres comme celui proposé par [Geiger et al., 2010, 2011] et que nous avons utilisé dans l'algorithme de détection d'obstacles par une approche SfM [§IV.4].

La construction du dictionnaire des mots-visuels (ou des sacs de mots, Bag of Words) consiste à réaliser un découpage de l'espace des descripteurs en fonction d'un critère de distance, voir figure III.35. Cette étape peut être effectuée en utilisant un algorithme généralisé de regroupement (clustering en anglais) comme l'algorithme K-moyennes (K-Means en anglais) [MacQueen et al., 1967] ou l'algorithme d'Espérance Maximisation (EM : Expectation–Maximization en anglais) conjointement à un découpage de l'espace utilisant le diagramme de Voronoï ou bien un algorithme de partitionnement du type KD-Tree [Bentley, 1975; Zhou et al., 2008]. Cette technique a déjà été utilisée dans l'approche de BoW par J. Sivic et A. Zisserman [Sivic and Zisserman, 2003] ou bien par D. Nister et H. Stewenius [Nister and Stewenius, 2006; Yang et al., 2009] qui ont développé une structure arborescente de vocabulaire basée sur l'algorithme *hierarchical k-means clustering* [Jain, 2010].

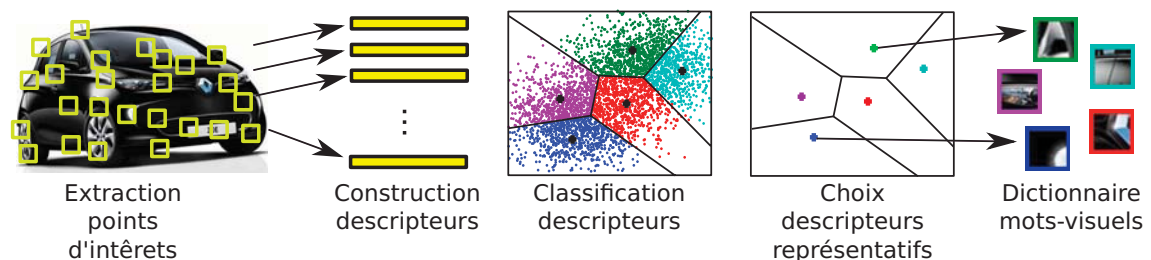


FIGURE III.35 – Schématisation du processus de création du dictionnaire de mots-visuels

III.3.4.2 Étape d'entraînement (cartographie)

Étant donné un vocabulaire de mots-visuels, l'étape suivante est de capturer la cooccurrence statistique de mots-visuels. Dans les techniques d'apprentissage automatique (machine learning,

en anglais) des structures arborescentes sont fréquemment utilisées. Par exemple pour l'approximation des distributions de probabilités de très grandes dimensions, il est possible d'utiliser un arbre où les valeurs des arrêts représentent les probabilités jointes à des variables aléatoires. Dans ce sens nous citons l'arbre Chow-Liu, un type de réseau bayésien qui propose de minimiser la divergence entre la probabilité réelle et celle approximée [Chow and Liu, 1968; El Fkihi, 2008].

Soit $\mathbb{P}(Z)$ la loi jointe à n variable discrètes $Z = \{z_1, z_2, \dots, z_n\}$. L'utilisation directe de la loi jointe $\mathbb{P}(Z)$ présente les inconvénients suivants : la mémoire occupée dans le cas binaire est de l'ordre de 2^n et le temps de traitement évolue suivant une loi d'ordre $O(2^n)$ donnée par la complexité de l'évolution de l'inférence. Dans ce contexte, si $\mathbb{P}(Z)$ est une distribution sans aucune structure spécifique, alors il est possible de l'approximer par une autre distribution $\mathbb{Q}(Z)$ ayant une représentation spécifique, par exemple un arbre, qui lui confère certains avantages à l'utilisation. Pour réaliser cette approximation il est nécessaire d'utiliser une mesure de distance entre deux probabilités. Une solution est d'utiliser une mesure informationnelle appelée divergence de Kullbak-Leibler, ou entropie croisée ou entropie relative. Alors la divergence de Kullbak-Leibler entre deux probabilités \mathbb{P} et \mathbb{Q} est exprimé comme suit :

$$D_{KL}(\mathbb{P}, \mathbb{Q}) = \sum_{x \in X} \mathbb{P}(x) \log \frac{\mathbb{P}(x)}{\mathbb{Q}(x)} \quad (\text{III.49})$$

En revenant sur notre problème de BoW, calculer la ressemblance entre deux descripteurs peut se réaliser simplement en utilisant la norme L_2 comme nous l'avons déjà présenté dans le cas d'appariement des points d'intérêt dans plusieurs images successives. Toutefois l'utilisation de ce type de métrique suppose une indépendance entre les mots-visuels, ce qui, en réalité n'est pas le cas. Les objets réels sont composés de plusieurs mots-visuels et cette liaison d'appartenance à un objet doit être prise en compte. Cela se réalise par l'utilisation de l'arbre Chow-Liu qui peut être employé comme un outil nous permettant de capturer la corrélation entre différents mots-visuels indiquant que le monde est composé des objets plutôt que de mots-visuels indépendants.

III.3.4.3 Étape de localisation - utilisation d'un modèle de navigation probabiliste

Fast Appearance Based Mapping ou FAB-MAP introduit par M. Cummins est une technique de navigation et cartographie qui n'utilise que la comparaison d'images sans utiliser aucune information métrique [Cummins and Newman, 2008; Paul and Newman, 2010; Cummins and Newman, 2011; Glover et al., 2012; Milford and Wyeth, 2012]. Ils utilisent un modèle probabiliste qui combine une information de ressemblance entre deux scènes avec une information d'originalité. Ainsi la probabilité de localisation sera faible pour les données qui sont très ressemblantes mais communes et sera forte pour les données ressemblantes et originales.

Notons par $|v|$ la taille du vocabulaire de mots-visuels. Notons également par $Z_k = \{z_1, z_2, \dots, z_{|v|}\}$ une observation de l'apparence d'une scène locale au moment k où z_q est une variable binaire indiquant la présence ($z_q = 1$) ou l'absence ($z_q = 0$) du $q^{\text{ème}}$ mot-visuel du vocabulaire. Par la suite \mathfrak{Z}^k est utilisé pour indiquer l'ensemble de toutes les observations jusqu'à l'instant k . Cela se traduit par $\mathfrak{Z}^k = \{Z_1, Z_2, \dots, Z_k\}$, voir figure III.36.

Introduisons une nouvelle notion : $\mathfrak{L}^k = \{L_1, \dots, L_{n_k}\}$ étant l'ensemble de n_k poses (*location* en anglais) distinctes qui correspondent aux images observées jusqu'à l'instant k . Utilisant un détecteur, nous observons (avec un certain bruit) un mot-visuel z_q qui correspond à un élément e_q de la scène. Cela se traduit par un modèle d'appartenance à une position de la carte sous forme d'une probabilité de visualisation de tous les éléments e_q de la scène depuis une position L_i et

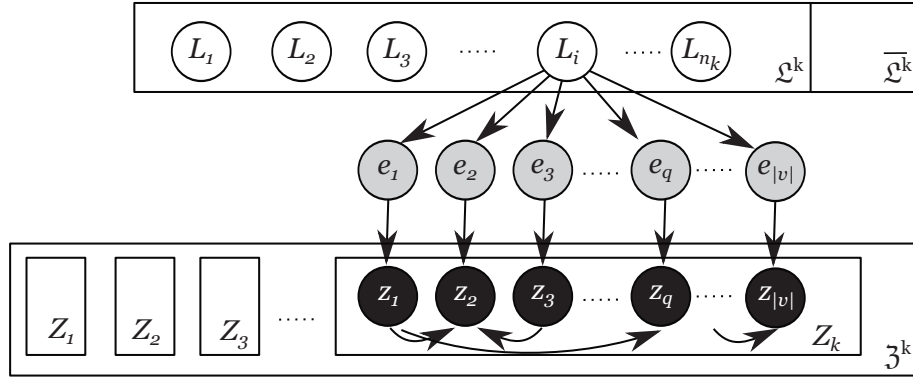


FIGURE III.36 – Schéma de principe du modèle bayésien proposé par FAB-MAP

s'exprime sous la forme suivante :

$$L_i : \{p(e_1 = 1|L_i), \dots, p(e_{|v|} = 1|L_i)\} \quad (\text{III.50})$$

où chaque élément e_q est généré indépendamment de la position. Si nous désignons \mathcal{D} un détecteur qui relie l'élément de la scène e_q du mot-visuel z_q alors le détecteur est spécifié ainsi :

$$\mathcal{D} : \begin{cases} p(z_q == 1|e_q = 0), & \text{probabilité de faux positif} \\ p(z_q == 0|e_q = 1), & \text{probabilité de faux négatif} \end{cases} \quad (\text{III.51})$$

Dans les travaux [Cummins and Newman, 2008, 2011] l'approche récursive suivante est présentée : supposons qu'un certain nombre d'images ont déjà été observées et cartographiées. Lors de la réception d'une nouvelle image, nous souhaitons calculer les probabilités que cette image ressemble aux autres observations déjà collectées. Dans ce cas, calculer la probabilité $p(L_i|\mathfrak{Z}^k)$ pour chaque position L_i s'écrit sous une forme conditionnelle et de manière récursive par l'équation de Bayes :

$$p(L_i|\mathfrak{Z}^k) = \frac{p(Z_k|L_i, \mathfrak{Z}^{k-1}) p(L_i|\mathfrak{Z}^{k-1})}{p(Z_k|\mathfrak{Z}^{k-1})} \quad (\text{III.52})$$

où $p(L_i, \mathfrak{Z}^{k-1})$ définit la probabilité a priori sur la position (prior belief, en anglais), $p(Z_k|L_i, \mathfrak{Z}^{k-1})$ représente la probabilité d'observation ou la vraisemblance (observation likelihood, en anglais) et le dénominateur $p(Z_k|\mathfrak{Z}^{k-1})$ est un terme de normalisation.

La vraisemblance : en supposant l'indépendance entre une observation courante et une autre antérieure, le terme $p(Z_k|L_i, \mathfrak{Z}^{k-1})$ de l'équation (III.52) s'exprime ainsi :

$$p(Z_k|L_i) = p(z_n|z_1, z_2, \dots, z_{n-1}, L_i) p(z_{n-1}|z_1, z_2, \dots, z_{n-2}, L_i) \dots p(z_2|z_1, L_i) \quad (\text{III.53})$$

qui peut être approximé sous une forme simplifiée grâce au modèle bayésien naïf ainsi :

$$p(Z_k|L_i) \approx \prod_{q=1}^{|v|} p(z_q|L_i) \quad (\text{III.54})$$

Il est également possible d'utiliser une approximation par arbre Chow-Liu qui tient compte de certaines dépendances entre les mots-visuels observés et qui s'exprime sous la forme :

$$p(Z_k|L_i) \approx p(z_r|L_i) \prod_{q=2}^{|v|} p(z_q|z_{p_q}, L_i) \quad (\text{III.55})$$

où dans l'arbre de Chow Liw z_r est le nœud racine et z_{p_q} est le nœud parent de z_q . C'est cette représentation qui sera utilisée par la suite et conformément au réseau bayésien illustré dans le

schéma de la figure III.36 le terme $p(z_q|z_{p_q}, L_i)$ s'exprime ainsi :

$$p(z_q|z_{p_q}, L_i) = \sum_{s_{e_q} \in \{0,1\}} p(z_q|e_q = s_{e_q}, z_{p_q}, L_i) p(e_q = s_{e_q}|z_{p_q}, L_i) \quad (\text{III.56})$$

L'a priori de localisation : le terme $p(L_i|\mathfrak{Z}^k)$ de l'équation (III.52) définissant l'a priori sur la position est obtenu par l'utilisation d'une hypothèse de voisinage : si le véhicule était dans la position L_i à l'instant k , alors il est très probable qu'il se situe dans une position L_{i-1} ou L_{i+1} à l'instant $k+1$. Dans les situations où il n'y a pas de positions voisines connues, alors une partie de la probabilité sera assignée à une nouvelle position L_{new} , le reste de probabilité étant réparti aux autres places dans l'ensemble \mathfrak{L}^k .

Le terme de normalisation : mesure de l'originalité de l'observation et directement lié à la problématique de *perceptual aliasing*. Si la carte finale est divisée en deux termes, un qui décrit l'information liée à l'environnement \mathfrak{L}^k déjà cartographié jusqu'à l'instant k et un autre lié à l'information pas encore cartographiée $\overline{\mathfrak{L}^k}$, alors le terme de normalisation de l'équation (III.52) s'écrit sous la forme suivante :

$$p(Z_k|\mathfrak{Z}^{k-1}) = \sum_{m \in \mathfrak{L}^k} p(Z_k|L_m) p(L_m|\mathfrak{Z}^{k-1}) + \sum_{u \in \overline{\mathfrak{L}^k}} p(Z_k|L_u) p(L_u|\mathfrak{Z}^{k-1}) \quad (\text{III.57})$$

Le deuxième terme de l'équation (III.57) ne peut pas être évalué directement car il suppose de connaître de l'information sur des positions pas encore cartographiées. Toutefois, il est possible de l'approximer utilisant une approche d'échantillonnage par méthode Monte-Carlo car nous supposons disposer d'un jeu de données d'apprentissage suffisamment grand enregistré lors d'une étape antérieure et qui est composé de données de position acquises d'une manière aléatoire. Dans ce cas l'équation (III.57) se réécrit ainsi :

$$p(Z_k|\mathfrak{Z}^{k-1}) \approx \sum_{m \in \mathfrak{L}^k} p(Z_k|L_m) p(L_m|\mathfrak{Z}^{k-1}) + p(L_{new}|\mathfrak{Z}^{k-1}) \sum_{u=1}^{n_s} \frac{p(Z_k|L_u)}{n_s} \quad (\text{III.58})$$

avec n_s la taille du jeu de données et $p(L_{new}|\mathfrak{Z}^{k-1})$ la probabilité a priori d'être à une nouvelle position.

La mise à jour du modèle : Étant donné une observation associée à une position, le modèle d'apparence peut être mis à jour pour chaque élément e_q comme suit :

$$p(e_q = 1|L_i, \mathfrak{Z}^k) = \frac{p(Z_k|e_q = 1, L_i) p(e_q = 1|L_i, \mathfrak{Z}^{k-1})}{p(Z_k|L_i)} \quad (\text{III.59})$$

III.3.4.4 Résultats de l'algorithme d'appariement d'images

Dans leurs travaux, [Cummins and Newman, 2011] présentent des résultats de l'algorithme FAB-MAP pour des trajectoires de 70 km ou bien 1000 km où les images ont été capturées par un système multi-caméras à champ recouvrant. Ils présentent également une solution partant de 2 800 images prises tous les 10 m dans l'environnement urbain, soit une longueur totale de 28 km de trajectoire. De ce jeu de données, les points d'intérêt et leurs descripteurs ont été extraits et un vocabulaire de 11 000 mots-visuels a été réalisé après l'étape de segmentation et regroupement. Les auteurs annoncent que la création du vocabulaire nécessite 2 heures de calcul sur un Pentium IV - 3 GHz.

Mais dans notre application de parking voiturier, nous souhaitons que l'algorithme donne de bons résultats pour des trajectoires relativement petites inférieures à 500 m. La difficulté du problème réside dans la très forte ressemblance des images prises très proches l'une de l'autre.

Initialement l'algorithme est conçu pour fonctionner avec un vocabulaire de mots-visuels issus des images prises d'un autre environnement que l'environnement d'évolution du véhicule. Cela rend l'algorithme générique. Pour réaliser notre objectif nous détournons cet usage et nous nous basons sur un effet de sur-apprentissage de l'algorithme pour améliorer sa qualité de détection. Rappelons que l'algorithme est composé de trois étapes [§III.2.2.3] : étape de création du dictionnaire, étape d'entraînement (cartographie) et étape de test (localisation). En fusionnant les premières deux étapes, les données du dictionnaire seront spécifiques à l'environnement où le véhicule évolue. Pour les jeux de données que nous détaillons par la suite le temps cumulé de ces deux étapes est d'environ 3–4 minutes. Effectivement ce n'est pas une étape qui se réalise en temps réel mais c'est un temps de calcul de post-traitement acceptable pour notre application dans une phase de mise en service du voiturier autonome.

Prenons deux jeux de données composés d'images prises au long de deux trajectoires du véhicule évoluant dans le même environnement lors de deux campagnes d'enregistrement. Le premier compose la trajectoire de référence et les images sont prises tous approximativement les 0,25 m ($\epsilon \approx 0,25$ m dans figure III.38). Le deuxième jeu de données est composé d'images prises à une fréquence de 5 Hz pour un véhicule roulant à une vitesse inférieure à 2 m/s.

Nous nous intéressons d'abord à détecter si les images dans la trajectoire de référence sont suffisamment distinctes. En utilisant comme images de la deuxième trajectoire exactement les mêmes images du premier passage, l'algorithme réalise un taux de réussite d'appariement de 99,85%. Ce résultat nous indique uniquement que nous disposons d'un algorithme performant capable de retrouver des configurations identiques de mots-visuels appartenant à une image dans un ensemble d'images.

Nous menons plus loin l'analyse pour qualifier les résultats de l'algorithme en utilisant des images très proches des images de référence. Pour cela nous réalisons l'essai suivant. Nous prenons comme trajectoire référence les mêmes images écartées approximativement de 0,25 m. Comme images pour le deuxième passage, nous prenons des images acquises toujours au long de la même trajectoire mais moins espacées.

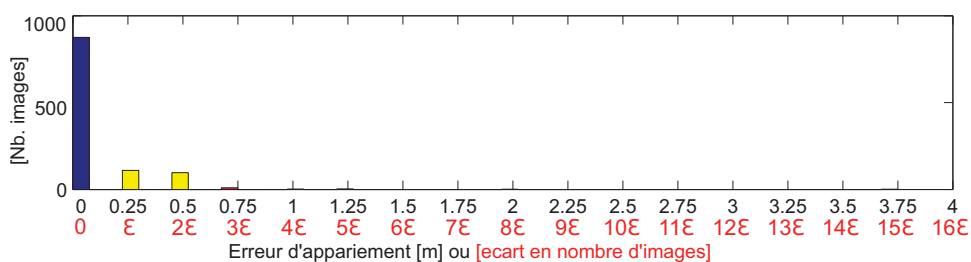


FIGURE III.37 – Histogramme : nombre d'images appariées vs. erreur d'estimation ; les conditions de test : trajectoire de référence composée d'images écartées à environ 0,25 m et trajectoire de test composée d'images situées sur la même trajectoire mais moins espacées

Prenons ϵ tel que défini dans la figure III.38 mais avec la particularité que les deux trajectoires sont superposées. Alors l'histogramme d'erreurs d'estimation III.37 s'interprète ainsi :

- 81% des images ont bien été appariées ;
- 10% des images ont été appariées avec une erreur égale $\pm 1\epsilon$ ce qui correspond à un appariement avec l'image voisine à l'image correspondante ;
- 8% des images avec une erreur = $\pm 2\epsilon$ (erreur de deux images) ;
- 0,5% des images avec une erreur = $\pm 3\epsilon$ (erreur de trois images) ;

- 0,5% des images ont été appariées avec des images situées au-delà de 10ϵ mais avec une probabilité de correspondance très faible ($\approx 10^{-3}$)

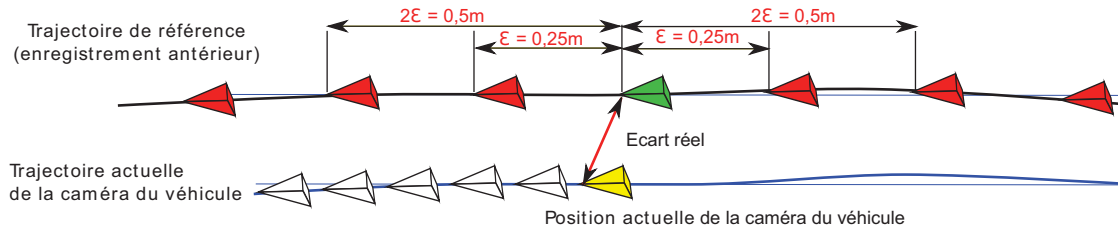


FIGURE III.38

Dans le troisième essai nous prenons une situation réelle. En première trajectoire nous gardons les images prises tous les 0,25 m des essais antérieurs mais, cette fois-ci, la deuxième trajectoire correspond à un vrai deuxième passage du véhicule dans le même couloir de circulation. Dans les figures III.40(a) à (d) nous présentons ces deux trajectoires superposées dans la même carte. L'écart réel du centre de l'essieu arrière du véhicule entre les deux trajectoires varie entre 0 m et ± 2 m. Pour ces conditions de test les résultats sont présentés dans l'histogramme de la figure III.39. Celui-ci nous indique que :

- 69,9% des images ont été appariées avec une erreur $\leq 2\epsilon$
- 87,7% des images ont été appariées avec une erreur $\leq 4\epsilon$
- 94,2% des images ont été appariées avec une erreur $\leq 8\epsilon$
- 97,1% des images ont été appariées avec une erreur $\leq 16\epsilon$

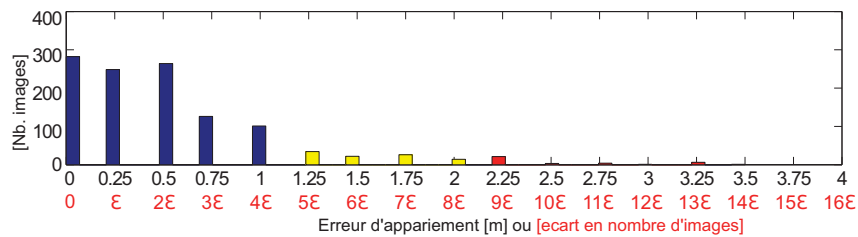


FIGURE III.39 – Histogramme : nombre d'images appariées vs. erreur d'estimation ; les conditions de test : trajectoire de référence composée d'images écartées à environ 0,25 m et trajectoire de test acquise dans le même couloir de circulation avec un écart réel inférieur à 2 m par rapport à la trajectoire de référence

Les figures III.40(a) à (c) nous indiquent par des segments la correspondance entre la position du véhicule lors du deuxième passage (symboles «*» bleus) et la position calculée suivant l'algorithme d'appariement dans la trajectoire de référence (symboles «*» noirs). La couleur du segment indique suivant une échelle de rouge à vert la probabilité d'appariement (de 0 à 1) obtenue par l'algorithme.

Suivant l'histogramme présenté de la figure III.39 environ 5,8% des images ont été appariées avec une erreur de ± 8 images ce qui correspond à ± 2 m dans notre situation. Les 5,8% correspondent à 85 mauvais appariements. Parmi ces mauvais appariements, il y en a 13 qui correspondent aux segments rouges dans la figure III.40(a) et qui ont une probabilité de bon appariement inférieure à 0,49. Comme l'algorithme nous retourne pour un bon appariement une probabilité supérieure à 0,9, alors nous pouvons prendre la décision de considérer comme des outliers les appariements avec une probabilité inférieure à un seuil.

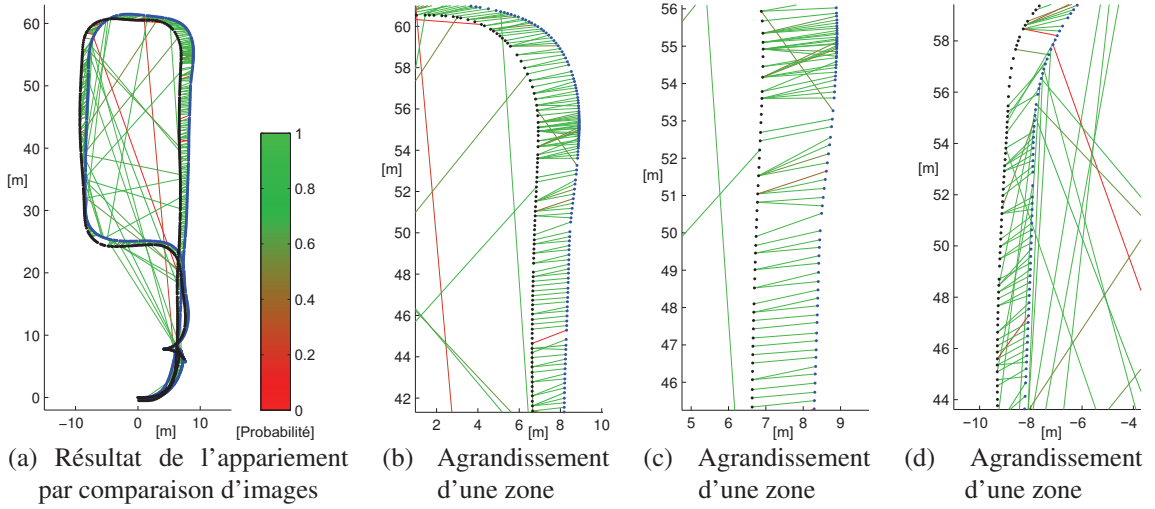


FIGURE III.40 – Résultat de l'appariement des images acquises lors de deux campagnes d'enregistrement ; les points noirs correspondent aux positions du véhicule lors de l'enregistrement des images pendant le premier passage et les points bleus respectivement au deuxième enregistrement ; les segments indiquent le résultat de correspondance de l'appariement et leurs couleurs suivant l'échelle de rouge à vert indiquent les probabilités d'appariement obtenues

Filtrage de la meilleur estimation par un modèle d'évolution

Malheureusement il reste beaucoup d'autres mauvais appariements. Après une analyse, nous nous rendons compte que ses situations correspondent à des images qui présentent beaucoup d'éléments en commun. Par conséquent l'algorithme réalise exactement le travail qu'il est censé faire et retourne des images avec une grande probabilité de ressemblance. Pour pallier ce problème nous nous proposons d'intégrer un modèle de déplacement en utilisant une estimation par filtre de Kalman. Comme données d'entrée nous utilisons le résultat de l'algorithme d'appariement. Par conséquent à l'instant k par rapport à l'image courante vue par la caméra du véhicule, nous disposons de la pose du véhicule $(x_k^{FabMap}, y_k^{FabMap})$ dans le repère de la trajectoire de référence \mathcal{R}_{Traj_1} correspondant à l'image la plus ressemblante. Prenons le modèle d'évolution suivant :

$$\begin{cases} x_{k+1} = x_k + \Delta t_k \cdot v_{x,k} \\ y_{k+1} = y_k + \Delta t_k \cdot v_{y,k} \\ v_{x,k+1} = v_{x,k} + \epsilon_{v_x} \\ v_{y,k+1} = v_{y,k} + \epsilon_{v_y} \end{cases} \quad (\text{III.60})$$

où ϵ_{Δ} et ϵ_{ω} sont les bruits d'évaluation à l'instant k ajoutés sur les paramètres v_x et v_y afin qu'ils puissent évoluer. Sinon le filtre de Kalman considérera leurs valeurs étant sûres et il ne les remettra pas en cause. Le modèle se réécrit sous la forme d'une fonction d'évolution et une fonction de mesure du type :

$$\vec{X}_k = f(\vec{X}_{k-1}, \vec{\epsilon}_{\vec{X},k}) \quad (\text{III.61})$$

$$\vec{Z}_k = h(\vec{X}_k, \vec{\epsilon}_{\vec{Z},k}) \quad (\text{III.62})$$

avec $\vec{\epsilon}_{\vec{X},k}$ et $\vec{\epsilon}_{\vec{Z},k}$ représentant les bruits de processus respectivement de mesures à l'instant k . Notons \vec{Z}_k le vecteur des mesures bruitées à l'instant k et \vec{X}_k le vecteur d'état composé de la position (x, y) et des composantes du vecteur vitesse (v_x, v_y) suivant l'axe \overrightarrow{OX} et \overrightarrow{OY} . Nous

réalisons l'estimation du vecteur d'état utilisant un algorithme à base de filtre de Kalman étendu.

$$\vec{X}_k = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{bmatrix} \quad \text{et} \quad \vec{Z}_k = \begin{bmatrix} x_k^{FabMap} \\ y_k^{FabMap} \end{bmatrix} \quad (\text{III.63})$$



FIGURE III.41 – A chaque position sur la trajectoire de référence nous assignons une incertitude de positionnement suivant la ressemblance des images dans son voisinage. A titre d'exemple, pour les deux images vertes qui se ressemblent l'ellipse d'incertitude est centrée sur le barycentre et englobe les deux positions

Le terme \mathbf{R}_{o_k} présent dans l'algorithme d'estimation par filtre EKF correspond à la matrice de covariance du bruit de mesure. Recevoir de l'algorithme d'appariement une position qui ressemble aux positions voisines est une information que nous pouvons exploiter et intégrer dans l'algorithme EKF à travers cette matrice. Dans la construction de l'histogramme III.37 nous avons utilisé l'information de mauvais appariement d'une image. Mais cette information se traduit dans une ressemblance de l'image courante avec une image voisine donc dans une ellipse d'incertitude de positionnement, voir figure III.41.

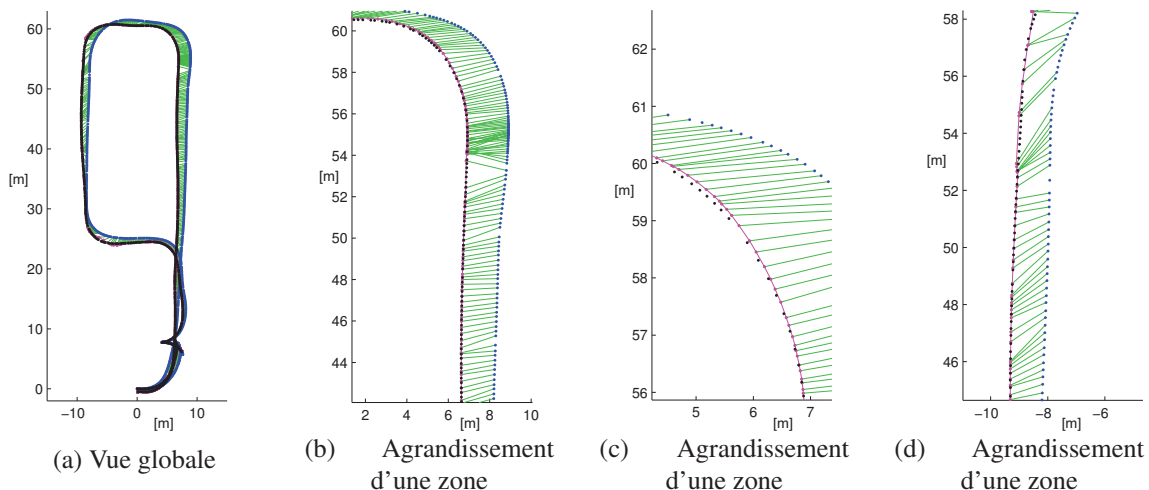


FIGURE III.42 – Résultat de l'algorithme de filtrage EKF utilisant un modèle de déplacement de la position estimée du meilleur appariement; les points noirs correspondent aux positions du véhicule lors de l'enregistrement des images pendant le premier passage et les points bleus respectivement au deuxième enregistrement; les segments indiquent le résultat de correspondance entre la position du véhicule et la position filtrée de l'image de référence (en magenta)

Dans cette partie nous avons montré qu'il est possible de se localiser en utilisant une technique de reconnaissance des images qui se ressemblent. Nous avons proposé également de rendre le résultat moins sensible au bruit en ajoutant un modèle d'évolution et en calculant une position estimée via un filtre de Kalman. Par la suite nous allons expliquer comment cette information de localisation globale peut être utilisée conjointement à une technique de localisation par des données issues des capteurs ultrason pour déterminer la position du véhicule dans toutes les situations.

Nous détaillons par la suite la stratégie d'utilisation de ces techniques de localisation présentées dans ce chapitre pour nous aider à déterminer la position de notre véhicule autonome.

III.4 Stratégie de localisation et bilan

Nous dédions cette dernière section du chapitre à la mise en commun des techniques de localisation choisies dans ce que nous appelons la stratégie de localisation du système de voiturier autonome. Rappelons-nous que l'objectif est de localiser un véhicule autonome dans un couloir de circulation observé dans une étape d'apprentissage à l'issue de laquelle nous disposons de plusieurs éléments comme :

- une trajectoire de référence parcourue par le véhicule durant cette étape et enregistrée utilisant l'approche odométrique [§III.3.1] ;
- des positions GPS autour de la trajectoire de référence ;
- un dictionnaire de mots-visuels et un modèle de détection de la ressemblance des images entraîné sur des données de référence [§III.3.4] ;
- une carte d'obstacle réalisée utilisant des données ultrason [§III.3.3] et complétée par des données visuelles à travers l'approche SfM [§IV.4] ;
- un couloir traversable autour de la trajectoire de référence.

Pour qu'un véhicule roule en automatique, il est impératif que toutes les conditions de sécurité soient réunies. Cela peut se réaliser à travers des validations successives d'un certain nombre d'étapes. Dans ce sens il faudrait distinguer les étapes suivantes :

- le démarrage du système autonome ;
- la demande de démarrage en mode autonome ;
- la validation que toutes les conditions de démarrage en mode autonome soient réunies ;
- le démarrage effectif du véhicule et la réalisation de la manœuvre en mode autonome.

La première étape fait référence au processus d'initialisation et de la préparation du système autonome pour la mise en marche. La deuxième étape correspond à l'action demandée par le conducteur pour que son véhicule débute la phase de manœuvre autonome. La troisième représente une étape de validation et enfin la dernière correspond à la réalisation effective de la manœuvre.

Il est souhaitable que l'étape de démarrage du système autonome se réalise automatiquement avant que le conducteur ne fasse sa demande à travers l'IHM. Dans l'application de voiturier autonome nous utilisons un GPS classique bas coût qui permet une estimation métrique globale de la position [§III.2.1.1]. Si l'environnement de référence est extérieur alors nous disposons des positions GPS autour de l'environnement d'apprentissage, sinon nous disposons des positions GPS indiquant le début de parking souterrain. L'imprécision de localisation de 1 à 30 m qui varie en fonction du nombre de satellites détectés ou l'impossibilité d'utiliser ce système dans des environnements intérieurs fait que nous l'employons uniquement dans un cadre restreint. Le véhicule surveille en continu sa position GPS et détecte automatiquement lorsque le conducteur se dirige vers une zone déjà apprise. Ainsi le système initialise le processus de reconnaissance de l'environnement automatiquement. Par conséquent, le moment où le conducteur est arrêté, le véhicule est censé connaître sa position et pouvoir proposer au conducteur de passer en conduite autonome.

Dès que le système de localisation est démarré, sur des fils d'exécution indépendants les trois techniques de localisation commencent à accumuler des données : la localisation par comparaison d'images, la localisation odométrique et la localisation par appariement des grilles d'occupation. C'est uniquement la première technique qui fournit une information de localisation dès le démarrage. Les deux autres ont besoin d'accumuler un historique. Pour cela, lorsque le conducteur

réclame la mise en mode autonome, le véhicule doit déjà avoir roulé et essayé de se localiser. Sinon le système doit demander explicitement au conducteur de continuer à rouler en mode manuel.

Dans les exemples présentés dans la section §III.3.3 nous avons conclu que le système de localisation par appariement de grilles d'occupation est capable de se localiser de plus en plus précisément avec l'enrichissement de la grille. Nous avons conclu également qu'il est possible d'intégrer dans l'algorithme des a priori de localisation diminuant ainsi l'incertitude et augmentant la vitesse de convergence. Ces a priori de localisation par l'appariement des grilles d'occupation seront donnés par le résultat de la méthode de localisation globale par ressemblance d'images.

Dans la section §III.3.4 nous avons montré des résultats de la localisation des images acquises lors d'un deuxième enregistrement dans le repère de la trajectoire préalablement apprise. Nous avons montré qu'il existe des situations où la forte ressemblance de l'environnement perturbe la méthode de localisation et que le résultat peut présenter des *sauts*. Ces inconvénients ont été réglés par l'ajout d'un modèle de déplacement qui a un rôle à la fois de filtrer les données aberrantes mais également d'estimer la nouvelle position suivant l'historique du déplacement.

En conclusion de ce chapitre, nous disposons d'une architecture d'un système de localisation permettant l'utilisation de l'information issue des capteurs GPS, odométriques, visuels et ultrason pour estimer la position du véhicule lors de l'évolution en mode autonome.

Le chapitre suivant est dédié à l'analyse de la zone traversable. Nous présenterons les approches développées de détection d'obstacles par capteurs ultrason et également par approche *Structure from Motion* utilisant l'information visuelle.

CHAPITRE IV

Détection d'obstacles et analyse de la zone traversable

Sommaire

III.1	Problématique	74
III.2	Localisation : État de l'art	74
III.2.1	Techniques de localisation absolue	75
III.2.2	Techniques de localisation relative	78
III.3	Évaluation des techniques de localisation proposées	86
III.3.1	Localisation par odométrie	86
III.3.2	Localisation par techniques de SLAM	95
III.3.3	Localisation par appariement des grilles d'occupation	103
III.3.4	Estimation des a priori de localisation et d'orientation par techniques de comparaison d'images	114
III.4	Stratégie de localisation et bilan	123

IV.1 Problématique

Comme nous l'avons déjà présenté, lors des phases d'apprentissage, le conducteur réalise manuellement les manœuvres de conduite pour garer le véhicule [§I.6]. Pendant cette étape, le système de voiturier autonome se charge d'enregistrer la trajectoire et tous les obstacles qu'il perçoit durant le déplacement. La problématique que nous allons aborder est la manière dont l'information liée aux obstacles et implicitement à la traversabilité de la zone est acquise, utilisée et stockée. Ceci nous amène à employer une structure de données de type de grille d'occupation pour le type d'information que nous souhaitons utiliser. Plus précisément, nous nous sommes intéressés à savoir uniquement si une zone observée est occupée ou non par un obstacle. Enfin nous souhaitons modéliser et accéder facilement et en temps réel à ces informations. Nous allons initialement remplir la carte d'occupation par l'information issue des capteurs ultrason, nous allons analyser les résultats et tirer les conclusions. Dans un second temps, nous allons pallier le problème du trop court champ de perception du capteur ultrason, en exploitant l'information vidéo.

IV.2 Gestion de la zone traversable : État de l'art

Nous nous intéressons d'abord à l'état de l'art lié à la problématique de carte d'occupation. Initialement la carte sera alimentée par les informations issues des capteurs ultrason. En complément du système de détection d'obstacle à base de capteurs ultrason, une technique de *Structure from Motion* est mise en place pour l'extraction des obstacles depuis une séquence d'images.

IV.2.1 Grille d'occupation

Dans cette section, nous détaillons la notion de grille d'occupation. Cela est une forme de représentation discrète de l'environnement sous forme de cellule d'une certaine taille où chaque cellule contient des informations relatives à la présence ou à l'absence des obstacles.

IV.2.1.1 Grille standard

Un critère de différenciation des grilles est la manière de stockage de la grille d'occupation. Par exemple on peut sauvegarder simplement une matrice de la taille de la grille (figure IV.1). Ce type de carte a l'avantage de permettre l'accès rapide à une donnée stockée à l'emplacement (i, j) . L'inconvénient est que l'espace mémoire utilisé évolue avec une loi $O(n^2)$, où n est une dimension de la grille pour une résolution choisie.

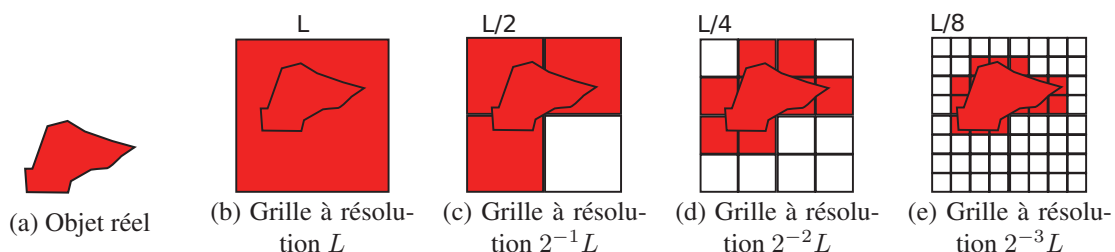


FIGURE IV.1 – Grille d'occupation à différents niveaux de résolutions : $L, 2^{-1}L, 2^{-2}L, 2^{-3}L$

Toujours dans la figure IV.1, nous présentons l'évolution de l'espace occupé par un objet dans un carte 2D suivant le niveau de résolution choisi.



FIGURE IV.2 – Méthodes de remplissage de la grille d'occupation

Nous reprenons le même problème de stockage de l'information de cellule occupée dans la grille, mais cette fois-ci pour un simple segment. Dans les figures IV.2(a) et (b) nous faisons ressortir une différence sur le remplissage de la grille en fonction du critère choisi dans la décision si une cellule est occupée ou non. Les deux méthodes les plus employées dans la littérature sont la méthode de tracé de segment de Borenstein [Bresenham, 1965; Raschke and Borenstein, 1990; Borenstein and Koren, 1991b,a; Ye and Borenstein, 2004] et la méthode de *raycast* [Roth, 1982; Ertl, 1996; Woop et al., 2005]. Suite à la comparaison des deux approches que nous avons réalisée et que nous présentons dans l'annexe A.3, la décision a été de poursuivre le développement en utilisant l'approche raycast car elle modélise plus fidèlement la notion d'occupation.

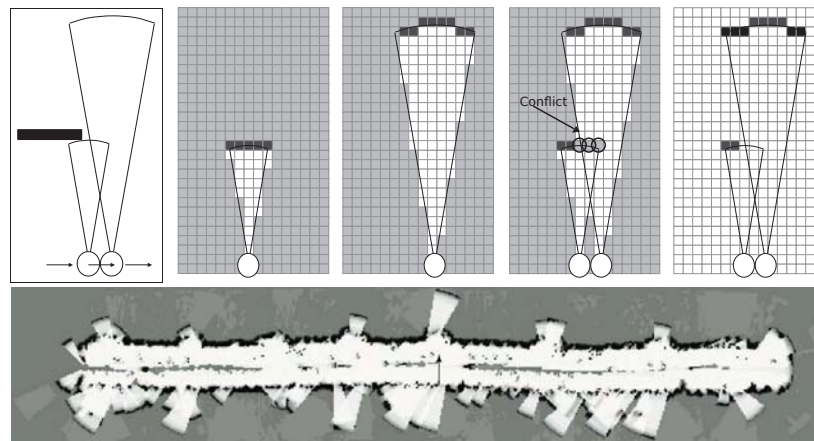


FIGURE IV.3 – Illustration d'un conflit de détection d'un obstacle par un capteur ultrason en mouvement et exemple de grille d'occupation obtenue par l'utilisation de la méthode probabiliste proposée par [Thrun, 2003]

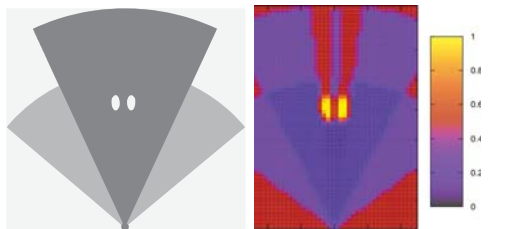


FIGURE IV.4 – Estimation de la position de deux objets utilisant des capteurs de différentes précisions angulaires [Coué, 2003]

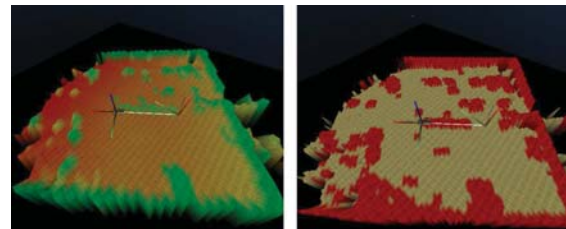


FIGURE IV.5 – Représentation de l'environnement par Modèle Numérique de Terrain Robosoft [Quemener, 2010]

La figure IV.3 présente les résultats des travaux de S. Thrun sur une technique de stockage de l'information issue d'un capteur de distance générique (capteur ultrason, stéréo-caméra, LiDAR, etc.) dans une grille 2D et la manière dont les conflits de détection sont traités [Thrun, 2003]. Nous présentons dans la figure IV.4 les résultats des travaux de C. Coué qui portent sur la définition d'un

modèle de représentation d'environnements dynamiques et encombrés, appelé *filtre d'occupation bayésien* [Coué, 2003]. Dans ces travaux, la représentation de l'environnement est découpé en cellules et, à partir de données capteur, pour chaque cellule une probabilité d'être occupée est calculée. La figure IV.5 illustre les résultats obtenus en simulation ou en rejeu par l'algorithme de reconstruction de MNT¹ proposé par *Robosoft* dans le cadre du projet 2RT3D. Si la variance de l'altitude dépasse un seuil, le nœud est classé comme non traversable [Quemener, 2010].

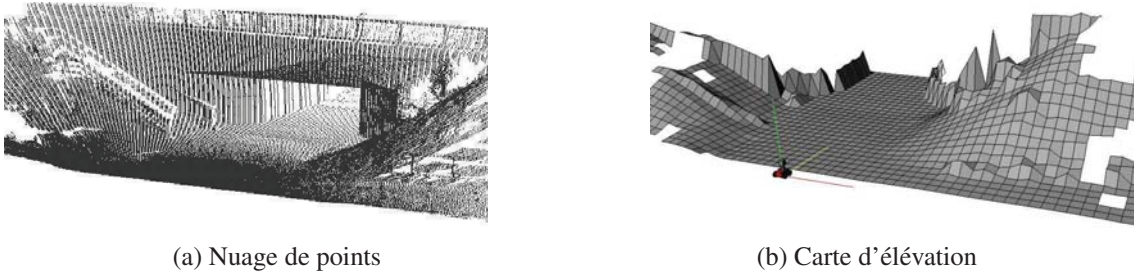


FIGURE IV.6 – Transformation d'un nuage de points enregistré avec un Lidar SICK LMS installé sur un système de balayage angulaire dans une carte d'élévation [Pfaff et al., 2007]

P. Pfaff et R. Kümmerle se préoccupent également à la problématique de stockage de l'information des obstacles [Pfaff et al., 2007; Kümmerle et al., 2008]. Un nuage de points 3D (figure IV.6(a)) est transformé dans une carte d'élévation (figure IV.6(b)) sur la base sur une grille discrète qui dans chaque cellule contient la valeur de l'élévation du sol ou des obstacles.

IV.2.1.2 Grille de d'occupation utilisant les quadtree

Une autre méthode plus efficace, a été proposée par [Finkel and Bentley, 1974] et démocratisée par [Samet, 1984] et elle concerne la stockage des données dans un arbre, donc de manière hiérarchique. Il propose une structure appelée *quadtree* (ou arbre quaternaire ou arbre Q) qui suppose de subdiviser successivement l'espace en quatre parties égales et de stocker l'information dans une structure de données de type arbre dans laquelle chaque nœud a quatre fils. La manière de découpage de l'espace peut être rectangulaire ou triangulaire comme dans les figures IV.7(a) et (b). Pour la suite de notre projet, le choix a été d'utiliser un découpage en zones rectangulaires pour des raisons de simplicité algorithmique et pour la rapidité calculatoire.

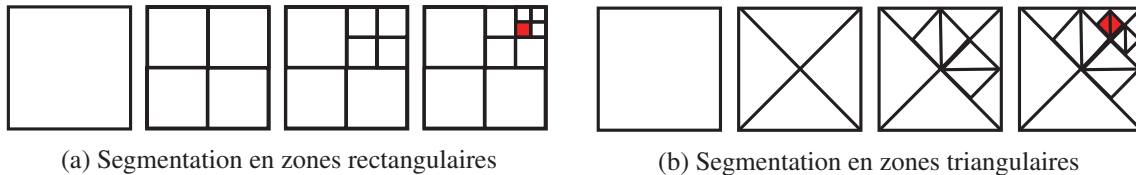


FIGURE IV.7 – Quadtree - méthode de segmentation de l'espace

Sur le même principe de découpage de l'espace planaire, la technique peut s'appliquer sur des volumes. Ainsi l'octree tel que présenté par [Szeliski, 1993] ou plus récemment par [Wurm et al., 2010] dans le système *Octomap* permet de découper successivement un cube en 8 cubes de volumes égaux (voir figures IV.8(a) et (b)).

1. Modèle Numérique de Terrain;

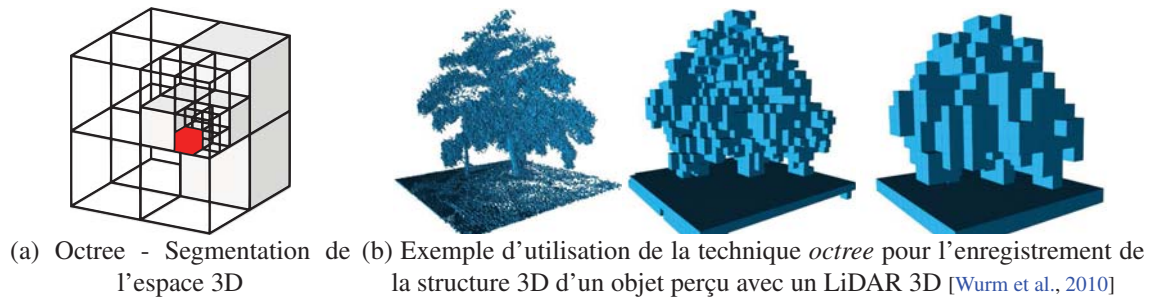


FIGURE IV.8 – Octree - Carte 3D multi-résolution

IV.2.1.3 Variations de grilles d'occupation

C. Witzgall propose l'utilisation d'une grille d'occupation radiale à résolution variable (figure IV.9(a)). Plus l'obstacle est loin du capteur, moins les détails seront sauvegardés car la résolution de la grille baisse avec l'augmentation de la distance [Witzgall et al., 2003].

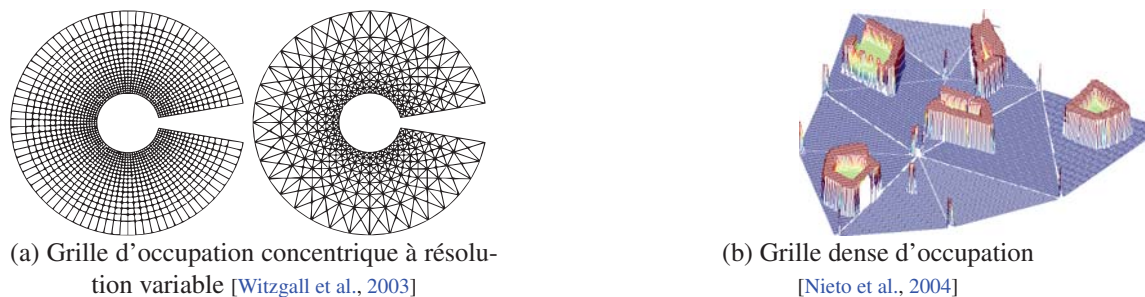


FIGURE IV.9

J.I. Nieto présente un modèle dense d'environnement (figure IV.9(b)) [Nieto et al., 2004]. Une carte d'amers et d'obstacles est décomposée en cartes triangulaires à l'aide des LTR¹. Une fois la décomposition réalisée, chaque carte dispose de son propre système de coordonnées donné par le barycentre. Ainsi, une carte plus grande est décomposée dans des sous-cartes pour la simplification du problème. Quelle que soit la méthode de stockage ou le principe de découpage, la grille d'occupation est une méthode simple de gestion de l'information liée à la traversabilité d'une zone. Nous utilisons également cet outil et nous revenons sur ce sujet dans la section §IV.3.1.

Par la suite nous nous intéressons aux algorithmes de type *Structure from Motion* pour étendre l'horizon de perception au-delà de la limite de 4,095 m des capteurs ultrason. En plus, l'utilisation d'une deuxième approche nous permettra de pallier les problèmes de perception des obstacles très angulés qui dévient les échos ou le problème de perception du sol comme étant un obstacle ou bien le problème de détection de l'obstacle dominant en termes de puissance de l'écho.

IV.2.2 Algorithmes du type *Structure from Motion*

L'acronyme SfM désigne ce que la communauté scientifique de traitement d'images appelle *Structure from Motion*. Cela fait référence à une gamme de techniques qui ont pour but d'estimer la structure 3D de l'environnement utilisant des séquences d'images qui éventuellement peuvent

1. Local Triangular Region;

être couplées à des informations de déplacement (odométrie, IMU, etc.). Nous introduisons ici ce concept car des techniques de SfM sont très liées aux techniques de SLAM.

Pour le SLAM, la contrainte de temps réel est une contrainte forte à respecter. Quant au SfM, les algorithmes passent par des étapes d'optimisation globale en post-traitement du type ajustement de faisceaux. Une deuxième différence est donnée directement par les noms des deux techniques. SfM a comme but de réaliser de la cartographie 3D, par conséquent la richesse de la carte et la fidélité de représentation sont importantes. Le SLAM réalise la cartographie mais avec un but différent : s'y localiser. Cela implique que la carte est d'une densité suffisante de points/amers pour permettre le suivi de position, la localisation globale et une relocalisation ultérieure.



FIGURE IV.10 – Création de cartes 3D de l'environnement utilisant des images issues de l'internet [Snavely et al., 2006]

Il semble important d'évoquer les travaux de [Snavely et al., 2006, 2008] car cela s'intègre dans le courant de développement de cartes de grandes dimensions utilisant de l'information en libre accès sur internet. La figure IV.10 montre le nuage de points 3D de la cathédrale Notre-Dame de Paris réalisé à travers des techniques SfM utilisant des photos prises par les touristes et postées sur l'internet. Sur le même principe [Agarwal et al., 2009] annoncent la création de la carte d'une partie de la ville de Rome dans une journée. Ce courant rejoint l'intention exprimée par Google de créer le nuage des points 3D issus de la totalité des images existantes dans leur *Google Street View*.

Sans avoir vocation à détailler ce sujet, nous présentons uniquement les bibliothèques et les applications suivantes comme l'état de l'art dans le domaine de reconstruction des nuages de points 3D : OpenMVG¹ [Moulon et al., 2012], *Theia* [Sweeney et al., 2015], COLMAP [Schönberger and Frahm, 2016], MicMac [Pierrot-Deseilligny and Paparoditis, 2006], MVE² [Fuhrmann et al., 2014], Ceres [Agarwal et al.] ou Libmv³ [Mierle and Maclean, 2008; Mierle, 2008].

Comme nous l'avion déjà présenté, certains travaux de recherche ont leurs racines dans les techniques SfM mais les applications développées sont de type SLAM. Dans ce sens on cite le PTAM⁴ [Klein and Murray, 2007, 2009], le SBA⁵ [Lourakis and Argyros, 2009], le DTAM⁶ [Newcombe et al., 2011] ou bien le LSD-SLAM⁷ [Engel et al., 2014].

Dans les sections suivantes, nous détaillons les deux approches développées. Tout d'abord nous détaillerons la création de la grille d'occupation et son alimentation en information issue des capteurs ultrason.

1. **Open** Multiple View Geometry; 2. **Multi-View** Reconstruction Environment; 3. ou **LMV** i.e. *Library for MultiView* Reconstruction; 4. **Parallel Tracking And Mapping**; 5. **Sparse Bundle Adjustment**; 6. **Dense Tracking and Mapping**; 7. **Large Scale Dimension-SLAM**;

IV.3 Localisation des obstacles détectés par les capteurs ultrason

Une des propriétés des capteurs ultrason consiste en une meilleure qualité et précision de mesure dans la partie centrale du cône de visibilité et une détection moins précise sur les parties latérales. La largeur du cône constitue à la fois un avantage et un inconvénient. Les figures II.30(a) à (c) et II.31(a) à (c) illustrent un cône de visibilité dans lequel les obstacles seront observés.

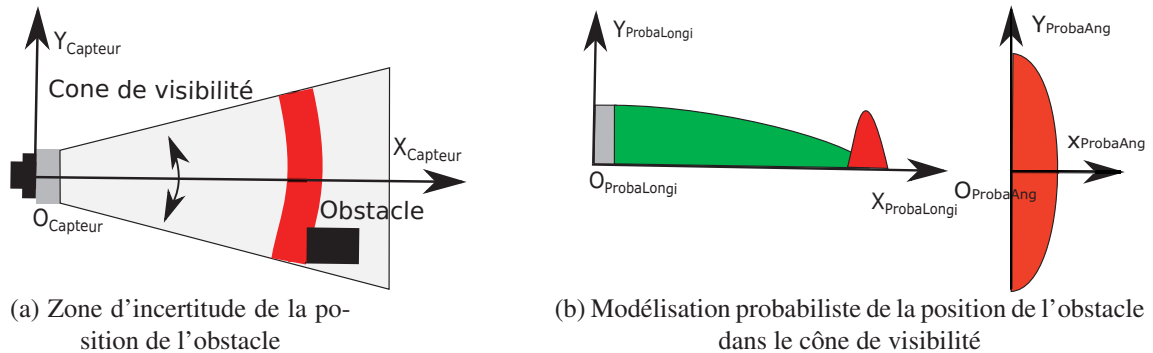


FIGURE IV.11 – La figure (a) présente la projection bidimensionnelle du cône de détection d'un capteur ultrason, d'un obstacle ainsi que de la zone d'incertitude de la position de l'obstacle. La figure (b) présente la modélisation probabiliste de calcul de la distance à l'obstacle

La figure IV.11(a) illustre en noir un obstacle qui se trouve dans le cône de visibilité du capteur. La couleur rouge définit la zone dans laquelle l'obstacle est potentiellement situé. La figure IV.11(b) illustre une approche probabiliste de la position de l'obstacle. L'inconvénient majeur de ce type de capteur est l'incertitude de positionnement de l'obstacle dans le cône d'observation. Par conséquent ce type de capteur aura une bonne précision longitudinale et une mauvaise précision angulaire. Cela peut aussi être considéré comme un avantage car cela permet de mieux balayer l'environnement avec un champ de vue large.

Dans la section §II.3.1.4, dédiée aux caractéristiques du SCU, nous avons détaillé le fait que le SCU renvoie 8 paires de points, $(ObjX_A, ObjX_B)$ avec X de 1 à 8, placées dans le repère véhicule et qui forment des segments de longueurs égales aux largeurs apparentes des obstacles observés. L'inconvénient de ce système vient du fait qu'il fournit l'accès uniquement aux 8 plus proches objets et que très souvent la largeur de ces obstacles n'est pas estimée. Pour ces raisons, nous n'avons pas pu réaliser une cartographie complète de l'environnement utilisant ces données post-traitées du SCU. En revanche, nous avons développé une approche géométrique. L'utilisation de plusieurs capteurs en émetteurs et récepteurs à la fois permet de localiser les obstacles par une technique de triangulation géométrique.

IV.3.1 Calcul de la position des obstacles par une approche géométrique

Nous rappelons que les mesures des capteurs sont nommées distance calculées à partir des *échos directs*, d_{EchosD/C_i} (en vert dans la figure IV.12(a)) et distances calculées utilisant les *échos croisés*, $d_{\text{EchosC}/(C_i/C_{i-1})}$ et $d_{\text{EchosC}/(C_i/C_{i+1})}$ (en bleu dans la figure IV.12(a)). Pour un capteur C_i les échos sont reçus par le récepteur du même capteur C_i et par les récepteurs des capteurs voisins C_{i-1} ou C_{i+1} (figures IV.12(a) et (b)). Dans ce cas on parle d'un triplet (récepteur, émetteur-récepteur, récepteur). Une situation à part se présente pour les capteurs latéraux (C_1, C_6, C_7, C_{12}) qui n'ont qu'un seul capteur voisin.

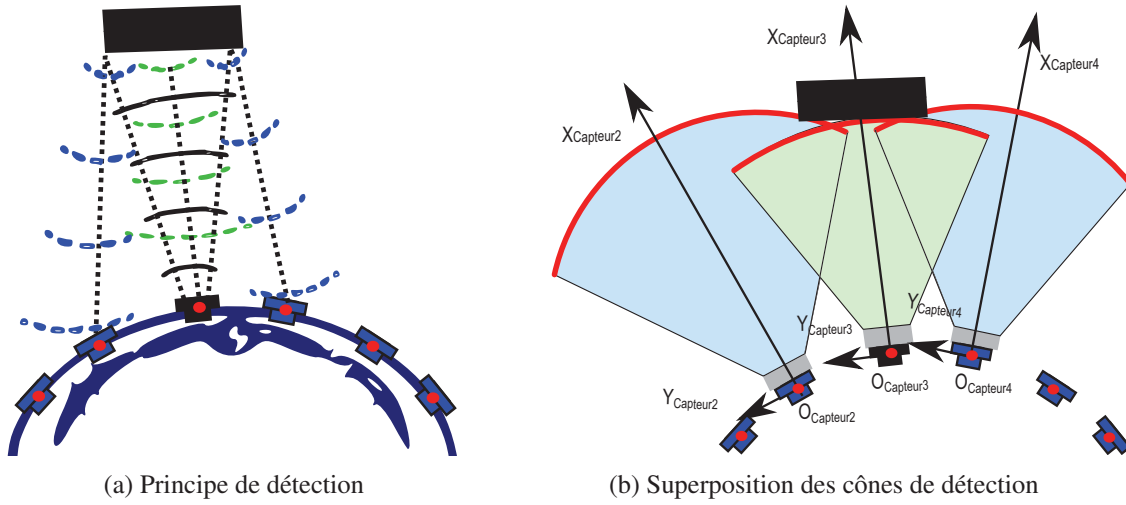


FIGURE IV.12 – Pour un capteur C_i les échos sont reçus par le récepteur du même capteur (C_i) et par les récepteur des capteurs voisins (C_{i-1} ou C_{i+1})

Les figures IV.13(a) et (b) présentent un cas de localisation utilisant uniquement d_{EchosD/C_i} et $d_{EchosC/(C_i/C_{i-1})}$. Considérant d_{EchosD/C_i} connu, alors le point $A(x_A, y_A)$ décrit un cercle de centre O_i et rayon R_i (figure IV.13(a)).

$$R_i = d_{EchosD/C_i} = d(O_i, A) \quad (IV.1)$$

$$R_i = \sqrt{(x_A - x_{O_i})^2 + (y_A - y_{O_i})^2}$$

L'ellipse peut être définie comme le lieu des points dont la somme des distances à deux points fixes, appelés foyers de l'ellipse, est constante et égale à une valeur fixée. Considérant $d_{EchosC/(C_i/C_{i-1})}$ connu, alors le lieux des points dont la somme des distances à deux points fixes est constante et égale à une valeur fixée, décrit une ellipse. Ainsi, le point $A(x_A, y_A)$ décrit une ellipse de foyers O_i et O_{i-1} (figure IV.13(b)) respectant la propriété suivante :

$$d(O_i, A) + d(A, O_{i-1}) = d_{EchosC/(C_i/C_{i-1})} = cst. \quad (IV.2)$$

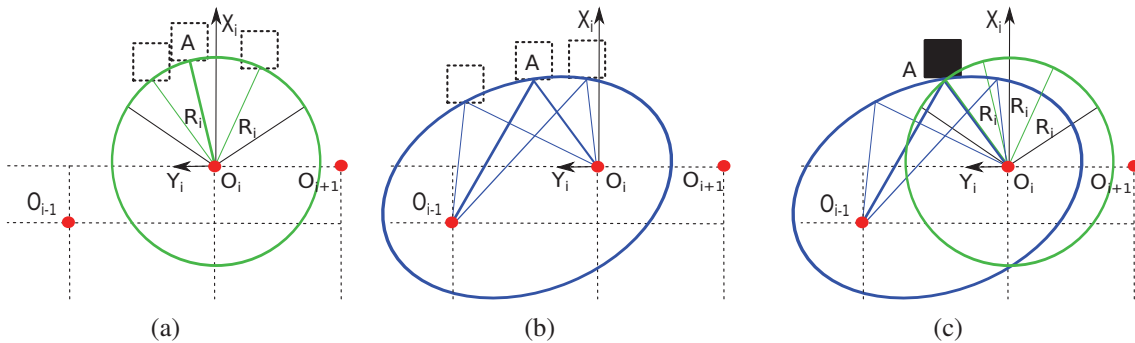


FIGURE IV.13 – Calcul de la position d'un obstacle par méthode géométrique. La contrainte $d_{EchosD/C_i} = cst$ impose au point $A(x_A, y_A)$ de décrire un cercle (figure (a)). La contrainte $d_{EchosC/(C_i/C_{i-1})} = cst$ impose au point $A(x_A, y_A)$ de décrire une ellipse de foyers O_i et O_{i-1} (figure (b)). La figure (c) illustre la position de l'obstacle déterminée suite à l'intersection d'un cercle et d'une ellipse

Par conséquent, la position de l'obstacle $A(x_A, y_A)$ représente le lieu d'intersection d'un

cercle avec une ellipse (figure IV.13(c)), la solution au système de deux équations :

$$\begin{cases} d_{EchosD/C_i} &= d(O_i, A) \\ d_{EchosC/(C_i/C_{i+1})} &= d(O_i, A) + d(A, O_{i+1}) \end{cases} \quad (IV.3)$$

$$\begin{cases} d_{EchosD/C_i} &= \sqrt{(x_A - x_{O_i})^2 + (y_A - y_{O_i})^2} \\ d_{EchosC/(C_i/C_{i+1})} &= d_{EchosD/C_i} + \sqrt{(x_A - x_{O_{i+1}})^2 + (y_A - y_{O_{i+1}})^2} \end{cases} \quad (IV.4)$$

Le système d'équation (IV.3) a deux solutions mais, nous n'allons garder que le point qui se situe dans les cônes de visibilité des deux capteurs.

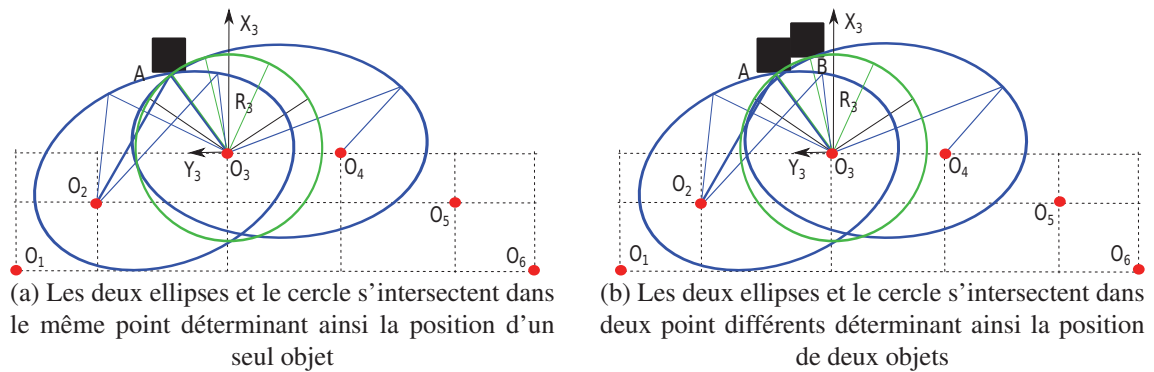


FIGURE IV.14 – Détection d'une ou de deux positions par un triplet (récepteur, émetteur-récepteur, récepteur)

La génération de SCU qui équipe le véhicule prototype utilise ce qu'on appelle des échos du *premier ordre*. C'est à dire que pour une onde ultrason, la distance à l'obstacle est calculée en se basant uniquement sur l'écho dont la puissance est maximale parmi les échos reçus. Ceci signifie également que parmi les obstacles, généralement ce ne sont que les obstacles les plus proches qui sont détectés. La disposition des capteurs ultrason sur différents angles favorise la perception des obstacles depuis plusieurs points et angles de vue. De plus, le fait d'observer les obstacles (dans le cas où leur disposition le permet) à la fois par les capteurs avant et arrière, aide à la construction d'une carte fidèle à l'environnement réel.

Un autre facteur à prendre en compte est le temps écoulé entre l'observation de l'obstacle et la réception des trames CAN. Pour pallier le problème de faible fréquence de mise à jour de données (10 Hz), avant la mise à jour de la carte, les positions des obstacles sont replacées dans le temps et dans l'espace. Pour réaliser cela, nous utilisons un *modèle bicyclette* pour créer en continu la trajectoire de déplacement du véhicule par mesures odométriques. Les détails de la création de la trajectoire seront présentés dans le chapitre V. Par la suite nous présentons des cartes résultantes dans le processus d'acquisition et cartographie mis en place.

IV.3.2 Résultats de la cartographie par le système ultrasons

En appliquant le principe précédemment énoncé pour tous les données des capteurs ultrason nous sommes capables d'avoir une localisation de tous les obstacles proches du véhicule. Comme nous ne disposons pas d'un LiDAR pour facilement comparer la qualité de la carte créée, nous avons mis en place des scénarii de test pour mesurer l'écart entre les distances perçues et celles mesurées.

IV.3.2.1 Analyse des résultats

Dans la figure IV.15(a) nous présentons une situation de cartographie d'un environnement contenant des obstacles de forme et dimension connue. L'image IV.15(b) permet de visualiser la scène : une route étroite avec une seule voie de circulation bornée par des buissons du côté gauche de la route (en vert) et une rangée des places de parking du côté droit (voitures en rouge), des poteaux (en magenta) et des arbres (en jaune). Utilisant cette scène nous pouvons retirer plusieurs caractéristiques qualitatives. Les buissons de manière générale absorbent les ultrasons (en orange nous signalons une zone de non détection) ou créent des multi-échos. Pour ces raisons-là nous retrouvons une variation de ± 0.2 m par rapport à la distance réelle.

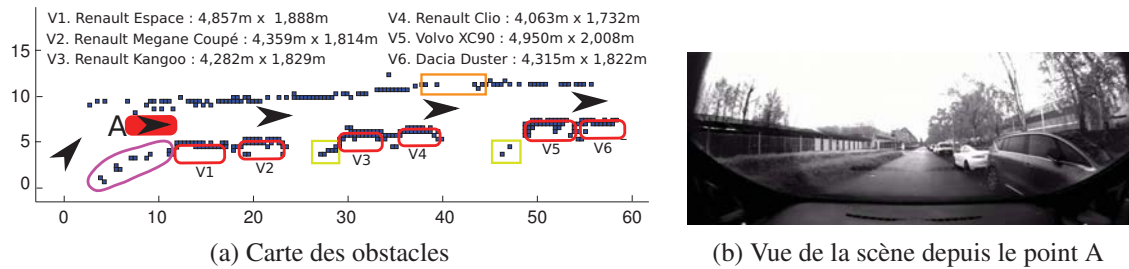


FIGURE IV.15 – Situation de scan d'une rangée de voiture pour mesurer les performances de la cartographie par SCU; en rouge des véhicules, en magenta des poteaux, en jaune des arbres, en orange un zone des buissons non détectés

Les objets de taille moyenne, comme des poteaux de 35 cm en diamètre (en magenta) sont bien détectés à condition qu'ils ne se trouvent pas dans la proximité d'un grand objet qui renvoie des échos plus puissants. Nous rappelons que le SCU utilise uniquement l'écho le plus puissant parmi tous les échos renvoyés. Une autre remarque est qu'il est difficile de détecter l'espace vide entre deux objets proches, comme pour les paires de véhicules (V_1, V_2), (V_3, V_4) ou (V_5, V_6). Concernant les distances observées jusqu'aux véhicules, nous retrouvons des écarts de moins de 5 cm. Ces résultats sont retrouvés pour toute détection d'objet qui a une surface de réflexion verticale et lisse.

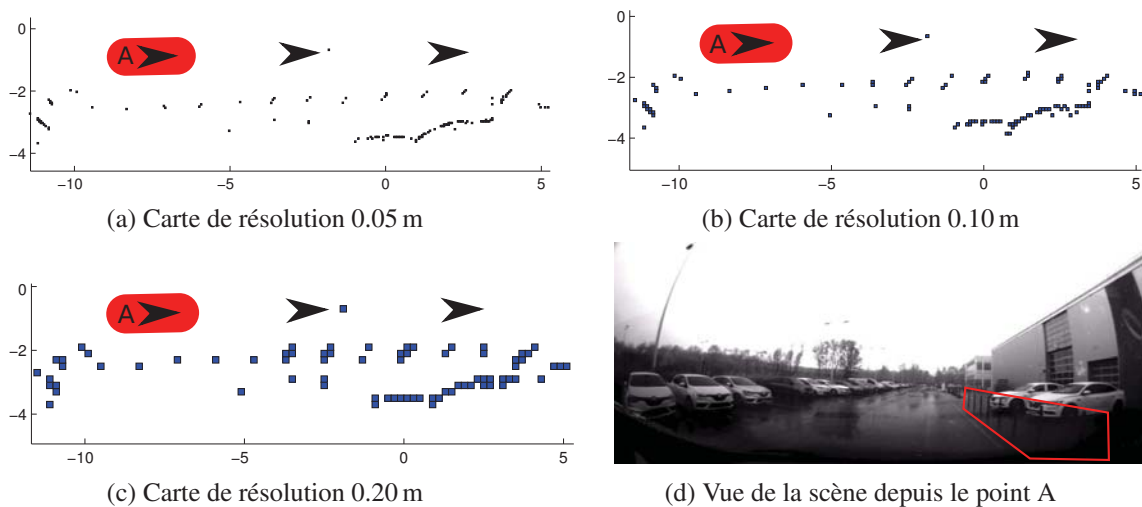


FIGURE IV.16 – Les figures de (a) à (c) illustrent une carte des objets de petite dimensions : une rangée des poteaux de 8 cm de diamètre écartés à 1,2 m. La figure (d) donne une vue de la scène pour le véhicule (en rouge) situé au niveau du point «A»

Les figures IV.16(a) à (d) présentent la carte d'une rangée des poteaux métalliques de 0,10 m en diamètre et écartés à 1,20 m. Leur détection se réalise pour une vitesse de déplacement de 0,80 m/s et une distance de détection d'environ 1 m pour les capteurs les plus proches de la rangée. La vitesse et la distance de détection faible permettent la bonne détection. En revanche une augmentation de la vitesse fait que les obstacles ne sont plus perçus.

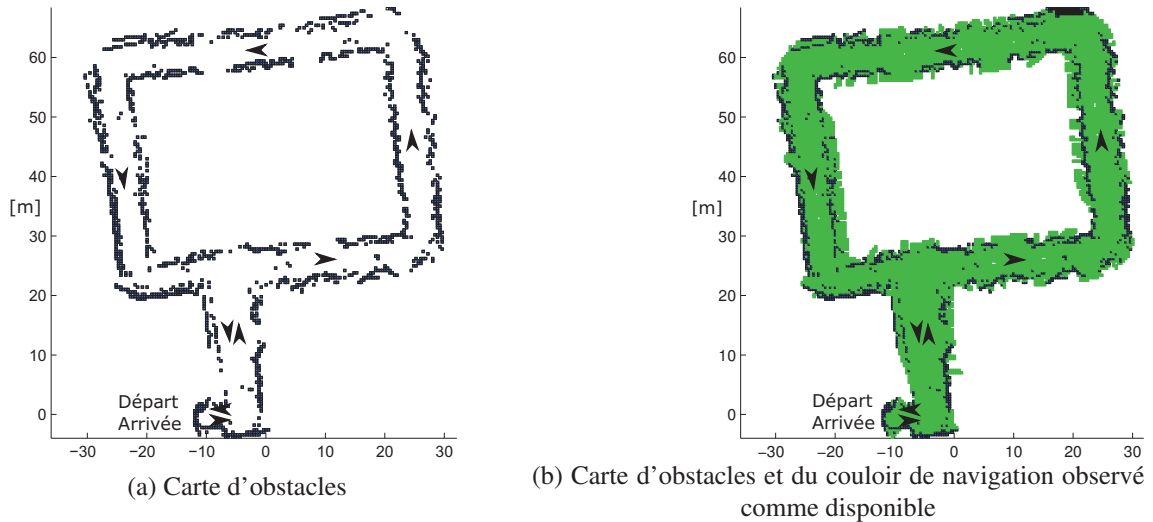


FIGURE IV.17 – Exemple de carte d'occupation et de couloir observé réalisée par fusion de données issues des capteurs ultrason et des capteurs odométriques

Les figures IV.17(a) et (b) présentent le résultat de la cartographie d'un environnement riche en obstacles comme, par exemple, un parking souterrain. La carte illustrée dans la figure IV.17(a) indique en bleu foncé l'espace occupé. Une représentation de ce type est inexacte car on ne fait pas la distinction entre l'espace vide et l'espace inexploré.

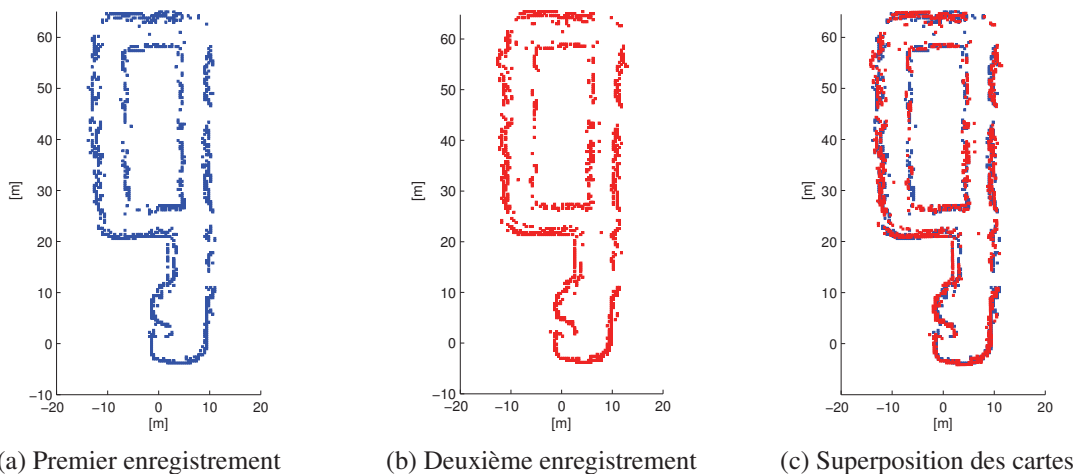


FIGURE IV.18 – Comparaison des deux cartes réalisées utilisant les capteurs ultrason lors des deux essais différents dans le même environnement

La figure IV.17(b) réalise cette distinction en affichant en vert l'espace vide connu, en bleu l'espace occupé et en blanc l'espace inexploré. Pour la réalisation de cette carte, uniquement une localisation relative par odométrie a été utilisée. Même si la trajectoire est longue d'environ 200 m, le véhicule revient dans le point de départ avec une erreur réelle de position de 0,07 m

et une erreur angulaire d'environ 2° que nous avons mesuré à travers le décalage dans l'image d'un point d'intérêt pris comme référence. Le sujet de construction de trajectoire par odométrie est détaillé dans la section §III.3.1.

Nous avons réalisé des enregistrements de la même zone de parking souterrain lors des deux essais indépendants (figures IV.18(a) et (b)). Les deux cartes qui en résultent sont manuellement superposées et le résultat est illustré dans la figure IV.18(c). Pour une résolution de 0,05 m la première carte est composée de 1936 cellules et la deuxième de 1888 cellules. Par superposition, nous constatons un taux de superposition des cellules de la deuxième carte sur celle de la première en valeur de 56%. En choisissant une résolution de 0,2 m la première carte sera composée de 1460 cellules et la deuxième de 1458. Cette fois-ci le taux de superposition s'élèvera à 76%. Nous reviendrons sur la problématique d'appariement des grilles d'occupation dans la section §III.3.3 avec une approche d'appariement de cartes par corrélation.

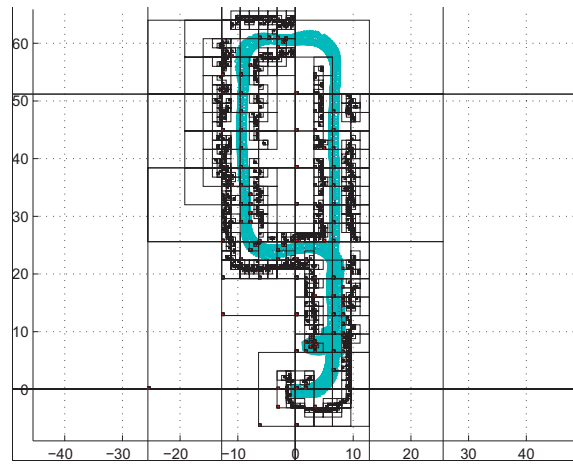


FIGURE IV.19 – Carte d'obstacle multi-résolution (résolution de $2^k \cdot 0,05$ m, avec k un nombre entier positif ou égal à zéro) et gabarit occupé par le véhicule durant le déplacement

L'utilisation d'un découpage successif en sous-sections (figure IV.19) et le stockage de l'information dans un arbre du type quadtree à la place d'une grille standard permettent une gestion dynamique de la résolution de la carte. Dans les figures IV.20(a) à (e), nous mettons en parallèle une même carte avec plusieurs niveaux de résolution de 3,2 m à 0,05 m.

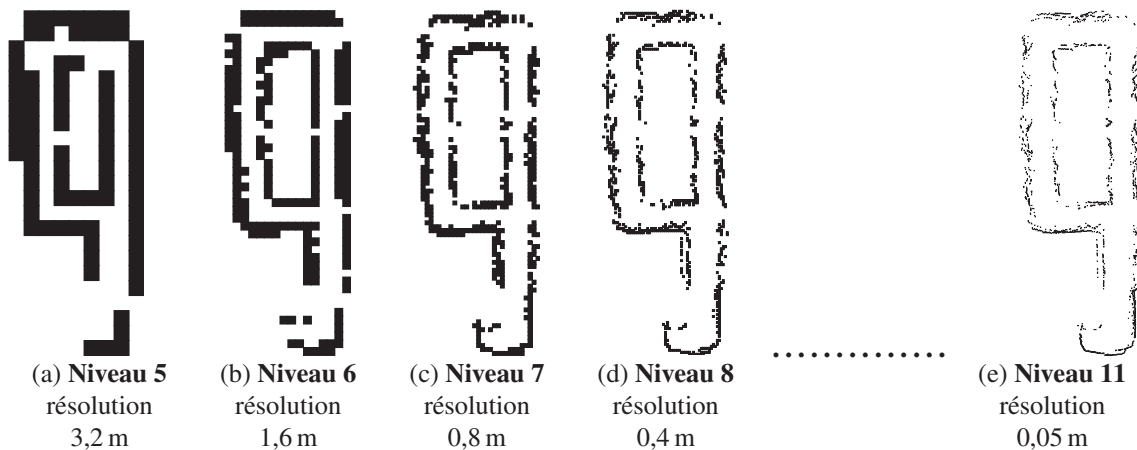


FIGURE IV.20 – Carte d'obstacle à plusieurs niveaux de résolutions stockée dans un quadtree

IV.3.2.2 Filtrage de la grille

Une étape importante consiste dans le filtrage de la carte d'obstacles au long de la trajectoire réalisée par le véhicule. Le cas le plus fréquent de fausse détection est rencontré en cas de freinage, quand le tangage du véhicule entraîne la détection du sol comme obstacle (figures IV.22(a) et (b)).

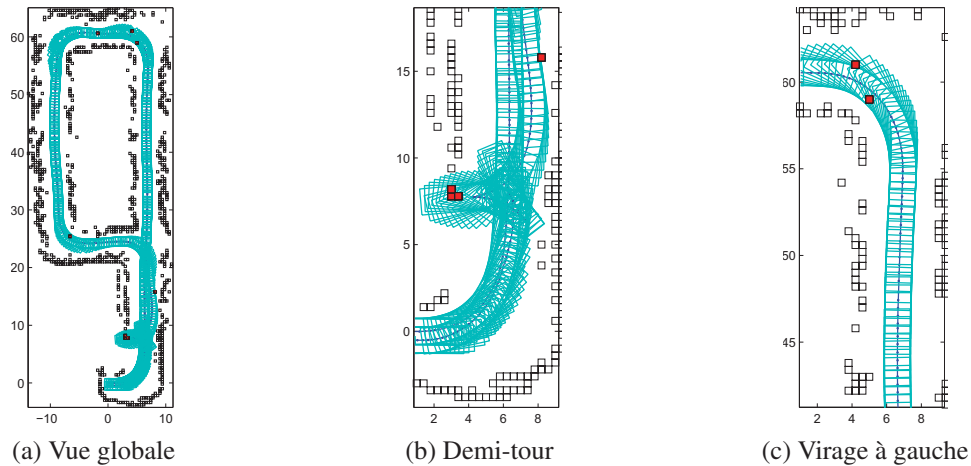


FIGURE IV.21 – Nettoyage de la carte originale par enlèvement des faux obstacles observés sur la trajectoire du véhicule. Ce phénomène est dû principalement au phénomène de *salut*, c'est à dire d'inclinaison des capteurs vers le plan de la route lors des phases de forts freinages durant les manœuvres à basse vitesse (figure (b)) ou avant un virage (figure (c))

Les figures IV.21(a) à (c), illustrent des rectangles colorés en rouge qui sont situés dans l'espace occupé par le véhicule lors des manœuvres. Ces obstacles seront effacés de la carte car leur présence n'est pas continue, ou il s'agit d'une fausse détection.

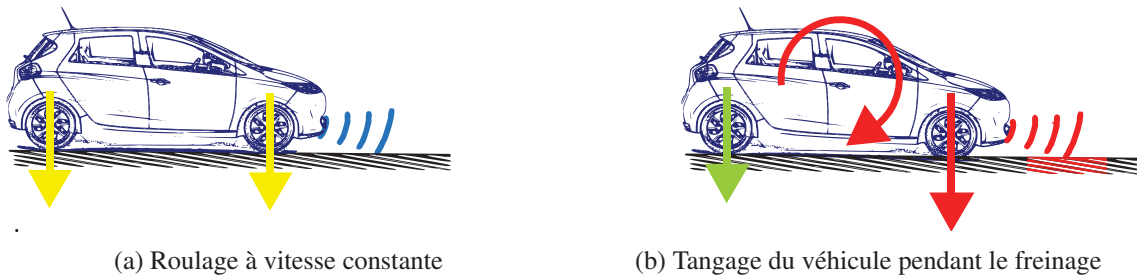


FIGURE IV.22 – Fausse détection du sol pendant une phase de freinage

IV.3.2.3 Analyse de l'espace de stockage

Nous nous intéressons à présent à la complexité de l'utilisation du quadtree, à la problématique de l'espace occupé par l'information utile, à l'ajout ou à la suppression d'un élément dans l'arbre (donc dans la carte) et à la gestion de la résolution utilisant cette structure. Pour cela nous rappelons qu'un arbre est composé d'une *racine*¹, de nœuds internes², de feuilles (ou nœuds externes)³, de branches et d'étiquettes⁴. Les éléments qui caractérisent un arbre sont : la profondeur

1. Nœud ne possédant pas de parent; 2. Élément possédant des fils (sous-branches); 3. Élément ne possédant pas de fils dans l'arbre; 4. Information associée au nœud;

d'un nœud¹, la hauteur de l'arbre², la taille de l'arbre³, la longueur de cheminement⁴.

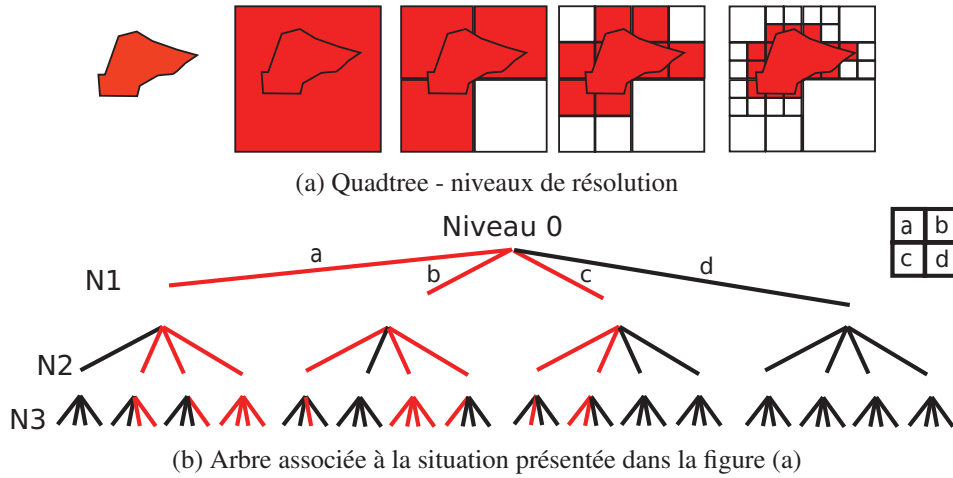


FIGURE IV.23 – Quadtree - Exemple d'arbre d'occupation. En rouge on retrouve les cellules occupées et les branches associées à ces cellules

Considérons r la profondeur maximale de l'arbre. Dans ce cas le quadtree complet a 4^r feuilles. Prenons l'exemple illustré dans la figure IV.23(b) : un quadtree complet à 3 niveaux de profondeur qui a $4^3 = 64$ feuilles, donc une capacité de sauvegarder 64 cellules. Pour sauvegarder ces 4^r cellules, l'espace mémoire nécessaire est égal au nombre de nœuds de l'arbre :

$$\frac{4^{r+1} - 1}{3} \quad (\text{IV.5})$$

En revenant au même exemple, pour sauvegarder 64 cellules, nous avons besoin de $\frac{4^4 - 1}{3} = 85$ unités de mémoire qui est supérieur à 64 unités de mémoire utilisées dans une grille classique. Mais l'avantage de l'utilisation des quadtree vient du fait que les cellules qui ne contiennent pas d'information ne seront pas stockées. Donc, dans notre cas, uniquement $1 + 3 + 8 + 15 = 27$ unités de mémoire sont stockées, un nombre bien inférieur au 64 utilisés par la grille classique.

Si nous souhaitons stocker 2 valeurs, le nombre minimal de nœuds nécessaire est égal à 5 si les cellules sont adjacentes, et peut monter jusqu'à $4r + 1$ nœuds si les deux cellules se trouvent à une distance maximale. Dans [Pemmaraju and Shaffer, 1994] ou [Pajarola, 1998] nous retrouvons une analyse approfondie de la problématique de l'espace occupé dont nous présentons uniquement les résultats. Donc pour une résolution $\frac{L}{2^r}$, où L est la dimension du rectangle initial, et r la profondeur de l'arbre, pour sauvegarder l'information de n cellules avec une structure quadtree, le nombre de nœuds, donc d'éléments à sauvegarder, est inférieur à :

$$8n \left[r - \log_4 \left(\frac{n}{2} \right) \right] + \frac{8n}{3} - \frac{1}{3} \approx O(n \log(n)) \quad (\text{IV.6})$$

Nous tenons à souligner le fait que l'espace occupé évolue suivant une loi $O(n \log(n))$. L'avantage de l'utilisation d'une structure de données de type arbre, en plus de l'optimisation de l'espace mémoire, est de pouvoir travailler à des niveaux de résolution différents (multiples de 2) et que même si un arbre a été construit à un niveau de résolution initialement décidé à $\frac{L}{2^r}$, par le simple ajout d'un niveau dans l'arbre, il passe à $\frac{L}{2^{(r+1)}}$. Le même raisonnement peut s'appliquer dans l'augmentation mais également dans la diminution de la résolution. Prenons par exemple la carte présentée dans les figures IV.20(a) à (e). En mesurant le rectangle englobant de la carte

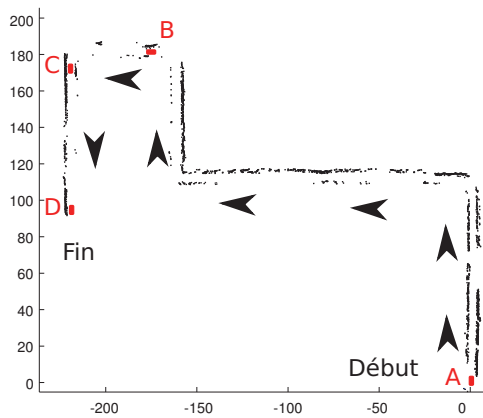
1. Nombre d'arêtes, de la racine au nœud ; 2. La plus grande profondeur d'une feuille de l'arbre ; 3. Nombre de nœuds (en comptant les feuilles ou non) ; 4. La somme des profondeurs de chacune des feuilles ;

d'obstacles, la dimension est de : 25 m × 70 m. Le tableau IV.1 illustre, pour cette carte avec une trajectoire de longueur approximative de 200 m dont 30 m dans le même couloir, le nombre de cellules occupées à différents niveaux de résolution et la capacité de stockage de la structure de donnée en cas d'utilisation d'une grille simple ou d'un quadtree.

Résolution	12,8 m	6,4 m	3,2 m	1,6 m	0,8 m	0,4 m	0,2 m	0,1 m	0,05 m
Cellules / Feuilles occupées	14	28	65	208	481	884	1 347	1 616	1 678
Capacité de la grille	132	266	532	1 062	2 124	4 246	8 492	16 984	33 968
Capacité de l'arbre	4 ³	4 ⁴	4 ⁵	4 ⁶	4 ⁷	4 ⁸	4 ⁹	4 ¹⁰	4 ¹¹

TABLE IV.1 – Analyse de la taille de la carte présentée dans les figures IV.19 et IV.20

Les figures IV.24(a) à (c) illustrent la carte d'un couloir de navigation d'une longueur approximative de 500 m réalisée en environnement extérieur. Par rapport aux exemples illustrés dans les figures IV.17 et IV.21, ce dernier couloir de navigation ne présente pas de boucle, donc des sections superposées. Il est intéressant à analyser cet exemple car la trajectoire a une longueur comparable à la longueur maximale imposée à l'application de voiture autonome (500 m). De plus, l'environnement extérieur est assez riche en obstacles.



(a) Carte d'obstacles



(b) Vue de la scène dans la position B du véhicule



(c) Vue de la scène dans la position C du véhicule

FIGURE IV.24 – Cartographie par capteurs ultrason d'un couloir de navigation d'une longueur approximative de 500 m

Résolution	12,8 m	6,4 m	3,2 m	1,6 m	0,8 m	0,4 m	0,2 m	0,1 m	0,05 m
Cellules / Feuilles occupées	62	144	282	458	1 058	1 769	2 567	2 886	2 931
Nombre de nœuds de l'arbre	137	281	563	1 111	2 169	3 938	6 505	9 391	12 322

TABLE IV.2 – Analyse de la taille de l'arbre pour la carte présentée dans la figure IV.24(a)

En prenant le rectangle englobant de la carte de 230 m × 210 m, pour une grille standard à une résolution de 0,05 m il faudrait stocker 19,32 millions de cellules. Le quadtree de cette carte est composé de 12 322 nœuds. Parmi ces nœuds, 2 931 sont occupés par des feuilles pour la résolution de 0,05 m, (voir tableau IV.2). Comme la place mémoire occupée par cet arbre (17 niveaux de profondeur et résolution de 0,05 m) dans un fichier de format «.mat» est uniquement de 20,2 kB, nous ne trouvons pas nécessaire de chercher à optimiser davantage cet espace mémoire.

IV.3.3 Bilan sur la perception de l'environnement utilisant le système de capteurs ultrason

Comme nous l'avons déjà exprimé, la fréquence de mise à jour de l'information issue du capteur ultrason est de 10 Hz. Ceci a un impact significatif sur l'algorithme de cartographie avec l'augmentation de la vitesse du véhicule. A même titre, la faible fréquence de mise à jour de l'information des obstacles et l'imposition de respect de distance de sécurité pose une contrainte forte sur la vitesse maximale admissible, à ce jour étant de 1.5 m/s. La fréquence d'acquisition ne représente pas un frein dans le développement du produit car le constructeur du SCU a accès à une fréquence bien supérieure. Par conséquent, il suffit de collaborer avec le constructeur dans l'intégration de la solution dans le calculateur du SCU pour éviter ainsi la latence et les limites de communication spécifiques au bus CAN.

Même si les capteurs ultrason n'ont pas été initialement prévus dans l'industrie automobile pour être utilisés dans une application de cartographie de l'environnement, les résultats montrent un énorme potentiel d'exploitation dans ce sens. Pour un environnement riche en obstacles et pour des vitesses de déplacement réduites les cartes reproduisent avec une fidélité importante l'environnement réel. Par contre la distance de perception à 4,095 m de ces capteurs ultrason est considérée petite pour une application de voiturier autonome. Pour que le véhicule puisse réagir dans les temps à toute situation imprévue, nous nous proposons d'étudier une approche complémentaire de détection d'obstacles par vision. Comme les techniques de cartographie de l'environnement du type SfM et type SLAM ont des bases communes mais des buts différents, nous avons initialement introduit les notions de localisation dans le chapitre III. Dans ce chapitre nous détaillerons les aspects liés à la détection de la structure de l'environnement donc à la détection d'obstacles.

IV.4 Enrichissement de la grille d'occupation avec approche *Structure from Motion*

Tout le développement réalisé pour le SLAM-EKF n'a pas été vain car cela nous a permis de comprendre sa complexité du SLAM visuel et de le transformer en un algorithme du type SfM. Nous rappelons que l'acronyme SfM ou Structure from Motion en anglais, désigne une technique de traitement d'images qui permet d'extraire des structures tridimensionnelles à partir de séquences d'images. Les techniques de SLAM et SfM présentent un certain nombre de points communs car les deux techniques se proposent de reconstruire les positions 3D des points observés dans les images successives mais pour des finalités différentes.

Cette technique de SfM est composée généralement de trois étapes :

- extraction des points d'intérêt ;
- appariement de ces points entre deux ou plusieurs images consécutives ;
- estimation du mouvement global des points et de la caméra.

Utilisant une technique d'estimation de la matrice fondamentale à partir de huit points [Languet-Higgins, 1987; Hartley, 1997; Grottschel et al., 2012] et à l'aide d'un processus statistique de consensus (RANSAC¹) [Fischler and Bolles, 1981; Raguram et al., 2008], il est possible de réaliser à la fois une mise en correspondance robuste des points suivi dans plusieurs images, mais également d'estimer la position dans l'espace 3D des points appariés dans les images.

1. RANdom SAmple Consensus;

Dans notre application de voiture autonome, le mouvement de la caméra est contraint par le mouvement du véhicule. La position de la caméra est exprimée au travers du vecteur de translation $\vec{T}_{(v \rightarrow c)}$ et la matrice de rotation $\mathbf{R}_{(v \rightarrow c)}$ de la caméra dans le repère véhicule. L'algorithme SfM a été initialement développé dans une approche de localisation et cartographie simultanée par caméra monoculaire (en anglais, Monocular SLAM - Simultaneous Localisation and Mapping) mais son but a été détourné vers la détection d'obstacles. Pour cela nous présenterons les techniques de détection d'obstacles après avoir détaillé la localisation par technique de SLAM dans la section §III.3.2.

IV.4.1 Algorithme *Structure from Motion*

Avant de présenter la méthode utilisée, nous présentons schématiquement la contrainte épipolaire.

IV.4.1.1 Le rôle de la contrainte épipolaire dans l'estimation de la position 3D d'un point

Dans la figure IV.25(a) nous présentons les éléments géométriques suivants : P point 3D appartenant à l'objet, O_{Im1} et O_{Im2} les centres de projection des deux caméras, p_{Im1} et p_{Im2} respectivement les points de projection dans les images 1 et 2 du point 3D P . Connaissant le point de projection p_{Im1} , il est possible d'affirmer que le point 3D P ne peut se trouver que sur le rayon rectiligne $\overrightarrow{O_{Im1} p_{Im1}}$ dans des positions possibles que nous notons P_1, P_2, P_3, \dots . Par la géométrie épipolaire, on sait que tous ces points P_i se projettent dans l'image 2 sur une même droite, appelée *droite épipolaire* (en bleu dans figure IV.25(a)). Par conséquent, la géométrie épipolaire permet de trouver une relation de mise en correspondance des points observés dans deux images par la même caméra sans connaître explicitement les positions des caméras.

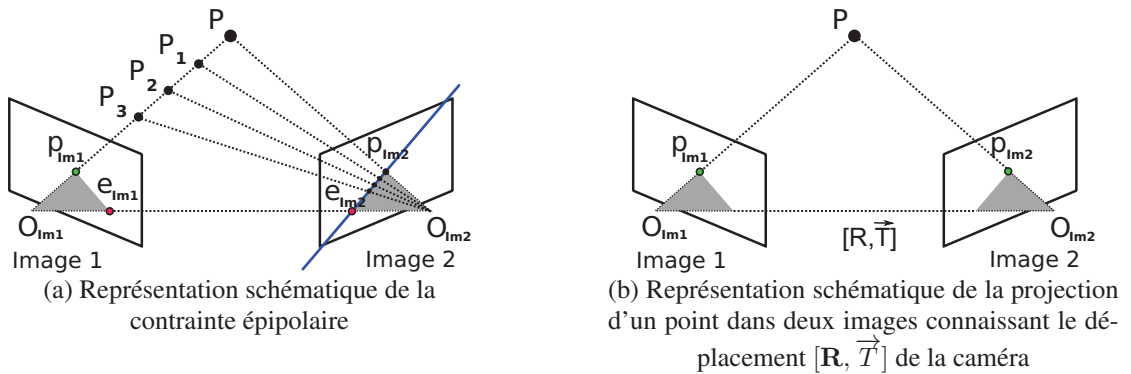


FIGURE IV.25

Soit $\bar{p}_{Imi} = [u_{Imi}, v_{Imi}, 1]^T$ les coordonnées homogènes du point p_{Imi} dans l'image i . Alors,

$$0 = \bar{p}_{Im2}^T \mathbf{F} \bar{p}_{Im1} \quad (\text{IV.7})$$

$$\mathbf{F} = \mathbf{K} \mathbf{E} \mathbf{K}^T \quad (\text{IV.8})$$

$$\mathbf{E} = [\vec{T}]_{\times} \mathbf{R} \quad (\text{IV.9})$$

où \mathbf{F} représente la matrice fondamentale, \mathbf{E} représente la matrice essentielle, $[\vec{T}]_{\times}$ est la matrice antisymétrique associée au vecteur $\vec{T} = [t_1, t_2, t_3]^T$ (cf. équation IV.10), \mathbf{K} la matrice des paramètres intrinsèques pour un modèle sténopé (cf. équation (IV.11)) où (u_0, v_0) sont les coordonnées en pixels du point principal dans l'image, d_x et d_y sont les dimensions élémentaires d'un pixel de la matrice et f est la distance focale.

$$[\vec{T}]_{\times} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \quad (\text{IV.10})$$

$$\mathbf{K} = \begin{bmatrix} \frac{f}{d_x} & 0 & u_0 \\ 0 & \frac{f}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{IV.11})$$

Il est à noter que la matrice fondamentale dépend à la fois des paramètres intrinsèques de la caméra mais aussi des paramètres de rotation et de translation caractérisant le mouvement de la caméra. Par conséquent si nous disposons des positions des points p_{Im1}, p_{Im2} , de la matrice de rotation \mathbf{R} et du vecteur de translation \vec{T} , la position du point 3D P est parfaitement connue [Longuet-Higgins, 1981, 1987].

IV.4.1.2 Estimation des positions 3D des points par la méthode de triangulation

Supposons disposer de deux images, du déplacement subi par la caméra entre les deux prises de vue et d'un ensemble d'inliers¹ sous forme de N paires de points d'intérêt appariées dans les deux images. Suite à une étape de calibrage de la caméra, nous disposons également des matrices des paramètres intrinsèques et extrinsèques avec grande précision ainsi que la matrice de projection exprimée sous la forme :

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \vec{T}] \quad (\text{IV.12})$$

Un point 3D $\bar{P} = [X, Y, Z, 1]^T$ (exprimé en coordonnées homogènes) est observé avec une caméra depuis deux points de vue différents. Soit m et m' les deux points projetés dans le plan image :

$$\begin{aligned} m &= [x_m, y_m, z_m]^T = \mathbf{P} \bar{P}_{3D} \\ m' &= [x'_m, y'_m, z'_m]^T = \mathbf{P}' \bar{P}_{3D} \end{aligned} \quad (\text{IV.13})$$

A cause des erreurs de mesures ou d'appariement la contrainte épipolaire n'est pas satisfaite. Dans ces conditions, une méthode pour calculer les coordonnées du point \bar{P}_{3D} est l'algorithme *Linear triangulation methods* [Hartley and Zisserman, 2003]. Pour éliminer le facteur d'échelle caractérisant les points m ou m' le produit vectoriel suivant est réalisé :

$$m \times (\mathbf{P} \bar{P}_{3D}) = 0 \quad (\text{IV.14})$$

En explicitant ce produit vectoriel, trois équations sont écrites pour chaque point dont uniquement deux sont indépendantes. Par exemple pour la première image les équations s'écrivent ainsi :

$$\begin{cases} x_m (\vec{P}_3^T \bar{P}_{3D}) - (\vec{P}_1^T \bar{P}_{3D}) = 0 \\ y_m (\vec{P}_3^T \bar{P}_{3D}) - (\vec{P}_2^T \bar{P}_{3D}) = 0 \\ x_m (\vec{P}_2^T \bar{P}_{3D}) - y_m (\vec{P}_1^T \bar{P}_{3D}) = 0 \end{cases} \quad (\text{IV.15})$$

1. Dans notre contexte le mot «inliers» désigne les données qui respectent certains critères et que nous qualifierons de «pertinentes». Les données qui ne respectent pas les critères imposés sont nommées «outliers» ou données «aberrantes» ;

où \vec{P}_i^T est la ligne i de la matrice de projection \mathbf{P} . Donc il est possible de retrouver une forme matricielle $\mathbf{A} \vec{P}_{3D} = 0$ où

$$\mathbf{A} = \begin{bmatrix} x_m(\vec{P}_3^T \vec{P}_{3D}) - (\vec{P}_1^T \vec{P}_{3D}) \\ y_m(\vec{P}_3^T \vec{P}_{3D}) - (\vec{P}_2^T \vec{P}_{3D}) \\ x'_m(\vec{P}_3^T \vec{P}_{3D}) - (\vec{P}_1^T \vec{P}_{3D}) \\ y'_m(\vec{P}_3^T \vec{P}_{3D}) - (\vec{P}_2^T \vec{P}_{3D}) \end{bmatrix} \quad (\text{IV.16})$$

L'équation peut être résolue par la méthode de décomposition en valeurs singulières (SVD - Singular Value Decomposition) [Golub and Kahan, 1965] où la matrice \mathbf{A} peut être décomposée ainsi :

$$\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (\text{IV.17})$$

avec \mathbf{U} et \mathbf{V} deux matrices orthonormales et \mathbf{S} une matrice diagonale composée des valeurs singulières. Le vecteur solution de l'équation $\mathbf{A} \vec{P} = 0$ est le vecteur propre à droite de la matrice associée à la valeur propre nulle. La méthode SVD range les valeurs singulières par ordre décroissant donc il s'agit de la dernière colonne \vec{V}_4 de la matrice \mathbf{V} . En final il suffit de normaliser pour éliminer le facteur d'échelle.

$$\vec{P}_{3D} = \frac{\vec{V}_4}{V_{44}} \quad (\text{IV.18})$$

Employant cette technique, un nuage de points 3D est créé et utilisé pour la détection des obstacles. Les résultats correspondant à cette partie seront présentés dans la section §IV.4.4 sur la mise en forme de nuages des points 3D sous forme de macro-obstacles 2D. Par la suite nous détaillerons la méthode d'appariement des points dans des images successives, la technique d'élimination des outliers et la stratégie utilisée pour l'estimation des poses de caméras.

IV.4.2 Appariement et suivi des points

Si dans l'algorithme l'EKF-SLAM précédent, le nombre de points observés dans chaque image n'était pas le but final, dans la détection d'obstacle un grand nombre de points doivent cependant être extraits et observés pour diminuer la probabilité de rater la détection d'un obstacle. La figure IV.26 illustre l'appariement de 3 000 points d'intérêt dans deux images successives utilisant la méthode que nous détaillons par la suite.



FIGURE IV.26 – Appariement des points entre deux images. Les segments rouges indiquent les directions de déplacement des points d'intérêts détectés dans deux images successives du flux vidéo enregistré pendant le déplacement du véhicule

Ce qui peut être facilement constaté est que la distance en pixels entre deux points d'intérêt est relativement faible entre deux images successives, 5-10 pixels pour notre application, dans

les parties très texturées de l'image. Dans ces conditions nous avons constaté qu'il n'est plus nécessaire de construire la zone d'incertitude ou le point sera projeté dans l'image suivante. Nous éliminons cette étape car elle est coûteuse en temps de calcul et peu utile car la zone de recherche calculée initialement est très grande et n'apporte pas d'informations supplémentaires.

La problématique d'extraction de points d'intérêt, de création de leurs descripteurs, du critère et de la méthode d'appariement est très largement débattue et détaillée dans la littérature. Lors du développement et de l'implémentation de la méthode d'appariement nous avons dû faire des choix qui nous semblaient pertinents en temps de calcul et fiabilité d'appariement. Initialement nous avons utilisé dans la méthode de SLAM-EKF des détecteurs des coins du type Harris et Shi-Tomasi. Ces deux détecteurs sont très efficaces et ont prouvé leur efficacité dans le temps mais nous avons voulu intégrer une méthode encore plus rapide. Dans ce cadre [Geiger et al., 2010, 2011] proposent une méthode d'extraction des points d'intérêt et d'appariement de descripteurs basés sur l'utilisation de trois filtres : Checkerboard, Blob et Sobel. Comme ces trois opérateurs s'appliquent sur la même image initiale \mathbf{Im} et le résultat de chacun est indépendant du résultat de l'autre, alors une première optimisation sera réalisée par parallélisation de ces calculs sur trois fils d'exécution différents. Même si cette approche a été initialement conçue pour la stéréovision, le principe d'appariement s'applique également à la monovision. Le choix de détecteur et de descripteur n'est pas unique et le fait d'avoir conçu une architecture logicielle modulaire nous permet à tout moment de facilement interchanger un module avec un autre plus performant, tel que le module de détection et appariement des points d'intérêt dans deux images. Nous n'avons pas réalisé des tests mais il nous semble important de citer le détecteur et le descripteur ORB (Oriented FAST and Rotated BRIEF) de plus en plus utilisé et reconnu par la communauté scientifique [Rublee et al., 2011; Grana et al., 2013; Tian and Shi, 2014]. Par la suite nous décrivons brièvement l'approche de détection et d'appariement de points d'intérêt que nous utilisons. Pour plus de détails voir l'annexe A.6.

Soit $\mathbf{Blob}_{5 \times 5}$ et $\mathbf{Checkerboard}_{5 \times 5}$ deux matrices de taille 5×5 tel que détaillé dans les figures IV.27(a) et (b). $\mathbf{Im}_{\mathbf{Blob}}$ et $\mathbf{Im}_{\mathbf{Checkerboard}}$ sont les deux images résultantes après l'application des deux filtres en vue de la détection des zones et les points d'intérêt dans l'image \mathbf{Im} . La figure IV.27(c) illustre le résultat obtenu pour l'application en parallèle des détecteurs Blob et Checkerboard.

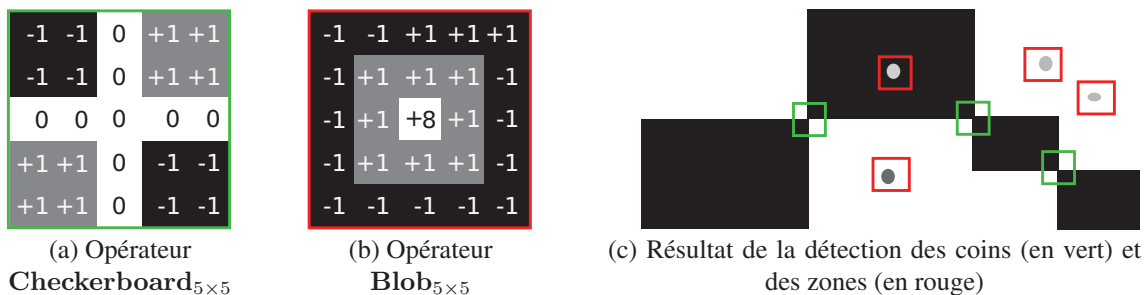


FIGURE IV.27 – Utilisation du détecteur de zones *Blob* et du détecteur de coins *Checkerboard* pour la détection des points d'intérêt dans la figure (c)

Comme nous l'avons déjà précisé, un troisième filtre est utilisé : le filtre de Sobel, généralement employé pour la détection des contours. Celui-ci calcule le gradient de l'intensité de chaque pixel, ce qui indique la direction de la plus forte variation du clair au sombre ainsi que le taux de changement dans cette direction. Notons par \mathbf{G}_h et \mathbf{G}_v les deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point (u, v) de l'image \mathbf{Im} . Les approximations des gradients horizontaux et verticaux peuvent

être combinées en \mathbf{G}_{uv} pour donner une approximation de la norme du gradient :

$$\mathbf{G}_{uv} = \sqrt{\mathbf{G}_{uv,h}^2 + \mathbf{G}_{uv,v}^2}, \quad \forall (u, v) \in \mathbf{Im} \quad (\text{IV.19})$$

Les résultats du filtre de Sobel nous sont utiles dans la construction du descripteur de chaque point d'intérêt. Ainsi pour des raisons d'optimisation du temps de calcul, un descripteur de taille 32 pixels est utilisé. Celui-ci est composé des valeurs choisies dans le voisinage du point d'intérêt, 16 valeurs étant choisies dans la matrice $\mathbf{G}_{horizontal}$ et d'autres 16 valeurs dans la matrice $\mathbf{G}_{vertical}$ suivant un masquage comme celui présenté dans la figure A.10 de l'annexe A.6.

L'utilisation d'un descripteur de taille 32 à la place de descripteurs qui sont généralement de taille 10×10 à 20×20 présente une plus grande sensibilité et un taux plus élevé de faux appariement mais le problème est résolu par l'appariement avec vérification croisée double comme détaillé dans la figure IV.28. Le critère d'appariement utilisé est un critère de distance entre deux vecteurs qui est défini comme la distance à la norme L_1 également appelée distance de Manhattan.

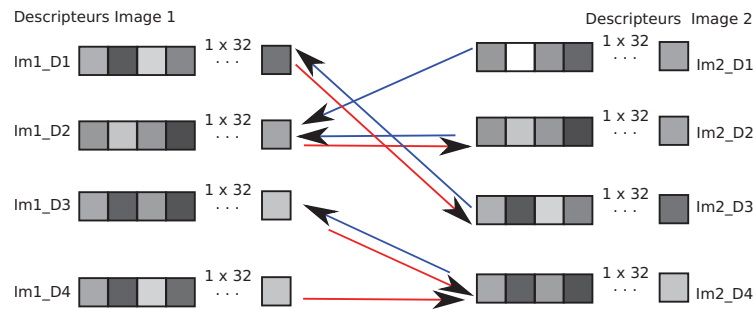


FIGURE IV.28 – Appariement par vérification croisée double : les flèches rouges indiquent les descripteurs $Im_1_D_i$ de l'image \mathbf{Im}_1 qui sont potentiellement les mêmes dans l'image \mathbf{Im}_2 ; inversement les flèches bleues indiquent les descripteurs $Im_2_D_i$ de l'image \mathbf{Im}_2 qui sont potentiellement les mêmes dans l'image \mathbf{Im}_1 . Les paires qui vérifient les deux critères seront considérées des appariements

[Geiger et al., 2010] montrent dans leurs travaux qu'il est possible de raffiner les positions des paires des points appariés. Ils proposent de garder inchangé le descripteur $Im_1_D_i$ de l'image \mathbf{Im}_1 mais de vérifier s'il n'y a pas un autre point (u, v) de l'image \mathbf{Im}_2 dans un rayon de 2 pixels qui potentiellement est situé à une distance de Manhattan plus petite.

Nous avons gardé dans notre approche cette stratégie d'appariement car elle nous a apporté la possibilité d'apparier plus de points d'intérêt dans le même intervalle de temps de calcul tout en apportant de la précision d'appariement par la vérification double croisée. En plus la méthode est facilement parallélisable grâce aux calculs indépendants ce qui nous a permis de mieux distribuer les tâches entre les processeurs et par conséquent de diminuer de plus le temps global de calcul.

IV.4.3 Stratégies d'estimation du déplacement relatif de la caméra

Rappelons-nous que nous disposons d'une méthode d'estimation de la trajectoire par une approche odométrique. Il serait possible de supposer que le véhicule circule sur un sol parfaitement plan et que la caméra ne subit aucun roulis ou tangage et que sa hauteur ne change pas. Calculer les positions des points 3D utilisant tous ces hypothèses fortes est possible mais cela ne reflète pas la réalité. Nous présentons deux stratégies qui nous permettront de retrouver le déplacement de la caméra et ensuite de retrouver les positions 3D des points :

- **stratégie 1** : Estimation du déplacement de la caméra par une approche de minimisation sous contraintes de mouvement imposées par l'odométrie ;
- **stratégie 2** : Estimation du déplacement de la caméra à un facteur d'échelle près et calcul du facteur d'échelle utilisant le déplacement enregistré par odométrie.

IV.4.3.1 Stratégie 1 : Estimation du déplacement de la caméra par une approche de minimisation sous contraintes de mouvement imposées par l'odométrie

Prenons l'exemple didactique illustré dans la figure IV.29. Nous disposons de N points de coordonnées (x_i, y_i) et nous souhaitons trouver la courbe qui interpole le mieux ce jeu de données. Ce problème classique peut être résolu utilisant un algorithme de minimisation du type Levenberg-Marquardt [Moré, 1978; Bartoli, 2003]. Pour illustrer cet algorithme, nous imposons comme contrainte d'interpolation une courbe du type $y(x) = ax^2 + bx + c$, nous définissons les paramètres à optimiser $X = [a, b, c]^T$ avec $X_0 = [a_0, b_0, c_0]^T$ les valeurs au démarrage de l'algorithme, un critère de minimisation du type somme d'erreur quadratique (équation (IV.20)) et un critère d'arrêt égal à une erreur de minimisation recherchée $\epsilon < 10^{-3}$.

$$\epsilon = \sum_{i=1}^N (y_i - \hat{y}(x_i))^2 \quad (\text{IV.20})$$

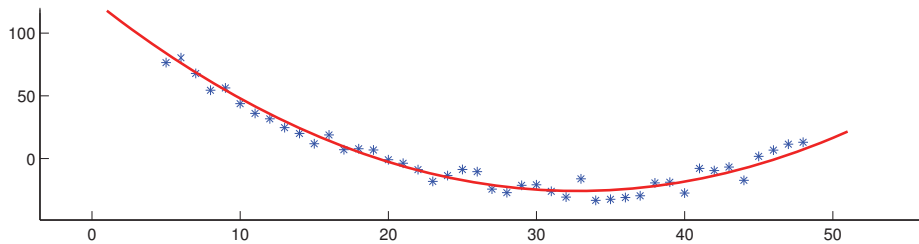


FIGURE IV.29 – Estimation de la courbe d'interpolation par minimisation Levenberg–Marquardt ; en rouge la courbe obtenue, en bleu le jeu de points initialement disponible

Sur le même principe de minimisation il est possible de créer des fonctions de coût plus complexes qui de par leur minimisation nous permettront d'estimer les paramètres définissant la roto-translation de la caméra. Rappelons-nous la contrainte épipolaire (équation (IV.7)) :

$$\bar{p}_{Im2}^T \mathbf{F} \bar{p}_{Im1} = 0 \quad (\text{IV.21})$$

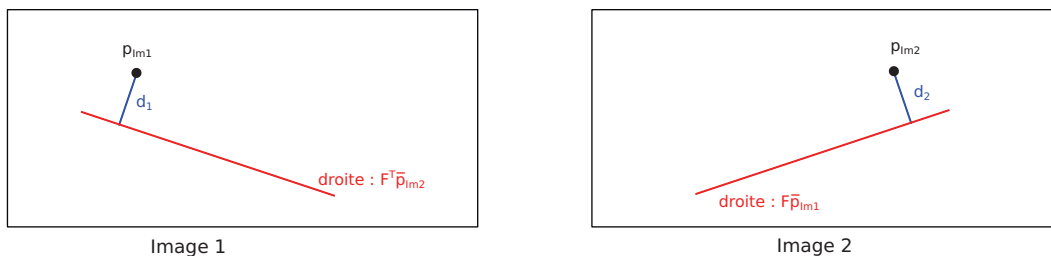


FIGURE IV.30 – Illustration de la distance épipolaire

La figure IV.30 illustre une situation de contrainte épipolaire non respectée où la distance entre un point et la droite épipolaire est non nulle. Pour la simplicité du calcul notons $\mathbf{F} \bar{p}_{Im1} = [a, b, c]^T$ la droite d'équation $ax + by + c = 0$. Alors la distance d_2 entre le point

\bar{p}_{Im2} de coordonnées $[x_a, y_b, 1]^T$ et la droite $(\mathbf{F} \bar{p}_{Im1})$ est donnée par l'équation (IV.22).

$$d_2 = d(\bar{p}_{Im2}, (\mathbf{F} \bar{p}_{Im1})) = \frac{|ax_a + by_b + c|}{\sqrt{a^2 + b^2}} \quad (\text{IV.22})$$

L'équation s'écrit également sous la forme suivante :

$$d_2 = \frac{\bar{p}_{Im2}^T \mathbf{F} \bar{p}_{Im1}}{\sqrt{(\mathbf{F} \bar{p}_{Im1})_1^2 + (\mathbf{F} \bar{p}_{Im1})_2^2}} \quad (\text{IV.23})$$

avec $(\mathbf{F} \bar{p}_{Im1})_i^2$ le carré de la $i^{\text{ème}}$ valeur du vecteur $\mathbf{F} \bar{p}_{Im1}$. Estimer la matrice fondamentale \mathbf{F} revient alors à minimiser une fonction de coût de type :

$$\sum_{i=1}^N d(\bar{p}_{Im2,i}, \mathbf{F} \bar{p}_{Im1,i})^2 = \sum_{i=1}^N \frac{(\bar{p}_{Im2,i}^T \mathbf{F} \bar{p}_{Im1,i})^2}{(\mathbf{F} \bar{p}_{Im1,i})_1^2 + (\mathbf{F} \bar{p}_{Im1,i})_2^2} \quad (\text{IV.24})$$

Une autre possibilité est de minimiser des distances épipolaires moyennes symétrique dans les deux images (*average symmetric epipolar distance*, en anglais) :

$$\begin{aligned} & \frac{1}{N} \sum_{i=1}^N [d(\bar{p}_{Im2,i}, \mathbf{F} \bar{p}_{Im1,i})^2 + d(\bar{p}_{Im1,i}, \mathbf{F}^T \bar{p}_{Im2,i})^2] = \\ & = \frac{1}{N} \sum_{i=1}^N \left[(\bar{p}_{Im2,i}^T \mathbf{F} \bar{p}_{Im1,i})^2 \left(\frac{1}{(\mathbf{F} \bar{p}_{Im1,i})_1^2 + (\mathbf{F} \bar{p}_{Im1,i})_2^2} + \frac{1}{(\mathbf{F}^T \bar{p}_{Im2,i})_1^2 + (\mathbf{F}^T \bar{p}_{Im2,i})_2^2} \right) \right] \end{aligned} \quad (\text{IV.25})$$

Même si la distance moyenne symétrique est plus précise, elle est plus coûteuse en temps de calcul. Par conséquent, nous considérons la forme la plus simple de fonction de coût à minimiser. Partant de l'équation (IV.22), on définit la distance épipolaire D_{EP} pour toute paire de points $\{\bar{m}_{Im1}, \bar{m}_{Im2}\}$ de l'espace normalisé suivant l'équation (IV.26) :

$$D_{EPi} = \bar{m}_{Im1}^T \mathbf{E} \bar{m}_{Im2} \quad (\text{IV.26})$$

où $\bar{m} \sim \mathbf{K}^{-1} \bar{p}$ sont les coordonnées homogènes du point 3D P dans le plan normalisé, $\mathbf{E} = [\vec{T}^n]_{\times} \mathbf{R}$ représente la matrice essentielle et \vec{T}^n le vecteur de translation normalisé.

Comme pour le cas précédent, nous avons besoin de certains éléments pour alimenter l'algorithme de minimisation Levenberg–Marquardt. D'abord nous construisons le vecteur à estimer $X = [t_x, t_y, t_z, \alpha_x, \alpha_y, \alpha_z]^T$ composé de six paramètres définissant la roto-translation subie par la caméra entre les deux prises de vues. Nous mentionnons que cette approche permet de retrouver la translation de la caméra à un facteur d'échelle près. Nous montrons qu'il est possible de retrouver ce facteur d'échelle en ajoutant des contraintes supplémentaires dans la fonction de coût à minimiser ou nous pouvons extraire le facteur d'échelle utilisant l'information odométrique. Pour assurer une plus rapide convergence de l'algorithme, nous initialisons le vecteur avec $X_0 = [t_{x0}, t_{y0}, t_{z0}, \alpha_{x0}, \alpha_{y0}, \alpha_{z0}]^T$ avec des valeurs cohérentes au déplacement réel : roulis, tangage et translation verticale approximativement nuls. Les autres trois paramètres (t_{x0}, t_{y0}, t_{z0}) seront initialisés en cohérence avec le déplacement de la caméra mesuré à travers l'information odométrique. Enfin nous souhaitons résoudre le problème non linéaire d'interpolation d'une courbe sous la forme :

$$\min_X \|f(X)\|_2^2 = \min_X (f_1(X)^2 + f_1(X)^2 + \dots + f_N(X)^2) \quad (\text{IV.27})$$

Pour notre situation $f_i(X)$ représente la distance épipolaire D_{EPi} définie suivant l'équa-

tion (IV.26). L'expression à minimiser devient alors :

$$\min_X \|f(X)\|_2^2 = \min_X \|f_{DEP}(X)\|_2^2 = \min_X (D_{EP1}(X)^2 + D_{EP2}(X)^2 + \dots + D_{EPN}(X)^2) \quad (IV.28)$$

avec N le nombre de paires des points appariés.

Très souvent le minimum global n'est pas atteint à cause des minimums locaux. Il est possible d'ajouter des contraintes sur X de type :

$$\limInf_k < X(k) < \limSup_k, \quad k \in \overline{1,6} \quad (IV.29)$$

Ce type de contrainte est appelée *bounding box* en anglais, ce qui revient à créer une solution se trouvant dans un hyper-rectangle englobant de sorte que les valeurs obtenues soient en corrélation avec un mouvement admissible de la caméra. Les contraintes sont prises en compte plutôt comme des conditions de test. Par conséquent cette méthode ne permet pas d'éviter les minimums locaux situés dans l'hyper-rectangle englobant. Par conséquent si nous souhaitons que les a priori sur le déplacement de la caméra soient pris en compte dans la convergence de la fonction, alors il faudrait les ajouter directement dans la fonction de coût. Pour notre application cela se réécrit suivant l'équation (IV.30) :

$$\min_X \left(\frac{\|f_{DEP}(X)\|_2^2}{N} + \frac{\|f_{Longi}(X)\|_2^2}{\sigma_{Longi}^2} + \frac{\|f_h(X)\|_2^2}{\sigma_h^2} + \frac{\|f_{\alpha x}(X)\|_2^2}{\sigma_{\alpha x}^2} + \frac{\|f_{\alpha y}(X)\|_2^2}{\sigma_{\alpha y}^2} \right) \quad (IV.30)$$

où $f_{Longi}(X)$ représente la différence entre le déplacement longitudinal mesuré par odométrie et celui estimé par vision, $f_h(X)$ l'erreur estimée sur le déplacement vertical par rapport à un déplacement nul, $f_{\alpha x}(X)$ et $f_{\alpha y}(X)$ sont les erreurs de roulis respectivement de tangage estimées par rapport à des valeurs nulles. Pour contrebalancer l'effet du poids de chacune des cinq fonctions dans la composition du $f(X)$ nous utilisons des pondérations : σ_{Longi}^2 , σ_h^2 , $\sigma_{\alpha x}^2$ et respectivement $\sigma_{\alpha y}^2$. Rappelons-nous que cette méthode permet d'estimer les trois paramètres de rotation et les trois paramètres de translation mais ces trois dernières sont à un facteur d'échelle près. Connaissant le déplacement longitudinal réalisé par le véhicule et en réalisant l'hypothèse que le sol est approximativement plan entre deux poses de la caméra, alors le terme $f_{Longi}(X)$ introduira la contrainte de déplacement dans la fonction à minimiser. Bien entendu, le facteur d'échelle peut être récupéré également après la minimisation de la fonction de coût.

Pour analyser les performances de l'algorithme sans introduire des erreurs dues au mauvais appariement nous réalisons la procédure de test que nous détaillons par la suite. Pour la simplicité des calculs, nous considérons la caméra initialement positionnée dans l'origine d'un repère global. Par RT_{12} est noté le déplacement relatif de la caméra entre deux poses. Un ensemble de N points 3D est construit en simulation tout en déterminant leurs projections dans les images. Pour que l'espace de recherche dans cette simulation soit en concordance avec l'environnement dans lequel évolue le véhicule, nous imposons les limites du parallélépipède rectangle à l'intérieur duquel se situent les N points 3D, comme illustré par la figure IV.31(a).

$$\begin{aligned} \text{Limites de l'environnement extérieur : } & \begin{cases} x \in [-60, 60] \text{ m} & \text{latéral} \\ y \in [-1, 335, 20] \text{ m} & \text{hauteur} \\ z \in [3, 60] \text{ m} & \text{axe de déplacement} \end{cases} \\ \text{Limites de l'environnement intérieur : } & \begin{cases} x \in [-15, 15] \text{ m} & \text{latéral} \\ y \in [-1, 335, 4.5] \text{ m} & \text{hauteur} \\ z \in [3, 40] \text{ m} & \text{axe de déplacement} \end{cases} \end{aligned}$$

Suivant les limites imposées, nous réalisons des tirages aléatoires. Pour chaque triplet $\{x, y, z\}$ ainsi constitué en coordonnées du point 3D, il est testé que ses projections se

situent bien à l'intérieur des limites des deux images (voir figure IV.31(b)). Si c'est le cas, le point est validé. L'algorithme s'arrête quand un nombre N de points 3D ayant les propriétés imposées ont été validés.

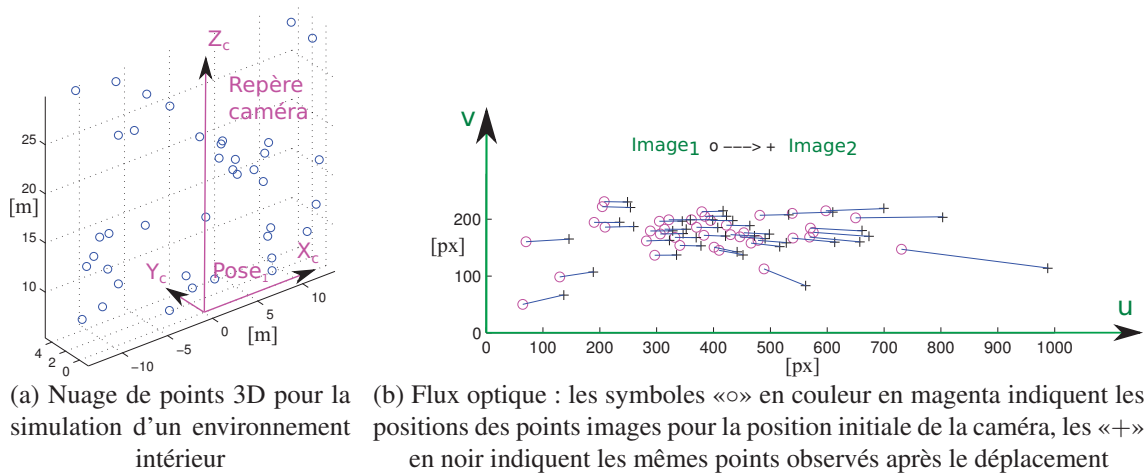


FIGURE IV.31

Lors de l'application de l'algorithme de minimisation Levenberg–Marquardt pour la fonction de coût (IV.30) nous constatons une convergence de l'algorithme après 50 - 70 étapes de réévaluation de la fonction de coût. Pour notre exercice la condition d'arrêt est atteinte après 67 étapes, comme l'illustre la figure IV.32(a). Grâce au jeu de données construit, nous disposons des vraies positions 3D des points, ce qui nous permet de vérifier que les points 3D projetés dans les deux images utilisant déplacement réel et estimé de la caméra sont en concordance, voir figure IV.32(b).

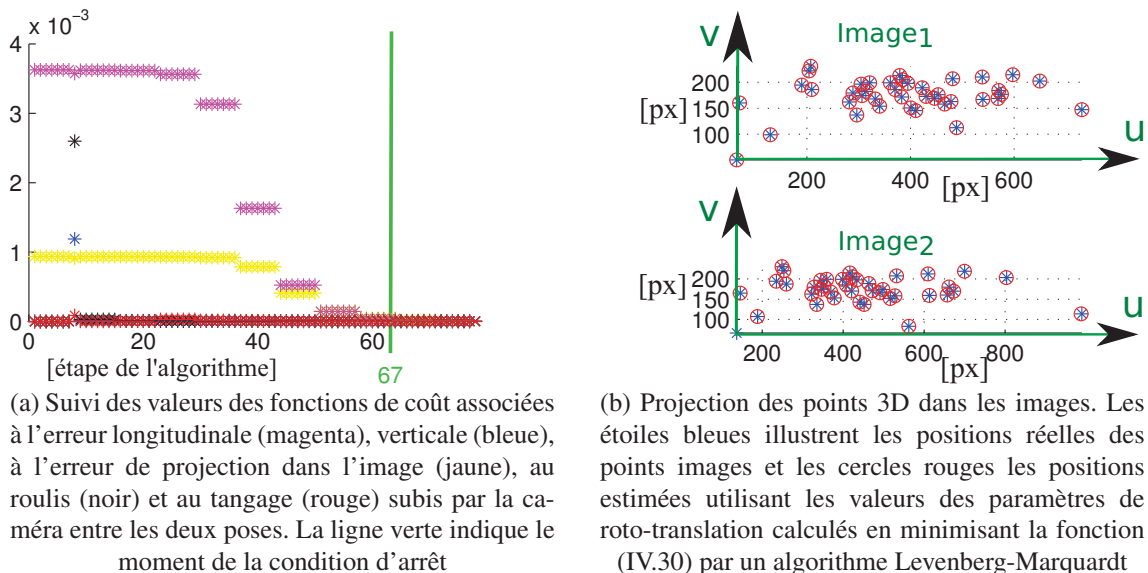


FIGURE IV.32

Sur le même principe de minimisation, il est possible d'estimer simultanément $3N + 6$ paramètres : les 6 paramètres de translation et rotation de la caméra mais également les positions 3D des N points. Ce principe porte le nom d'ajustement de faisceaux ou *Bundle Adjustment* en anglais. En gardant l'approche proposée dans les deux exemples antérieurs, pour estimer les positions des points 3D, il suffit d'ajouter dans le vecteur de paramètres à optimiser les $3 \times N$

coordonnées. Le vecteur X devenant ainsi :

$$X = [t_x, t_y, t_z, \alpha_x, \alpha_y, \alpha_z, x_1, y_1, z_1, \dots, x_N, y_N, z_N]^T \quad (\text{IV.31})$$

En pratique l'optimisation de cette fonction est encore plus coûteuse que la précédente ce qui s'avère à être une tâche difficilement réalisable en temps réel. De plus il faudrait minimiser sur plusieurs images successives, ce qui demande un ajout de paramètres représentant les roto-translations relatives des caméras. Nous constatons qu'il y a le risque qu'un minimum local soit atteint avant que les coordonnées 3D ne soient bien estimées. En pratique une étape intermédiaire est proposée pour créer le vecteur de valeurs initiales qui va initialiser l'algorithme avec un X_0 très proche du minimum global recherché.

En conclusion, nous ne choisissons pas cette stratégie basée sur la minimisation d'une fonction de coût pour retrouver les 6 paramètres de roto-translation de la caméra entre deux poses pour plusieurs raisons. D'abord, cette approche demande à minimiser une fonction coûteuse. En plus, le temps de calcul peut varier du simple au triple en fonction de la direction de descente de gradient initialement choisie. Encore plus contraignant est de tomber sur un minimum local. Enfin, la contrainte la plus forte est l'intolérance aux mauvais appariements (outliers). Une paire de points donnée en entrée à un effet de levier important dans la fonction à minimiser. Il y a des techniques qui permettent d'identifier les éléments qui ont un effet de levier très important mais ces techniques sont encore plus coûteuses en termes de temps de calcul et inadaptées au traitement temps réel. Par conséquent, pour pallier le problème il faut impérativement utiliser en plus une technique d'élimination d'outliers.

Nous verrons que la deuxième stratégie que nous présentons par la suite réalise l'estimation robuste de la matrice essentielle (ou fondamentale) par l'utilisation d'une approche statistique. De plus, à la fin du processus nous disposerons également des paires des points mal appariés dans les deux images (outliers).

IV.4.3.2 Stratégie 2 : Estimation du déplacement de la caméra à un facteur d'échelle près et calcul du facteur d'échelle utilisant le déplacement enregistré par odométrie

Supposons disposer d'un ensemble de N appariements dans deux images. Trouver le déplacement de la caméra revient à trouver la matrice fondamentale \mathbf{F} qui respecte la contrainte épipolaire pour l'ensemble de points. Pour la suite notons \mathbf{F} ainsi :

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{F}_{13} \\ \mathbf{F}_{21} & \mathbf{F}_{22} & \mathbf{F}_{23} \\ \mathbf{F}_{31} & \mathbf{F}_{32} & \mathbf{F}_{33} \end{bmatrix} \quad (\text{IV.32})$$

Alors pour le point i la contrainte épipolaire s'écrit : $\vec{p}_{Im_1,i}^T \mathbf{F} \vec{p}_{Im_2,i}^T = 0$ ou sous forme développée :

$$\begin{bmatrix} x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{F}_{13} \\ \mathbf{F}_{21} & \mathbf{F}_{22} & \mathbf{F}_{23} \\ \mathbf{F}_{31} & \mathbf{F}_{32} & \mathbf{F}_{33} \end{bmatrix} \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = 0 \quad (\text{IV.33})$$

$$\begin{bmatrix} x_i x'_i & x_i y'_i & x_i & y_i x'_i & y_i y'_i & y_i & x'_i & y'_i & 1 \end{bmatrix} \vec{F} = 0 \quad (\text{IV.34})$$

où $\vec{F} = [\mathbf{F}_{11} \ \mathbf{F}_{12} \ \mathbf{F}_{13} \ \mathbf{F}_{21} \ \mathbf{F}_{22} \ \mathbf{F}_{23} \ \mathbf{F}_{31} \ \mathbf{F}_{32} \ \mathbf{F}_{33}]^T$

En répétant le raisonnement pour tous les N points, un système d'équations s'écrit ainsi :

$$\begin{bmatrix} x_1x'_1, & x_1y'_1, & x_1, & y_1x'_1, & y_1y'_1, & y_1, & x'_1, & y'_1, & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_Nx'_N, & x_Ny'_N, & x_N, & y_Nx'_N, & y_Ny'_N, & y_N, & x'_N, & y'_N, & 1 \end{bmatrix} \vec{F} = 0 \quad (\text{IV.35})$$

L'équation ainsi écrite sous la forme $\mathbf{A} \vec{F} = 0$ peut être résolue par la méthode SVD (Singular Value Decomposition) [Golub and Kahan, 1965] :

$$\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (\text{IV.36})$$

avec \mathbf{U} et \mathbf{V} deux matrices orthonormales et \mathbf{S} une matrice diagonale composée des valeurs singulières positives triées en ordre décroissant. Pour un système surdéterminé, comme celui-ci, la matrice \mathbf{A} a plus de lignes que de colonnes. Par conséquent, la solution du système est donnée au sens des moindres carrés.

A cet instant nous disposons de deux méthodes distinctes pour calculer la matrice fondamentale \mathbf{F} . Dans la littérature l'approche SVD est considérée plus stable et plus rapide qu'une minimisation itérative par la méthode de Levenberg-Marquardt dans cette situation. En conclusion nous utiliserons l'approche SVD à faible nombre de points combinée avec une approche stochastique par tirage aléatoire. Cette combinaison nous permettra à la fois de déterminer le déplacement de la caméra mais également d'identifier des points outliers.

IV.4.3.3 Stratégie choisie : Estimation du déplacement de la caméra par une approche stochastique

Pour le système d'équations présenté par l'équation (IV.35), si nous choisissons plus de 8 points, alors la solution donnée sera une optimisation au sens des moindres carrés. Par contre si nous utilisons uniquement 8 points pour résoudre le système d'équations, nous employons *l'algorithme de huit points* (*eight-point algorithm* en anglais) proposé par [Longuet-Higgins, 1987]. Dans cette situation nous disposons de 8 équations et 9 inconnues. Par conséquent la neuvième valeur singulière S_{99} sera nulle, ce qui amène à trouver la solution recherchée dans la dernière colonne de la matrice \mathbf{V} .

Cet algorithme nous permet de retrouver la matrice fondamentale \mathbf{F} (ou bien la matrice essentielle \mathbf{E} si nous disposons des paramètres de calibration intrinsèques), donc du déplacement de la caméra entre deux poses à un facteur d'échelle près. Comme les appariements ne sont pas parfaits, nous faisons appel à une méthode robuste d'association des points entre les deux images. Il s'agit d'une estimation de la matrice fondamentale par approche stochastique du type RANSAC (RANDOM SAMPLE CONSENSUS). Pour mieux expliquer le concept, nous détaillons dans l'annexe A.7 un exemple simple de calcul des paramètres d'une droite qui approxime le mieux un nuage de points donné.

Nous prenons les notations suivantes :

- n le nombre minimum de données nécessaires pour ajuster le modèle (8 paires de points dans notre cas) ;
- k le nombre d'itérations de l'algorithme ;
- p la probabilité que l'algorithme RANSAC produit un résultat correct lorsqu'il sélectionne les n points à une itération ;

- w la probabilité de choisir un point pertinent à chaque fois qu'un seul point est sélectionné. Cela se réécrit sous la forme de l'équation (IV.37) et représente le taux d'inliers dans l'ensemble de données :

$$w = \frac{\text{nombre de points pertinents dans les données}}{\text{nombre de points dans les données}} \quad (\text{IV.37})$$

Cependant, le nombre d'itérations k peut être déterminé à partir d'un résultat théorique. En supposant que les n points nécessaires pour l'estimation d'un modèle soient sélectionnées de manière indépendante, w^n correspond à la probabilité que l'ensemble de n points correspondent à des valeurs pertinentes. Le complément $1 - w^n$ correspond à la probabilité qu'au moins un des n points soit un cas aberrant. Sachant que l'algorithme RANSAC exécutera k itérations indépendantes, alors $(1 - w^n)^k$ est la probabilité que l'algorithme ne choisisse jamais un ensemble de n points qui seraient tous pertinents. Par conséquent,

$$(1 - w^n)^k = 1 - p \quad (\text{IV.38})$$

avec p imposé généralement à 0,99. Cela nous permet d'extraire k comme suit :

$$k = \frac{\log(1 - p)}{\log(1 - w^n)} \text{ itérations} \quad (\text{IV.39})$$

Par extension, nous appliquons le même principe RANSAC pour estimer statiquement le plus probable déplacement de la caméra entre deux poses. Par tirage aléatoire de groupes de 8 points, la matrice fondamentale est calculée utilisant l'approche SVD précédemment exposée. Pour chaque matrice \mathbf{F} ainsi calculée, le nombre des inliers est compté suivant un critère de distance épipolaire (ou bien de distance de Sampson) inférieure à un seuil imposé. Comme solution sera choisie la matrice \mathbf{F} pour laquelle le nombre d'inliers détectés est maximal. Pour raffiner ce résultat, une autre matrice \mathbf{F}_{max} est calculée par la décomposition SVD mais cette fois-ci utilisant tous les inliers obtenus dans l'étape précédente. La figure IV.33 illustre en couleur verte le déplacement des points d'intérêt de type inliers suivant le déplacement de la caméra, en rouge les outliers et en jaune les droites épipolaires.

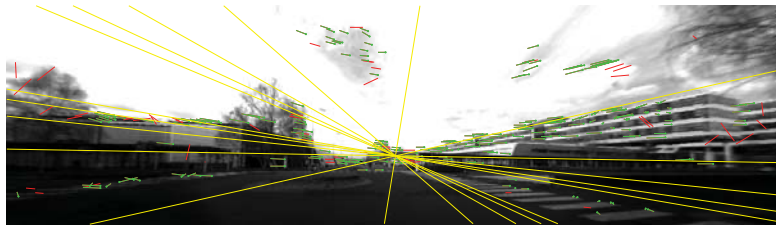


FIGURE IV.33 – Estimation du déplacement de la caméra par la méthode des huit points, la couleur verte représente les paires de points d'intérêt déterminés comme inliers, en rouge les outliers et en jaune les droites épipolaires

En utilisant cette méthode d'estimation de la pose de la caméra par l'approche stochastique, l'élimination des outliers est implicite.

IV.4.3.4 Optimisation du temps de calcul par ajout d'une étape de prétraitement

Même si l'utilisation de l'appariement par vérification double croisée diminue considérablement le taux de faux appariements, il peut rester des *outliers*. Dans une première étape nous faisons appel à la librairie proposée par [Shewchuk, 1996] sur la détection des outliers utilisant une approche

de décomposition de l'espace des points d'intérêt dans l'image courante dans des triangles. La méthode utilisée est la triangulation de Delaunay que nous avons déjà appliquée dans l'algorithme $\chi - Shape$ pour le décèlement de formes, voir l'annexe A.2. La figure IV.34 illustre le maillage obtenu par suite à l'application de la méthode Delaunay sur l'ensemble des points d'intérêt.

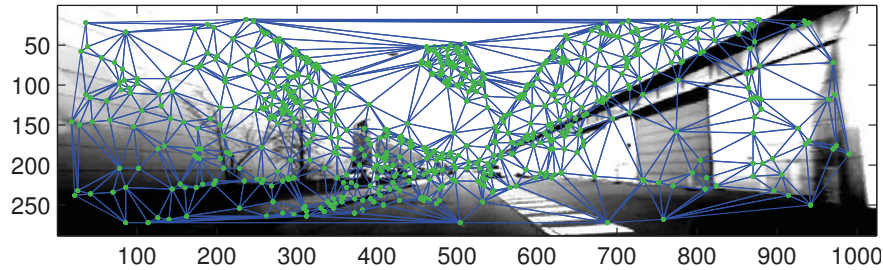


FIGURE IV.34 – Construction d'un maillage des points d'intérêt par la méthode de triangulation de Delaunay

Le principe d'élimination des outliers repose sur l'idée simple qu'un maillage réalisé sur un ensemble de points appartenant à l'image a évolué vers une contraction ou expansion tout en gardant approximativement la même décomposition de maillage [Lhuillier and Quan, 1999; Dou and Li, 2012; Zhang et al., 2013; Chen et al., 2013]. La figure IV.35 illustre une situation de détection de deux paires d'outliers ($p_8 - p_2$ et $p_5 - p_9$) marquées par des segments/«fils» rouges. Dans la littérature, nous constatons une utilisation relativement réduite de la méthode d'élimination des paires d'outliers par la méthode Delaunay. En échange l'utilisation d'un processus de consensus statistique du type RANSAC pour l'analyse de la cohérence du mouvement des points en corrélation avec le mouvement de la position de la caméra est le plus utilisé [Fischler and Bolles, 1981; Raguram et al., 2008; Derpanis, 2010]. Nous reviendrons sur ce principe que nous utilisons également.

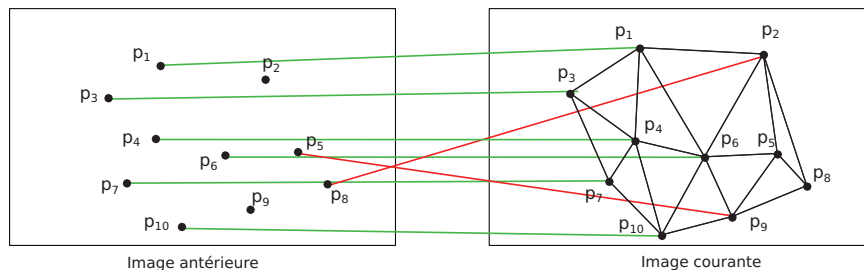


FIGURE IV.35 – Détection des mauvais appariements des points dans deux images successives par la méthode de triangulation de Delaunay. Les segments en rouge mettent en évidence les paires des points $p_8 - p_2$ et $p_5 - p_9$ détectés comme outliers

En analysant le temps d'extraction de 2 000 points d'intérêt cumulé au temps d'appariement de ces points dans l'image précédente, au temps de raffinement des positions des appariements et à l'élimination d'outliers par la triangulation de Delaunay, nous mesurons un temps de calcul total de **5 à 7 ms**.

A cet instant nous disposons d'une méthode plus rapide d'extraction de points d'intérêt dans une image, de mise en commun des points dans deux images successives et d'élimination d'outliers par une méthode purement géométrique. L'application de ces algorithmes pour tous les images permet de garder un suivi des déplacements des points d'intérêt à travers les images successives, voir figure IV.36. Notons par $Trace_{p_i}$ la trace du point image p_i dans plusieurs images. Cela permet aux algorithmes d'estimation des points 3D de calculer et éventuellement d'affiner leurs positions.

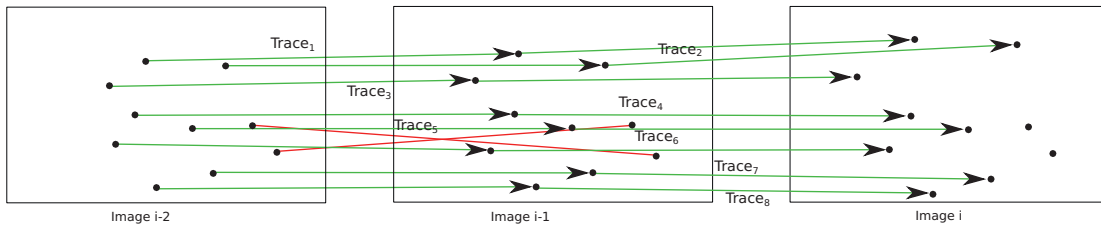


FIGURE IV.36 – Suivi des points d'intérêt sur plusieurs images consécutives («traces» des points)

Comparaison des résultats

Par la suite nous nous proposons d'illustrer le gain de temps de calcul réalisé sur l'algorithme SfM par l'utilisation d'une étape de prétraitement d'élimination des outliers par la méthode de Delaunay. Pour cela nous prenons un même jeu de données de 2 500 images et pour chaque paire d'images consécutives nous disposons de deux ensembles de points d'intérêt appartenant à chaque image et de leurs appariements. Dans l'exemple didactique de l'annexe A.7 il a été facile de calculer le paramètre w correspondant à la probabilité de choisir un point pertinent à chaque fois qu'un seul point est sélectionné. Dans notre cas, nous ne pouvons pas mesurer w de la même manière. Pour déterminer le taux d'inliers (ou d'outliers) avec une grande précision, nous paramétrons exceptionnellement l'algorithme RANSAC pour qu'il réalise 20 000 itérations indépendantes pour chaque couple d'images. Ensuite nous réalisons cet essai avec et sans prétraitement d'outliers par la méthode de Delaunay.

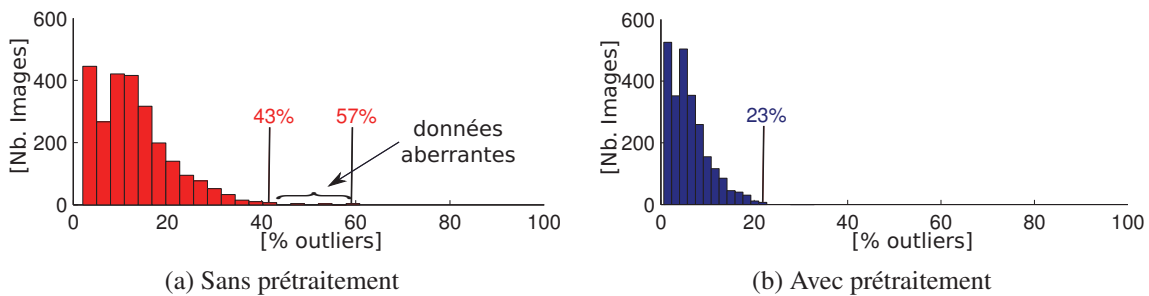


FIGURE IV.37 – L'histogramme du pourcentage d'outliers détectés dans les 2 500 paires d'images par l'algorithme RANSAC pour les deux situations d'utilisation ou non d'un prétraitement par la méthode Delaunay

Comme il était attendu et illustré dans les histogramme des figures IV.37(a) et (b), ne pas réaliser un prétraitement implique de trouver plus d'outliers dans les images. Utilisant ces histogrammes, nous pouvons extraire le taux d'inliers correspondant au paramètre w et de déterminer le nombre d'itérations théorique k pour l'algorithme RANSAC.

Pour l'analyse du temps de calcul, nous avons paramétré l'algorithme RANSAC (sans prétraitement) à un nombre d'itérations $k = 4000$ conformément à l'équation (IV.40). Pour cette situation, nous avons pu constater que l'algorithme a besoin d'environ 50 ms pour accomplir sa tâche. Analyser par la méthode RANSAC plus de données en sachant que statistiquement elles sont composées d'un taux plus important d'outliers c'est une perte de temps. Lors des essais nous avons mesuré que l'étape de prétraitement Delaunay nécessite 5 à 7 ms pour analyser en moyenne 2 000 paires de points d'intérêt. Même si cela représente un coût supplémentaire, nous avons également constaté que pour le même nombre d'itérations $k = 4000$, globalement l'algorithme RANSAC est plus rapide de 5% à 10% en utilisant la phase de prétraitement. Ce qui s'explique par une baisse du nombre de distances épipolaires à analyser pour décider si un point est un inlier

ou un outlier.

$$k_{57\%} = \frac{\log(1 - 0,99)}{\log(1 - (1 - 0,57)^8)} \approx 4\,000 \text{ itérations} \quad (\text{IV.40})$$

$$k_{43\%} = \frac{\log(1 - 0,99)}{\log(1 - (1 - 0,43)^8)} \approx 410 \text{ itérations} \quad (\text{IV.41})$$

$$k_{23\%} = \frac{\log(1 - 0,99)}{\log(1 - (1 - 0,23)^8)} \approx 34 \text{ itérations} \quad (\text{IV.42})$$



FIGURE IV.38 – Comparaison du temps de calcul pour les deux approches RANSAC sans puis avec utilisation d'une étape de prétraitement d'élimination d'outliers par la méthode de Delaunay ; en jaune est illustré le temps de calcul utilisé dans l'étape de prétraitement par la méthode Delaunay ; en bleue est désigné le temps de calcul utilisé par l'algorithme RANSAC utilisant uniquement 100 itérations

Mais un gain considérable se fait par le changement du paramètre k . Les résultats théoriques illustrés dans les équations (IV.40) à (IV.42) soutiennent notre affirmation que la diminution du nombre d'itérations k n'affecte pas le résultat global. Effectivement les essais montrent que la diminution du nombre d'itérations de $k = 4\,000$ à $k = 100$ n'a pas un effet ressenti sur la qualité des résultats. Nous avons préféré choisir $k = 100$ et pas $k = 34$ car le gain de temps de calcul de l'algorithme RANSAC d'un facteur 40 est déjà suffisamment important, passant de **50 ms à 1-2 ms**. En plus choisir le $k = 100$ nous permet de travailler avec des jeux de données qui ont en composition jusqu'à 32% d'outliers tout en assurant la même probabilité de calculer un bon résultat. Dans le cas de l'utilisation d'une étape de prétraitement, il faut tenir compte de son temps de calcul lors de la comparaison des résultats. Ainsi, la figure IV.38 montre le temps total de l'algorithme RANSAC pour les deux approches.

Nous disposons donc à cet instant d'une méthode nous permettant de calculer le déplacement de la caméra entre deux poses à un facteur d'échelle près. En connaissant le déplacement du véhicule par odométrie et en faisant l'hypothèse que le sol est plat sur des courtes distances, nous retrouvons le facteur d'échelle. Par la suite nous détaillons la méthode d'estimation des positions des points 3D qui seront associées à la grille d'occupation.

IV.4.4 Mise en forme de nuages des points 3D sous forme de macro-obstacles 2D

Nous cherchons dans cette partie à reconnaître les obstacles de l'environnement et pour ce faire, nous utilisons les points 3D issus de l'algorithme SfM dont la hauteur est située dans l'intervalle $[0,1, 1,8]$ m. Ces points 3D seront projetés sur le plan du sol et uniquement les coordonnées 2D seront retenues dans la carte d'obstacles.

Dans cette approche les hypothèses suivantes sont utilisées :

- Le sol est approximativement plan ;
- Le véhicule se déplace à faible vitesse (1 m/s à 2 m/s) ;
- La texture de l'environnement est riche ;

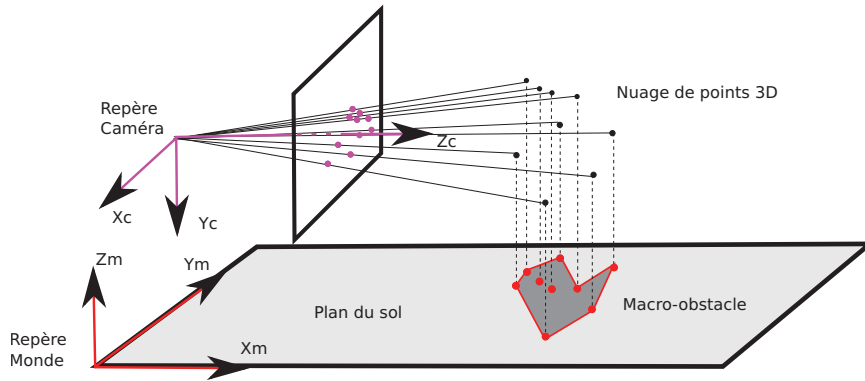


FIGURE IV.39 – Détection des obstacles par approche *Structure from Motion* et création des macro-obstacles 2D

- La hauteur de la caméra varie peu par rapport au sol ;
- Seuls les objets plus hauts qu'un certain seuil sont considérés comme obstacles ;
- L'environnement est majoritairement composé d'objets statiques.

Une analyse du résultat renvoyé par l'algorithme SfM relève que si nous considérons uniquement les points projetés sur le plan du sol, lors du stockage en grille d'occupation, l'obstacle contiendra des parties creuses (voir points rouges de la figure IV.39). Ce phénomène est corrigé par une mise en forme des nuages des points dans ce que nous appelons des macro-obstacles. Nous souhaitons nous attarder maintenant sur les détails de ces algorithmes.

IV.4.4.1 Algorithme de partitionnement de données

Par la suite nous supposons disposer d'un nuage de points 3D issu de l'algorithme de SfM et des projections de ces points sur le plan du sol. Nous utilisons d'abord l'algorithme DBSCAN¹ proposé par [Ester et al., 1996]. Cet algorithme a pour but de partitionner des ensembles de données sur une logique de proximité des éléments constituant suivant un critère de distance. De plus, cet algorithme se veut résistant au bruit et avec une capacité à détecter ce bruit en exigeant un nombre minimal d'éléments (Min_{Elem}) dans le voisinage de chaque point 2D. Le principe de fonctionnement de l'algorithme de partitionnement de données DBSCAN est détaillé dans l'algorithme 4 de l'annexe A.1.

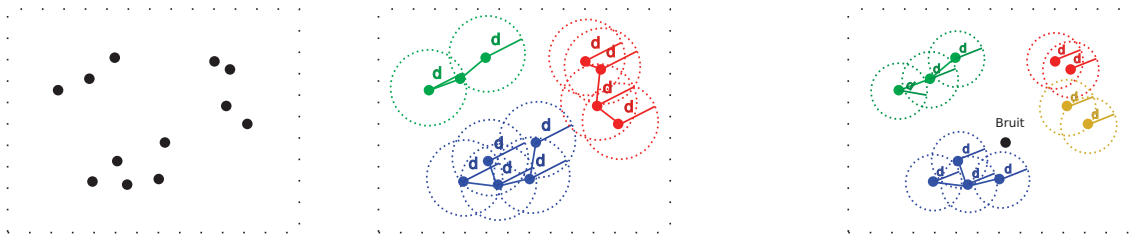


FIGURE IV.40 – Résultat de l'algorithme DBSCAN pour une distance de voisinage d et un nombre minimal d'éléments dans la classe $Min_{Elem} = 1$

1. Density-Based Spatial Clustering of Applications with Noise ;

IV.4.4.2 Algorithme « χ -Shape» pour le décèlement de formes

Une fois avoir réalisé le partitionnement des points dans des obstacles distincts, il faut trouver un formalisme pour exprimer la même information contenue dans chaque classe mais d'une manière compacte. Le choix a été de réaliser l'enveloppe convexe utilisant l'algorithme χ -Shape [Duckham et al., 2008] détaillé dans l'annexe A.2.

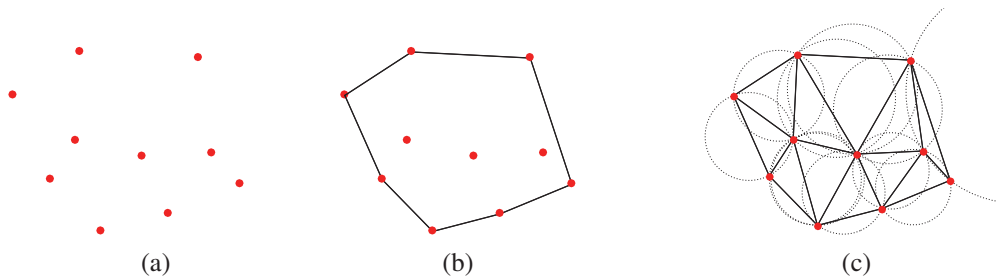


FIGURE IV.41 – Résultat de la triangulation Delaunay

Soit un nuage de points 2D résultat de l'algorithme DBSCAN. Nous appliquons à ce nuage un triangulation de Delaunay [Delaunay, 1934]. Cela représente une décomposition de l'enveloppe convexe dans un ensemble fini des triangles de telle façon qu'aucun point P ne soit à l'intérieur du cercle circonscrit d'un des triangles (figures IV.41(a) à (c)).

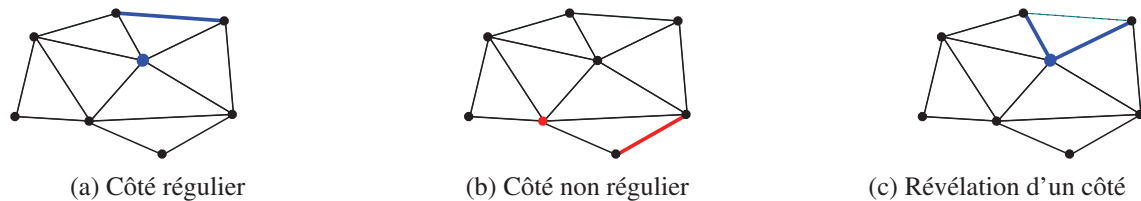
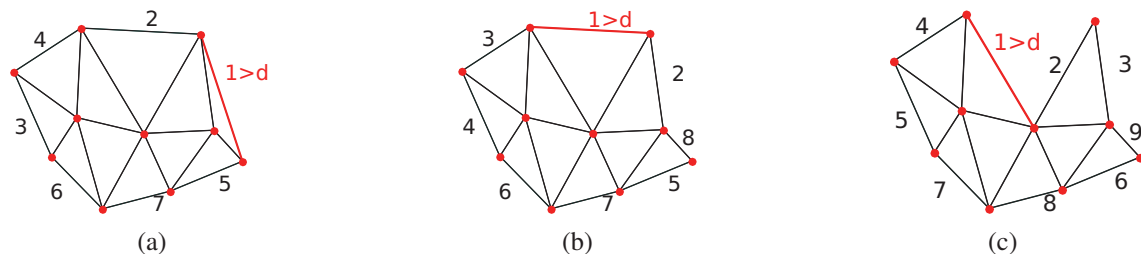
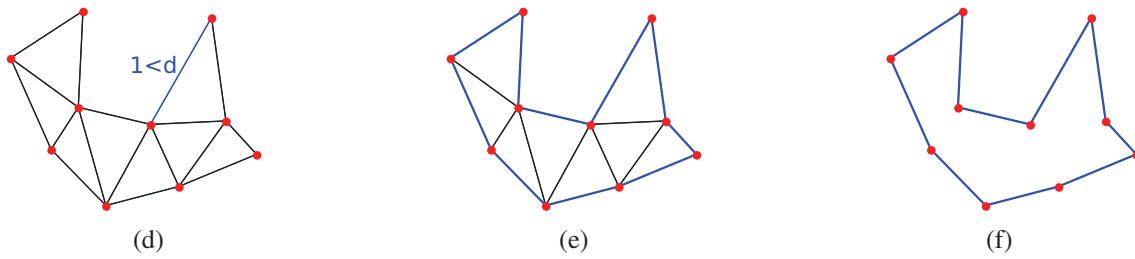


FIGURE IV.42 – Illustration des côtés réguliers, non réguliers ou de la révélation d'un côté

Avant de détailler l'algorithme « χ -Shape» nous introduisons quelques nouvelles notions. On appelle un **côté régulier** (voir figure IV.42(a)) si le sommet opposé à ce côté n'est pas un point de frontière, dans le cas contraire on l'appelle **côté non régulier** (voir figure IV.42(b)). Lorsqu'un côté frontière est supprimé, comme dans la figure IV.42(c), deux autres côtés sont rajoutés et intégreront la nouvelle frontière. Ce processus est appelé **révélation**.



FIGURE IV.43 – Résultats de l'algorithme χ -Shape

Les figures IV.43(a) à (f) illustrent le fonctionnement de l'algorithme « χ -Shape». Par la suite nous donnons exemple d'une situation réelle avec des nuages de 400 à 1 500 points, l'algorithme ressortira des polygones englobants composés de 10 à 30 arêtes ce qui permettra une manipulation plus simple et plus compacte du même volume d'information.

IV.4.5 Résultats de la cartographie par approche *Structure from Motion*

Prenons une carte de points 3D réalisée par l'approche *Structure from Motion* présentée. Parmi tout l'historique des points 3D, nous gardons à l'affichage uniquement les points situés dans un rayon de 10 m autour de la caméra et dont la hauteur est située dans la plage [0,1, 1,8] m. Dans les figures IV.44(a) à (c) nous montrons des nuages de points 2D issus de la projection de points 3D calculés par SfM et projetés sur le plan du sol. Dans les mêmes images, nous présentons également les points regroupés dans des macro-obstacles, sous forme des polygones englobants où les points respectent un certain critère de distance.

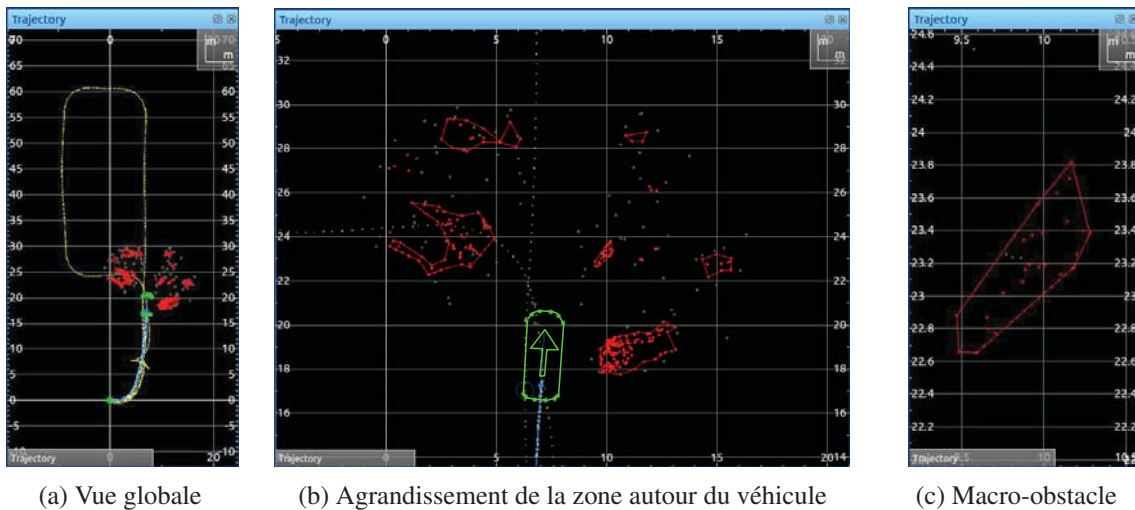
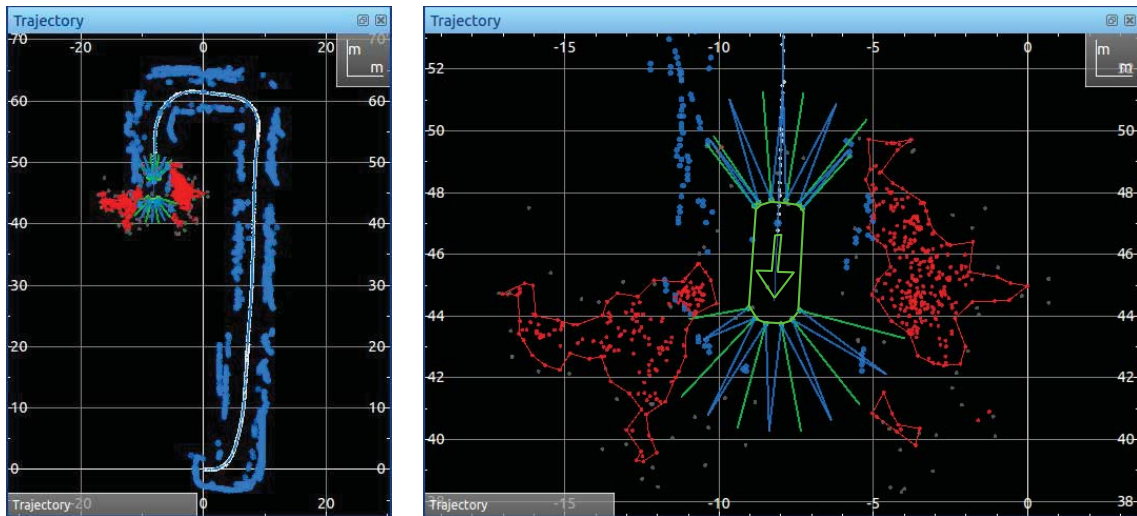


FIGURE IV.44 – Projection sur le plan du sol des points 3D résultés de la détection par SfM (points en rouge) et classification dans des macro-obstacles (les polygones englobants des nuages de points)

Par conséquent, en reprenant la même situation mais cette fois-ci en fusionnant l'information de détection d'obstacles avec système de capteurs ultrason ou par la technique de SfM par vision, nous retrouvons une carte d'obstacles comme celle présentée dans les figures IV.45(a) et (b).

Le dernier exemple que nous présentons ici est la cartographie d'un environnement extérieur (figures IV.46(a) et (b)). La figure IV.46(c) illustre en bleu les obstacles détectés par SCU (G) et



(a) Vue globale des obstacles détectés dans un parking souterrain

(b) Agrandissement de la zone autour du véhicule

FIGURE IV.45 – Superposition des résultats de détection d'obstacles par approche SfM (en rouge) et capteurs ultrason (en bleu). Concernant l'approche SfM, sont retenus uniquement les points détectés dans un rayon de 10 m autour de la caméra et cumulés sur une distance 7 m (utilisation d'un principe de fenêtre glissante)

en couleur rouge les nuages de points des macro-obstacles détectés par approche SfM dans un rayon de 10 m : deux panneaux de circulation (*C* et *F*), trois arbres (*A*, *B* et *D*) et un piéton en mouvement (*E*). Utilisant cette situation nous pouvons tirer deux appréciations qualitatives. Les poteaux de 7 cm en diamètre des panneaux de circulation n'ont pas été détectés ni pas ultrason ni par approche SfM. Par contre le panneau en lui-même a été détecté par approche SfM.

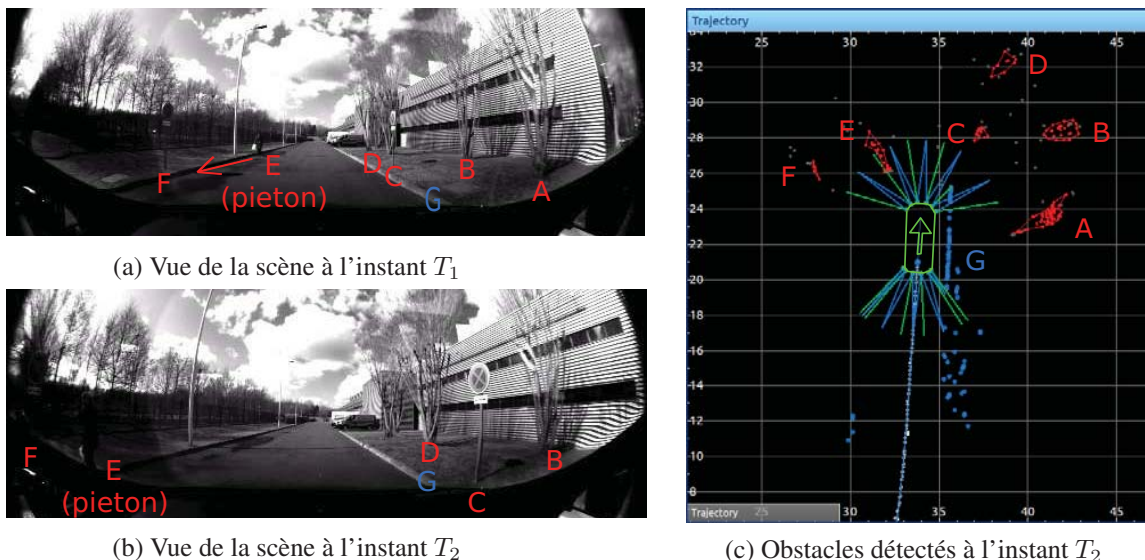
(a) Vue de la scène à l'instant T_1 (b) Vue de la scène à l'instant T_2 (c) Obstacles détectés à l'instant T_2

FIGURE IV.46 – Superposition des résultats de détection d'obstacles par approche SfM (en rouge) et capteurs ultrason (en bleu)

La deuxième constatation est que les bordures de 0,12 m de hauteur n'ont pas été détectées par approche SfM. Pour ne pas détecter le sol comme un obstacle, dans l'approche SfM nous réalisons un filtrage des points 3D qui sont situés à une hauteur de moins 0,15 m. Les erreurs

d'appariement et de calcul des positions des points 3D font qu'en pratique notre algorithme ne garantit pas la détection des obstacles dont la hauteur est de moins de 0,20 m. En revanche, ces bordures sont détectées par les capteurs ultrason, voir les points bleu parallèles à la direction d'avancement du véhicule notée avec la lettre G dans la figure IV.46(c).

Nombre de points	100	200	400	600	800	1 200	1 400	1 600
Temps de calcul	2 ms	3 ms	4 ms	6 ms	9 ms	14 ms	17 ms	19 ms

TABLE IV.3 – Temps de calcul cumulé des algorithmes χ - Shape et DBSCAN

Le tableau IV.3 montre l'évolution du temps de calcul cumulé des algorithmes χ -Shape et DBSCAN. Les résultats sont conformes au traitement en temps réel et sont réalisés en mono-thread sur un processeur i7-4900MQ (8MB Cache) à 4 cœurs et 8 fils d'exécution à fréquence de 2,80 GHz, mémoire vive de 16GB et système d'exploitation Ubuntu 12.04LTS-32 bits. Le résultat est conforme à la complexité théorique $O(n^2)+O(n\log(n)) = O(n^2)$ et, en fonction de la richesse en texture de l'environnement, le résultat peut varier entre 100 et 1 600 points. Bien entendu, pour une application de détection d'obstacle l'appariement de plus de points entre des images successives augmente la probabilité d'une bonne détection de tous les obstacles. Comme nous l'avions montré dans les exemples antérieurs, les obstacles sans texture, comme les murs peints en blanc, ne sont pas détectés, d'où la nécessité de coupler cette technique de détection d'obstacles avec une technique à base de capteurs ultrason.

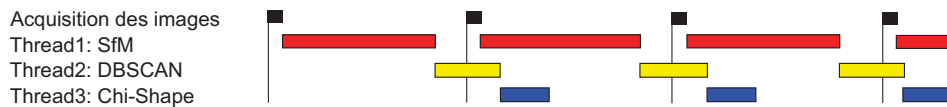


FIGURE IV.47 – Agencement des algorithmes SfM, DBSCAN et χ - Shape

Ces travaux d'analyse quantitative ainsi que la mise en forme des obstacles ont été réalisés en étroite collaboration avec M. Keilatt Adriantavinson, pendant son stage de d'étude d'ingénieur chez Renault du mars à août 2015 [Adriantavinson et al., 2015]. Nous avons réalisé une application qui utilise pour cette partie de détection d'obstacle trois fils d'exécution : un pour l'algorithme SfM, un deuxième pour l'algorithme DBSCAN et un troisième pour l'algorithme χ - Shape. Comme les trois algorithmes s'appliquent l'un à la suite de l'autre, alors il a été possible de les mettre sur 3 fils d'exécution en parallèle. Ainsi il a été possible d'optimiser le temps total d'exécution par du traitement en temps masqué et de respecter la contrainte de traitement de 20 images par seconde que nous nous sommes imposée.

IV.5 Analyse des résultats globaux et bilan

Dans cette section nous présentons des cas de figure pour montrer les avantages et les limites des deux systèmes.

IV.5.1 La complémentarité des deux systèmes de perception

La figure IV.48(a) présente un environnement riche en obstacles mais également riche en texture. En longeant une allée avec des véhicules garés dans des rangées situées des deux côtés du couloir de circulation, le système de voiturier est capable de détecter les obstacles utilisant le

SCU. Il est à remarquer que le système de détection d'obstacle par vision permet, comme attendu, de détecter la position des obstacles plus éloignés. Pour notre application, les résultats montrent qu'une qualité élevée de la détection est assurée pour une limite d'horizon de perception que nous choisissons suite à des mesures expérimentales à 10 m. C'est à dire que même si les points lointains sont suivis en continu, leur utilisation dans la carte d'obstacles se fait uniquement quand ils se trouvent dans un rayon de 10 m, autour de la caméra.

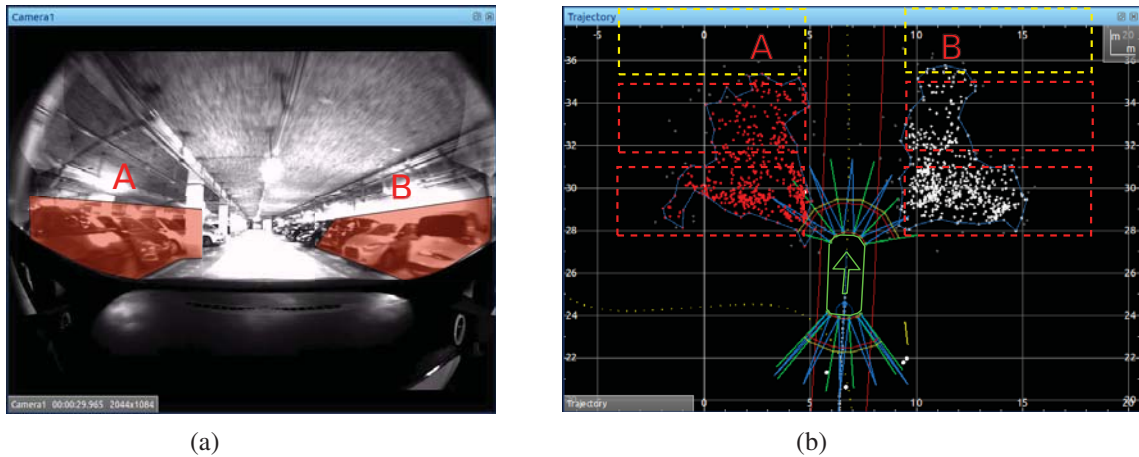


FIGURE IV.48 – Interprétation de la scène : exemple de détection de rangée de voitures où l'algorithme est capable d'apparier plus de 200 points pour chaque rangée. Dans ce cas il est beaucoup plus efficace de manipuler ou d'envoyer l'enveloppe de l'obstacle par le bus CAN que d'envoyer tous les points, sans pour autant perdre de l'information

IV.5.2 Limitations

Dans la figure IV.49(a) nous avons choisi de montrer un environnement intérieur riche en obstacles. Dans la même image, de *A* à *F* sont notés les obstacles qui nous intéressent. La figure IV.49(b) présente le résultat de l'algorithme SfM. Il est facile de constater que 4 obstacles sur 6 ont bien été détectés mais d'autre non. C'est le cas des obstacles *A* et *E*.

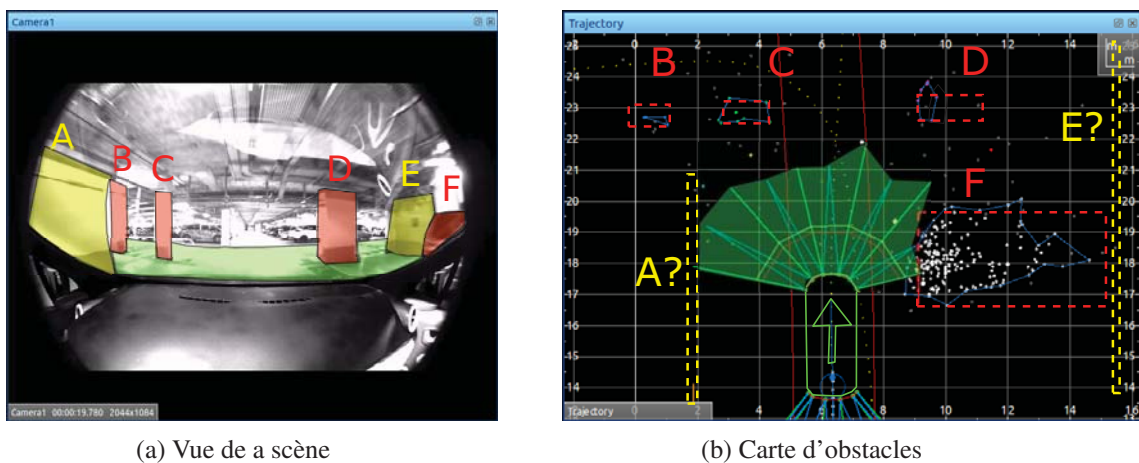


FIGURE IV.49 – Argumentation de la non détection de certains obstacles par la caméra ; les obstacles en rouge (*B*, *C*, *D*, *F*) observés dans la figure (a) se retrouvent bien, par contre, dans la figure (b) les obstacles *A* et *F* n'ont pas été observés car les murs sont peints en blanc et l'algorithme d'extraction de points d'intérêts n'a pas trouvé de points

En nous intéressant de près au problème de non détection, nous constatons que la cause est le manque de texture sur les murs. Par conséquent, l'incapacité d'identification des points d'intérêt ne fait que bloquer la suite de l'algorithme de SfM. Ceci montre que l'utilisation de la caméra comme unique moyen de perception s'avère insuffisante et doit être compensée par un autre capteur.

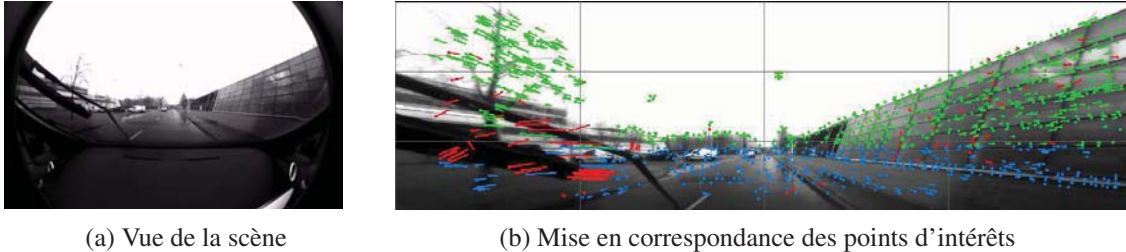


FIGURE IV.50 – Nettoyage du pare-brise par temps de faible pluie ; les points correspondants aux essuie-glaces sont considérés comme des outliers (points et segments rouges dans la figure (b))

Les performances des algorithmes de vision sont très vite limitées dans des situations de mauvais éclairage et d'intempérie. Même si la caméra est installée derrière le pare-brise et que le véhicule est équipé de balais d'essuie-glaces et des feux d'éclairage, à ce jour les algorithmes qui prennent en compte l'obstruction partielle du champ de vue ne sont qu'à leur début. Par exemple les figures IV.50(a) et (b) montrent une scène de nettoyage du pare-brise par temps de pluie. La pluie n'est pas traitée comme un cas différent de fonctionnement. L'algorithme seul élimine les gouttes d'eau et les balais d'essuie-glace car lors de l'appariement des points d'intérêts dans des images successives, les points correspondant à ces objets ne respectent pas un critère de cohérence du mouvement de la caméra par rapport à la scène.

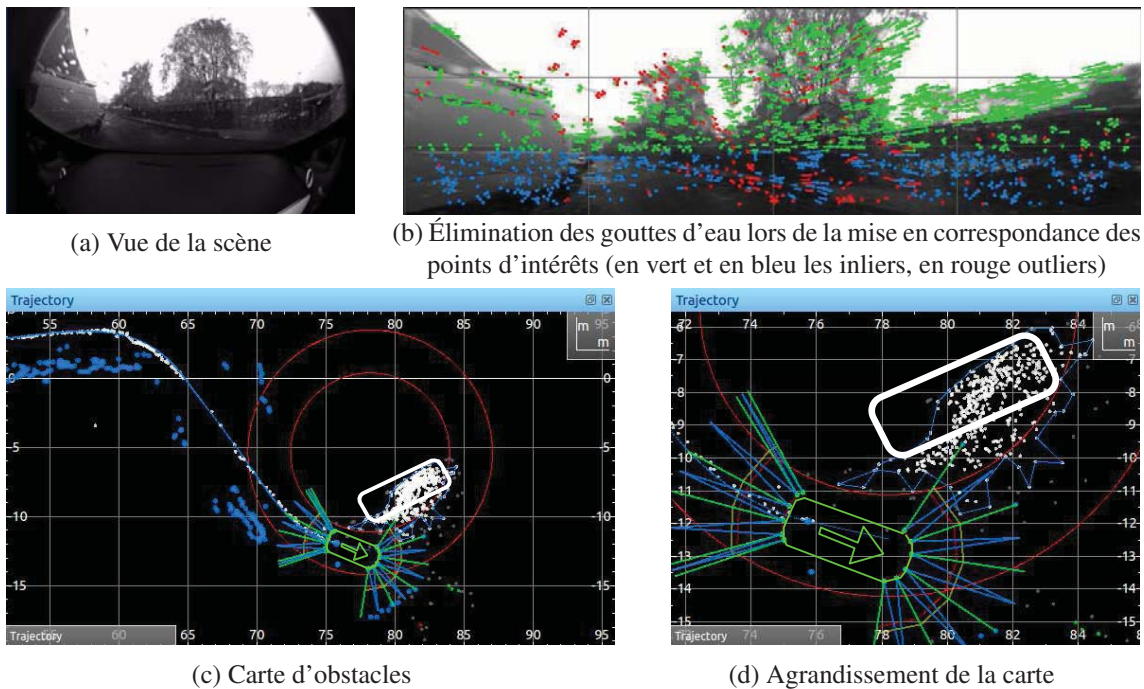


FIGURE IV.51 – Performances de l'algorithme SfM par temps de pluie ; les points blanc correspondent au véhicule garé sur la gauche

Les figure IV.51(a) à (d) détaillent une situation avec détection d'obstacles par temps de pluie. Dans la carte nous pouvons constater que les points blancs détectés par approche SfM correspondant au véhicule garé sur la gauche présentent une disparité importante relative au vrai contour de l'objet détecté.

IV.5.3 Bilan

Ce chapitre a été destiné à l'analyse de la zone traversable effectuée par notre système de voitureur autonome. Nous avons abordé la problématique d'obstacles physiques, mais également des obstacles réglementaires, leur détection et suivi dans le domaine routier et dans une situation de parking.

Nous avons réalisé un bilan des capteurs de distance, nous avons justifié nos choix d'utiliser un système de 12 capteurs ultrason et une caméra monoculaire avec objectif fish-eye tout en détaillant leurs spécificités, avantages et limitations.

Nous avons, ensuite, détaillé notre approche de cartographie par système de capteurs ultrason, les performances obtenues et les raisons qui nous ont conduit à utiliser une approche de *Structure from Motion* pour étendre la zone de perception.

Enfin nous avons conclu que l'utilisation duale caméra - SCU permet d'un côté de pallier le problème de non détection des obstacles non texturés (comme le mur blanc) et en même temps d'avoir un horizon de perception supérieur à celui des capteurs ultrason.

Le chapitre suivant est dédié à la navigation du véhicule et utilise les fonctionnalités de perception, de localisation et de gestion du couloir traversable construites dans les chapitres II à IV. Nous présenterons les problématiques de planification globale et locale avec une illustration de la création de trajectoire pour notre situation de voitureur autonome.

CHAPITRE V

Navigation

Sommaire

IV.1	Problématique	128
IV.2	Gestion de la zone traversable : État de l'art	128
IV.2.1	Grille d'occupation	128
IV.2.2	Algorithmes du type <i>Structure from Motion</i>	131
IV.3	Localisation des obstacles détectés par les capteurs ultrason	133
IV.3.1	Calcul de la position des obstacles par une approche géométrique	133
IV.3.2	Résultats de la cartographie par le système ultrasons	135
IV.3.3	Bilan sur la perception de l'environnement utilisant le système de capteurs ultrason	142
IV.4	Enrichissement de la grille d'occupation avec approche <i>Structure from Motion</i>	142
IV.4.1	Algorithme <i>Structure from Motion</i>	143
IV.4.2	Appariement et suivi des points	145
IV.4.3	Stratégies d'estimation du déplacement relatif de la caméra	147
IV.4.4	Mise en forme de nuages des points 3D sous forme de macro-obstacles 2D	157
IV.4.5	Résultats de la cartographie par approche <i>Structure from Motion</i>	160
IV.5	Analyse des résultats globaux et bilan	162
IV.5.1	La complémentarité des deux systèmes de perception	162
IV.5.2	Limitations	163
IV.5.3	Bilan	165

V.1 Problématique

Nous rappelons que l'objectif est de réaliser un système de voiturier autonome capable de piloter le véhicule entre une zone de dépose du conducteur et la place de parking et inversement entre la place de parking et la zone de reprise du conducteur. Dans les chapitres précédents nous avons détaillé les systèmes de perception, d'analyse de la zone traversable et de localisation du véhicule. Le cinquième chapitre est dédié à la navigation du véhicule.

Pour transformer un véhicule commandable en un véhicule autonome nous avons besoin d'une trajectoire et d'un système de commande/contrôle pour le suivi de trajectoire et pour la régulation de la vitesse (voir figure V.1). Dans ce chapitre nous nous intéressons à la réalisation d'un module de création de la trajectoire locale en lien avec une planification globale tout en tenant compte des facteurs suivants :

- la trajectoire doit guider le véhicule depuis sa position actuelle jusqu'à la place de parking ou à la zone de prise/dépose du conducteur dans le couloir déjà cartographié dans l'étape d'apprentissage ;
- Pour des raisons de sécurité, le véhicule peut circuler uniquement à l'intérieur d'un couloir de navigation cartographié ;
- en cas d'obstacles non cartographiés situés sur la trajectoire de référence, une trajectoire locale devra être créée pour permettre au véhicule de le contourner. Très souvent l'obstacle ne peut pas être observé en totalité depuis un point de vue. Dans ce cas une trajectoire intermédiaire sera créée pour permettre au véhicule de mieux observer l'espace traversable et dans un second temps de créer une trajectoire d'évitement.
- il faudrait tenir compte dans le choix des algorithmes d'évitement d'obstacles que le système de détection par capteurs ultrason est limité à un horizon de perception d'uniquement 4,095 m (avec une bonne précision jusqu'à 3 m) et que la détection d'obstacles par approche *Structure from Motion* est limitée à 8 m.

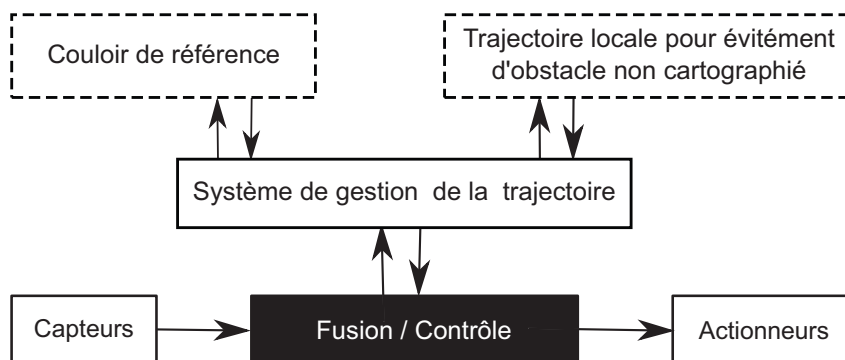


FIGURE V.1 – Schéma de principe du fonctionnement du système de gestion de la trajectoire et de ses interactions

Par la suite nous nous intéressons à l'état de l'art dans le domaine de la navigation avec un focus sur la trajectoire locale d'évitement d'obstacle.

V.2 Navigation : État de l'art

Dans la littérature nous retrouvons souvent deux grandes classes d'approches : planification locale et planification globale. La première approche s'occupe de techniques permettant au véhicule de prendre des décisions pour un court horizon de temps et de distance. Nous retrouvons ici des techniques de création de trajectoire, de contrôle/commande du véhicule pour le suivi de trajectoire ou plus généralement pour le déplacement du véhicule entre deux poses. Nous ajoutons également les techniques de prise de décision pour assurer la sécurité et l'intégrité du véhicule durant le déplacement, comme par exemple le freinage d'urgence. Dans ce premier contexte nous parlons plutôt de guidage que de planification. Par contre le terme planification retrouve tout son sens dans la deuxième catégorie d'approches. La planification globale comprend des techniques qui permettent au véhicule de trouver un chemin admissible en s'appuyant sur des connaissances a priori telles qu'une carte de l'environnement. Par la suite, nous nous intéressons à ces deux classes de techniques de planification.

V.2.1 Planification globale

Comme nous l'avons déjà précisé, la planification globale se propose de créer des trajectoires dans un environnement cartographié. Dans le cas où la carte est complète, les méthodes se proposent de donner une solution directe. Dans le cas où la carte est incomplète, il existe des méthodes qui se proposent de donner une solution à la manière dont l'espace devrait être exploré pour parvenir à trouver un chemin jusqu'à la destination souhaitée.

Dans la section §III.3.1 nous avons détaillé la problématique de la contrainte mécanique du véhicule. Nous avons vu lors de la création de la trajectoire que la contrainte cinématique non-holonome du véhicule est transposée dans la trajectoire à travers une contrainte de courbure. Une trajectoire est dite admissible si, par sa construction, elle respecte les contraintes cinématiques du véhicule. Il existe trois types de trajectoires : *sans manœuvre*, *avec arrêt* ou *avec manœuvre*. Dans le premier cas la vitesse du véhicule ne change pas de signe, dans le deuxième cas le véhicule marque des arrêts et enfin dans le troisième cas le véhicule change de sens de circulation.

Nous nous intéressons d'abord aux approches de création de trajectoires holonomes. Dans ce sens, les premières méthodes utilisées sont celles de champs de potentiels [Khatib, 1986]. En utilisant un principe de champs répulsifs pour les obstacles et des champs attractifs pour les points objectifs, il est possible de changer le problème de planification en un autre problème d'optimisation globale pour trouver un chemin qui relie la position du véhicule d'une destination qui prend la forme d'un minimum global de la fonction.

D'autres approches sont basées sur la discrétisation de l'espace afin de produire des graphes de connectivités. Une fois les graphes construits, le problème de trajectoire est transformé en une recherche d'un chemin dans un arbre. Pour la création du graphe de connectivité nous mentionnons les approches de discrétisation exactes [Schwartz and Sharir, 1983a,b] ou bien les approches de discrétisation approchées (par exemple en utilisant des cellules carrées à taille fixe ou variable, comme dans le cas des quadrees). D'autres approches comme les graphes de visibilité [Huang and Chung, 2004], les graphes de Voronoï (issus du diagramme de Voronoï) [Beeson et al., 2005] permettent de créer directement des graphes contenant des trajectoires admissibles. Nous rappelons également les techniques d'exploration de l'espace cartographié utilisant des algorithmes du type RRT (Rapidly-exploring Random Trees) [LaValle, 1998; LaValle and Kuffner Jr, 2000] ou l'algorithme A* (A Étoilé) [Duchon et al., 2014]

ou l'algorithme Diskstra [Dijkstra, 1959; Kang et al., 2008]. Ces trois algorithmes de référence très utilisés sont à la base de beaucoup de versions plus rapides et plus efficaces en matière de temps de calcul et dans la manière dont l'espace est exploré.

Toutes ces méthodes ont été initialement prévues pour de la robotique holonome et ne prennent pas en compte les contraintes non holonomes dans la construction de la trajectoire. Par contre, elles sont utilisées dans le cadre des algorithmes dits en deux étapes. C'est-à-dire, un ou plusieurs chemins sont construits pour relier la position du véhicule à la destination tout en contournant les obstacles. Ensuite, les trajectoires sont déformées pour intégrer les contraintes cinématiques du véhicule [Sussmann, 2012]. L'inconvénient de cette approche est qu'elle n'assure pas le succès de la transformation holonome dans une autre non holonome. Généralement plusieurs trajectoires sont construites dans la première étape en espérant qu'au moins une pourra être déformée avec succès tout en gardant la contrainte de non-collision entre le véhicule et les obstacles.

Nous ne nous proposons pas de détailler de plus cette classe d'algorithmes de planification globale car dans notre système de voiturier autonome, le véhicule circulera uniquement à l'intérieur d'un couloir de navigation connu depuis une étape d'apprentissage. L'exploration de l'environnement sera faite pour des raisons de sécurité en conduite manuelle. Ce qui nous intéresse davantage ce sont les techniques de planification locale. Dans son couloir de navigation le véhicule peut rencontrer des obstacles non cartographiés qu'il devra éviter lors de son déplacement.

V.2.2 Planification locale

Nous rappelons que la planification locale (ou le guidage) fait référence aux techniques permettant au véhicule de prendre des décisions pour un court horizon de temps et de distance. Lors de la création de trajectoire se pose le problème du chemin optimal. Dans ce sens il a fallu définir un critère d'optimalité. Ainsi, un critère très utilisé est la longueur de la trajectoire entre deux positions dans des situations de présence ou pas d'obstacles.

Pour des trajectoires optimales avec arrêts (selon la classification présentée) L. E. Dubins a démontré qu'en absence d'obstacles elles sont constituées uniquement d'arcs de cercle et de segments et qu'il existe un nombre de six configurations possibles : LRL, RLR, RSL, RSR, LSR et LSL (voir figures V.2(a) à (c)). Les trois lettres R, L, S signifiant en ordre un arc de cercle tournant en sens horaire (R de Right), un arc de cercle tournant en sens trigonométrique (L de Left) et enfin un segment de droite [Dubins, 1957; Bui et al., 1994]. Les travaux de L. E. Dubins ont été généralisés et, en ce qui concerne les trajectoires optimales avec manœuvre (la vitesse change de signe), J. Reeds et L. Shepp ont démontré qu'elles sont constituées toujours des arcs de cercle et des segments lignes droites auxquels sont ajoutés les symboles + et - qui indiquent le signe de la vitesse (L^+ , L^- , R^+ , R^- , S^+ , S^-). Nous ne donnons pas la totalité des configurations ici, mais les auteurs ont démontré qu'il n'existe que 48 configurations possibles [Reeds and Shepp, 1990; Soueres and Laumond, 1996].

Le grand inconvénient de l'utilisation de ce type de trajectoire avec arrêts vient du fait qu'elles sont composées des segments d'arc de cercle et de segments de droite reliés de manière tangentielle. Cela signifie que dans les points de connexion il y a des discontinuités de courbure qui oblige le véhicule à s'arrêter pour braquer les roues directrices. Il existe cependant des situations où cette approche n'est pas très contraignante. Par exemple dans le cas du parking latéral, le nombre le plus réduit des manœuvres est atteint si le braquage des roues se fait à l'arrêt (voir par exemple

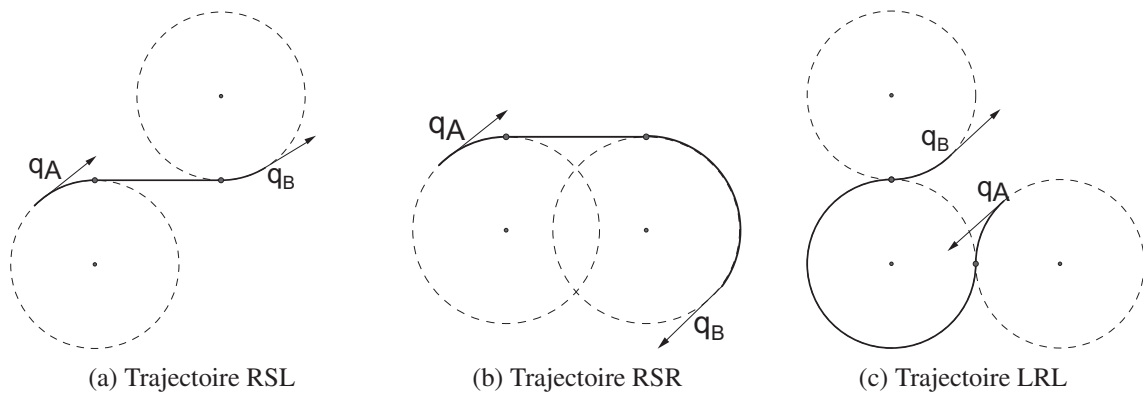


FIGURE V.2 – Exemple de trajectoires optimales avec arrêts [Dubins, 1957]. Par les lettres R, L, S (Right, Left, Segment) sont notés les éléments qui composent la trajectoire

les travaux de [Vorobieva, 2014] sur les manœuvres de parking latéral de type créneau utilisant des arcs de cercles).

Une autre classe de trajectoire existe dans la littérature et elle est appelée trajectoire à courbure continue. À nouveau le problème de trajectoire optimale a été posé et plusieurs critères ont été proposés. Par exemple, à vitesse constante de minimiser l'accélération latérale ou les variations d'accélération latérale. Cela revient à minimiser l'intégrale au carré de la courbure respectivement de la dérivée de la courbure le long de la trajectoire [Kanayama and Hartman, 1990]. Dans ce cas, les courbes obtenues sont des clothoïdes respectivement de spirales cubiques. Un autre critère est de minimiser les variations de l'accélération latérale pour une vitesse longitudinale qui peut varier. Cela se traduit le plus souvent dans une minimisation de l'intégrale au carré de la dérivée de l'accélération. Dans ce cas, les courbes résultantes ont des coordonnées cartésiennes de polynômes d'ordre cinq [Takahashi et al., 1989].

Dans les travaux de [Scheuer, 1998] une classification des trajectoires à courbure continue en deux classes nous est proposée. La première correspond à des trajectoires dont la courbure est un polynôme de l'abscisse curviligne (des cas particuliers sont les clothoïdes). Le grand avantage de cette première catégorie consiste dans une formulation simple par rapport aux paramètres de configuration. Dans la deuxième classe sont regroupées les courbes à coordonnées cartésiennes polynomiales ou bien des courbes à coordonnées polaires polynomiales ou enfin des B-splines. Cette deuxième catégorie présente l'inconvénient majeur de nécessiter des calculs coûteux dus à l'utilisation directe des contraintes de courbure.

Ce type de trajectoire a commencé à être utilisé massivement dans les années '70 pour la construction des autoroutes [Jaforte, 1976] car elles permettent de limiter l'accélération latérale lors du changement de direction. Cela se traduit dans l'ajout d'une borne sur la dérivée de la courbure ce qui correspond à une limitation de la vitesse de braquage des roues directrices.

Il a été démontré par [Boissonnat et al., 1994] que parmi les courbes à dérivée de courbure bornée les clothoïdes sont les plus optimales dans le sens de minimisation de l'accélération latérale pour une vitesse donnée. Ils ont également montré que, dans ce nouveau contexte, les trajectoires sont composées soit d'un nombre fini de morceaux de clothoïdes soit d'un nombre infini de morceaux de clothoïdes, du moment où elles contiennent au moins un segment de droite. [Kostov et al., 1995; Degtiariova-Kostova and Kostov, 1998] ont démontré que les trajectoires proposées par J. Reeds et L. Shepp pour relier deux positions du véhicule sans réaliser des manœuvres peuvent être remplacées par une classe de trajectoires sous optimales en matière de longueur. Ces trajectoires à courbure continue

dépassent la longueur des trajectoires Reeds-Shepp d'au maximum une distance qui est fonction de la valeur-limite imposée à la dérivée de la courbure.

Enfin, nous nous intéressons à la méthode des tentacules ou du faisceau de trajectoires. Pour la nécessité de gagner en réactivité, plusieurs trajectoires admissibles sont calculées hors ligne et la trajectoire la plus adaptée à la situation sera choisie ultérieurement en temps réel. En fonction de la méthode de construction de chaque trajectoire, nous trouvons dans la littérature des tentacules en forme d'arcs de cercle [Von Hundelshausen et al., 2008], des tentacules en forme de clothoïdes [Chebly et al., 2014; Alia et al., 2015; Mouhagir et al., 2016] ou bien des tentacules parallèles à la route ou à une trajectoire de référence [Delmas, 2011; Chu et al., 2012]. Nous reviendrons sur ces approches dans les sections suivantes.

Pour parvenir à mettre en place la stratégie d'évitement d'obstacle que nous avons envisagée et que nous détaillerons dans les sections suivantes, nous nous sommes intéressés également aux études réalisées par H. Vorobieva dans le cadre sa thèse [Vorobieva, 2014]. Ses travaux portent sur la création de trajectoires clothoïdales admissibles adaptées aux situations de parking latéral du type créneau. Mais avant d'entrer dans les détails techniques de la création de trajectoire, nous donnons quelques propriétés générales de la clothoïde.

V.2.2.1 Propriétés générales de la clothoïde

Nous nous intéressons maintenant à la clothoïde, une courbe dont la courbure κ varie linéairement avec son abscisse curviligne s . En reprenant les notations consacrées [Scheuer, 1998; Fraichard and Scheuer, 2004; Latombe, 2012] la courbure s'exprime ainsi :

$$\kappa(s) = \sigma s + \kappa(0) \quad (\text{V.1})$$

avec σ une constante caractérisant la clothoïde et $\kappa(0)$ la courbure de la trajectoire à l'origine.

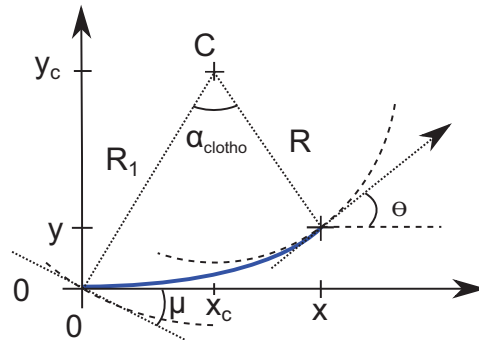


FIGURE V.3 – Trajectoire clothoïdale pour une configuration initiale $q_i = (0, 0, 0, 0)$

Considérons la configuration $q = (x, y, \theta, \kappa)$ définie comme un quadruplet et composée de la position et l'orientation du centre de l'essieu arrière du véhicule dans un repère monde \mathfrak{R}_m ainsi que la courbure instantanée κ (l'inverse du rayon de braquage instantané). Dans la section §III.3.1 nous avons présenté le modèle d'Ackermann et la relation entre la courbure instantanée κ et l'angle de braquage β de la roue moyenne avant pour un modèle bicyclette.

Dans ses travaux, H. Vorobieva considère une configuration initiale du véhicule $q_0 : (x, y, \theta, \kappa) = (0, 0, 0, 0)$. Cela signifie que le véhicule se trouve dans l'origine du repère et que les deux roues avant directrices sont droites. Pour la simplicité des équations nous prenons d'abord la même hypothèse sur laquelle nous reviendrons dans les sections suivantes. Par conséquent, si le véhicule avance avec une vitesse longitudinale et une vitesse de braquage vers la

gauche constantes, alors il est en train de décrire une clothoïde (voir figure V.3). Si s est l'abscisse curviligne, alors toute configuration q_s du véhicule à une distance s de la configuration initiale q_0 est décrite par l'équation (V.2) [Vorobieva, 2014].

$$q_s = (x, y, \theta, \kappa) = \left(A\sqrt{\pi} C_f \left(\frac{s}{\pi A} \right), A\sqrt{\pi} S_f \left(\frac{s}{\pi A} \right), s\kappa, \kappa \right) \quad (\text{V.2})$$

avec :

$$\kappa = \sigma s \quad (\text{V.3})$$

$$A^2 = \frac{2s}{\kappa} \quad (\text{V.4})$$

$$C_f(s) = \int_0^s \cos \frac{\pi u^2}{2} du \quad (\text{V.5})$$

$$S_f(s) = \int_0^s \sin \frac{\pi u^2}{2} du \quad (\text{V.6})$$

où $1/\kappa$ est le rayon de braquage instantané du centre de l'essieu arrière, A est le paramètre de la clothoïde, $C_f(s)$ et $S_f(s)$ sont les intégrales de Fresnel. Comme les équations précédentes sont difficiles à manipuler, nous utilisons la forme discrète approximée sous forme de série entière des équations de Fresnel.

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} \quad \text{et} \quad \cos x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} \quad (\text{V.7})$$

Sachant que les fonctions trigonométriques $\sin x$ et $\cos x$ s'écrivent sous forme de série, conformément aux équations (V.7), alors les coordonnées (x, y) de chaque point de la clothoïde sont données par les expressions (V.8).

$$\begin{cases} x = \sum_{n=0}^{\infty} \frac{(-1)^n s^{4n+1}}{(2n)!(4n+1)(\sqrt{2}A)^{4n}} \\ y = \sum_{n=0}^{\infty} \frac{(-1)^n s^{4n+3}}{(2n+1)!(4n+3)(\sqrt{2}A)^{4n+2}} \end{cases} \quad (\text{V.8})$$

Pour un calcul en temps réel, une approximation des coordonnées peut être faite en utilisant un n relativement petit ($n \leq 8$ comme le montre le tableau V.1). A remarquer que dans le calcul d'une position nous n'avons pas besoin de la position antérieure, mais uniquement de la position initiale. Ainsi, connaissant uniquement le paramètre A de la clothoïde, ces équations nous permettent de calculer directement la position que le véhicule occupera après avoir parcouru une distance s .

Lim.Sup. n :	1	2	3	4	5	6	7	8
Erreur [m] :	1 666	254, 54	12, 89	0, 27	0, 029	$1, 7 \cdot 10^{-5}$	$5, 92 \cdot 10^{-8}$	$1, 29 \cdot 10^{-10}$

TABLE V.1 – Comparaison des coordonnées une clothoïde idéale avec celles approximées par une série (cf. equation (V.8) pour des clothoïdes de paramètre $A = \sqrt{200}$ (voir figure V.4). Variation de l'erreur de calcul des coordonnées de la clothoïde en fonction du choix de la limite supérieure de la série ($n = \{0, 1, \dots, \text{limSup}\}$)

Dans la littérature nous retrouvons des séquences de trajectoires sous optimales à allure symétrique. A titre d'exemple [Vorobieva, 2014] appelle *séquences VCC* (séquences de virages à courbure continue) des trajectoires dont la courbure respecte un des deux profils présentés dans la figure V.5(a). Si lors du virage la courbure maximale κ_{max} est atteinte, alors le virage sera composé d'un arc de cercle encadré par deux clothoïdes symétriques comme le montre la figure V.5(b). Enfin, si les roues ne braquent pas au maximum, la trajectoire sera composée de deux clothoïdes symétriques comme dans la figure V.5(c).

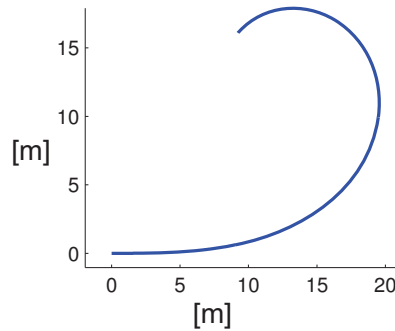


FIGURE V.4 – Clothoïde de courbure initiale nulle, caractérisée par un paramètre $A = \sqrt{200}$ et de longueur totale de 40 m utilisée comme base de comparaison dans le tableau V.1

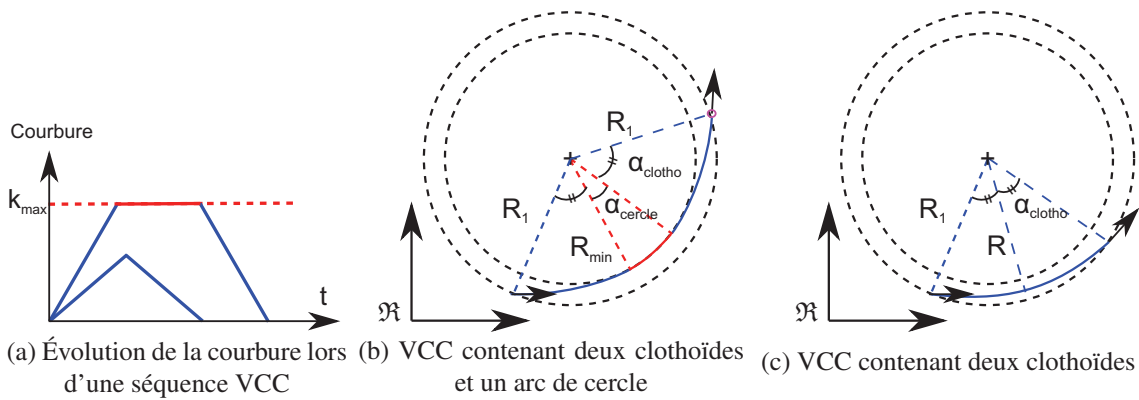


FIGURE V.5 – Séquences de virages à courbure continue (VCC) pour deux profils de courbures : avec et sans atteinte d'une courbure maximale κ_{max}

V.2.2.2 Chemins bi-élémentaires avec courbures initiale et finale nulles

Il est possible de combiner deux courbes continues dans ce qu'on retrouve dans la littérature sur le nom de *chemins bi-élémentaires* [Scheuer, 1998; Vorobieva, 2014]. A. Scheuer a démontré qu'il existe un virage à courbure continue entre deux configurations $q_A : (x_A, y_A, \theta_A, \kappa_A = 0)$ et à $q_B : (x_B, y_B, \theta_B, \kappa_B = 0)$ si et seulement si ces poses sont symétriques par rapport à la droite les reliant. Les figures V.6(a) et (b) illustrent des poses symétriques et également des poses non symétriques. Deux poses sont symétriques si elles vérifient la contrainte géométrique exprimée par l'équation (V.9).

$$(x_B - x_A) \sin\left(\frac{\theta_B + \theta_A}{2}\right) = (y_B - y_A) \cos\left(\frac{\theta_B + \theta_A}{2}\right) \quad (V.9)$$

En pratique, deux poses sont très rarement symétriques mais, il est possible de trouver des poses intermédiaires symétriques à la fois à la pose initiale q_A et à la fois à la pose finale q_B . Il est prouvé dans [Scheuer, 1998] que de telles poses existent et que l'ensemble de ces poses est soit un cercle si $\kappa \neq 0$, soit une droite si $\kappa = 0$. Ces deux situations sont illustrées dans les figures V.7(a) et (b) et les poses intermédiaires q_{Int} sont définies par les équations (V.10) et (V.11)

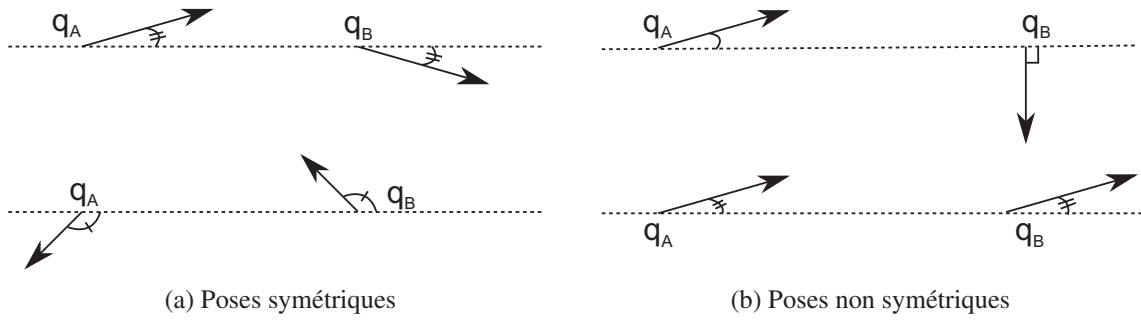


FIGURE V.6 – Exemples de poses symétriques et de poses non symétriques [Vorobieva, 2014]

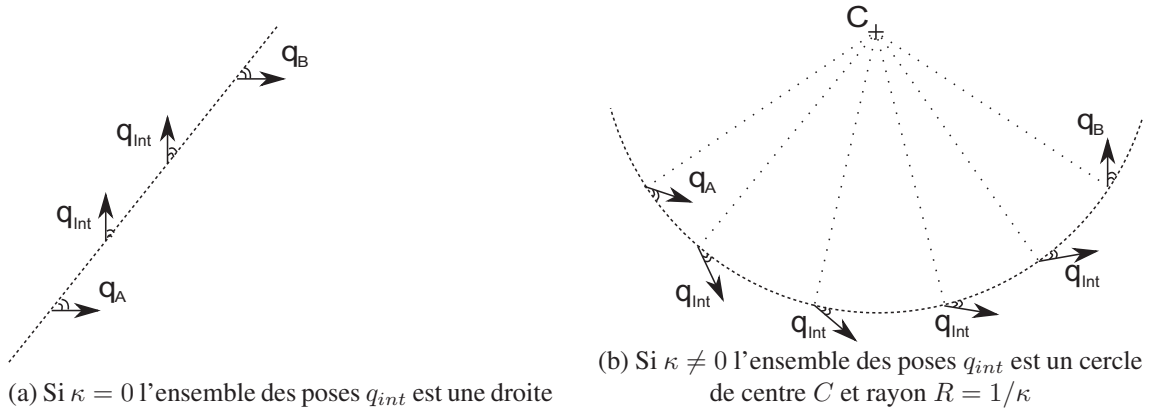


FIGURE V.7 – Poses à la fois symétriques à q_A et à q_B [Vorobieva, 2014]

où s représente la distance curviligne sur le cercle ou sur la droite de l'ensemble des poses.

$$\text{Si } \kappa \neq 0 : q_{Int}(\kappa, s) = \begin{cases} x_{Int}(\kappa, s) = \frac{\sin(\kappa s)}{\kappa} \\ y_{Int}(\kappa, s) = \frac{\cos \beta - \cos(\kappa s)}{\kappa} \\ \psi_{Int}(\kappa, s) = \kappa s - \frac{\alpha + \alpha'}{2} \end{cases} \quad (\text{V.10})$$

$$\text{Si } \kappa = 0 : q_{Int}(0, s) = \begin{cases} x_{Int}(0, s) = s \\ y_{Int}(0, s) = 0 \\ \psi_{Int}(0, s) = -\alpha \end{cases} \quad (\text{V.11})$$

où α et α' sont les angles formés entre les poses q_A respectivement q_B et le segment $[AB]$. Ainsi des trajectoires peuvent être construites en utilisant deux segments de clothoïdes pour relier une configuration q_A à une autre q_B , tel que présenté dans la figure V.8.

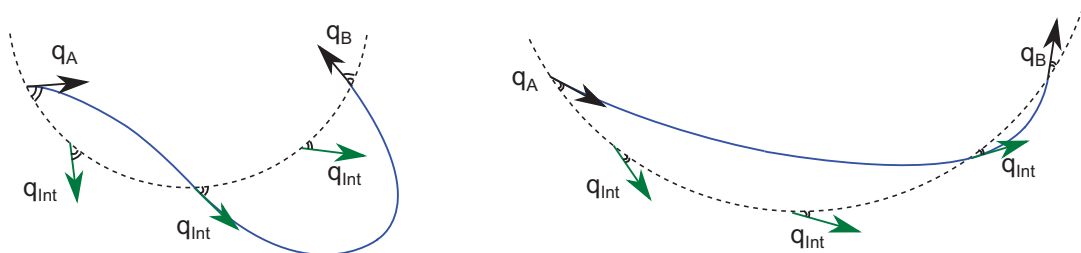


FIGURE V.8 – Exemple de chemins bi-élémentaires [Vorobieva, 2014]

La figure V.9 illustre le fait que entre deux poses plusieurs chemins bi-élémentaires. Par conséquent il faudrait construire une stratégie pour le choix de la meilleure trajectoire suivant un critère tout en tenant compte que la trajectoire doit pouvoir être réalisable par le véhicule.

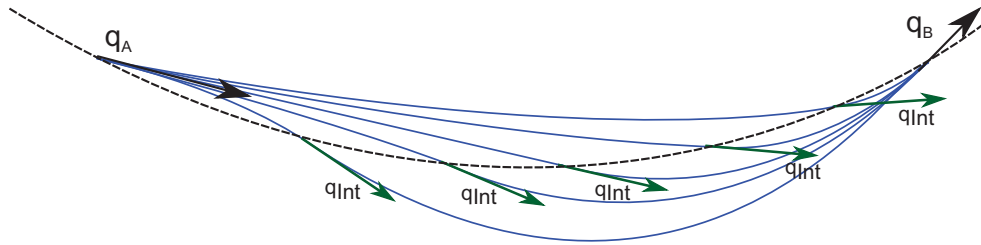


FIGURE V.9 – Variation des courbures maximales des séquences des virages à courbure continue en fonction du point intermédiaire [Vorobieva, 2014]

La particularité de la trajectoire bi-élémentaire dans la forme exposée vient du fait que configuration initiale et également dans la configuration finale les courbures sont nulles. Cela se traduit par des roues directrices situées dans une position de roulage en ligne droite.

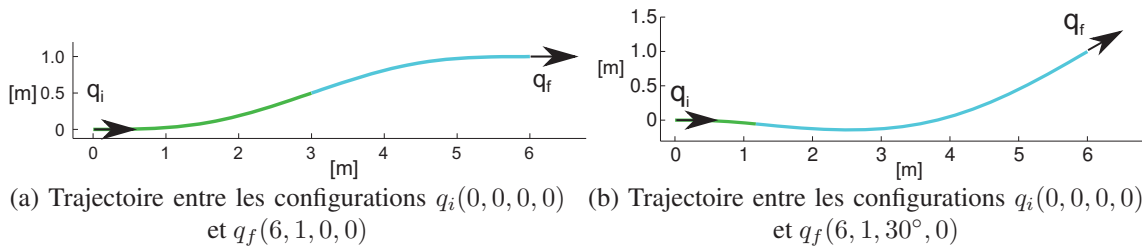


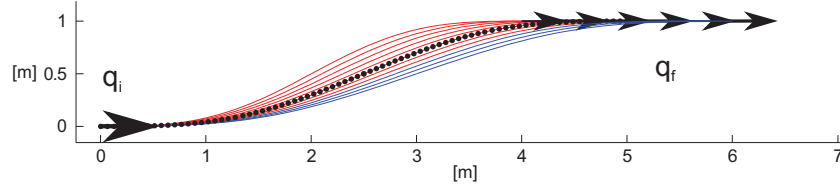
FIGURE V.10 – Trajectoires admissibles à courbure continue construites utilisant la méthode proposée par [Scheuer, 1998; Vorobieva, 2014] avec la contrainte de courbures instantanées nulles dans les configurations initiale et finale

Dans la figure V.10(a) nous présentons un exemple de trajectoire construite pour une configuration initiale $q_i(0, 0, 0, 0)$ et une configuration à l'arrivée $q_f(6, 1, 0, 0)$. Les deux couleurs (vert et cyan) indiquent les deux portions de clothoïde dans la composition de la trajectoire et les flèches le sens d'avancement du véhicule. La figure V.10(b) illustre une trajectoire pour une même configuration initiale mais pour une configuration finale $q_f(6, 1, 30^\circ, 0)$.

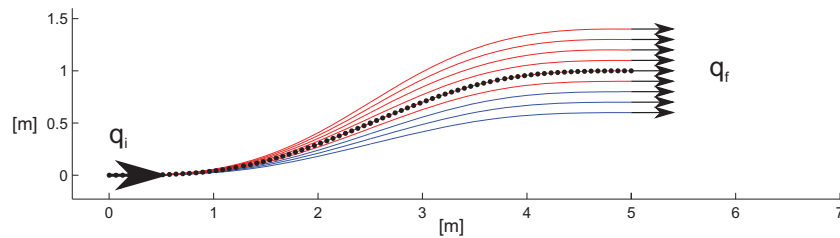
V.2.2.3 Faisceau de trajectoires admissibles (ou non) avec courbures initiale et finale nulles

Outre le fait que l'existence d'une courbure nulle en début de la trajectoire peut être problématique, il existe des cas où cette méthode ne fournit pas de trajectoire admissible entre deux configurations. Dans les figures V.11(a) à (c) en pointillés noirs sont représentées les trajectoires initialement souhaitées qui ne sont pas admissibles. Dans les faisceaux de trajectoires créés nous représentons en rouge et en bleu les trajectoires non admissibles respectivement admissibles du point de vue du respect de la contrainte de la courbure maximale. Il est observable que, même si la trajectoire demandée n'est pas admissible, il existe une infinité de configurations dans le voisinage de la position finale $q_f(x + \epsilon_x, y + \epsilon_y, \theta + \epsilon_\theta, \kappa = 0)$. Par exemple, dans la figure V.11(d) nous combinons une variation latérale et longitudinale de la position finale. Par conséquent, trouver une trajectoire admissible revient à utiliser un critère de choix d'une trajectoire parmi un faisceau de

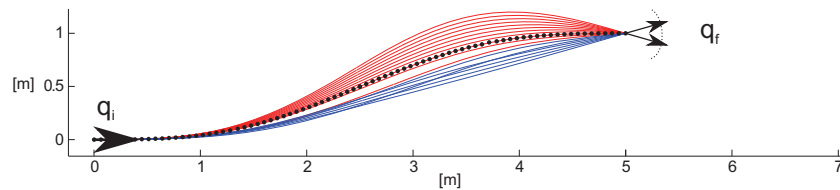
trajectoires. Un autre critère généralement utilisé est la fluidité de la manœuvre qui se traduit dans la plus petite accélération latérale.



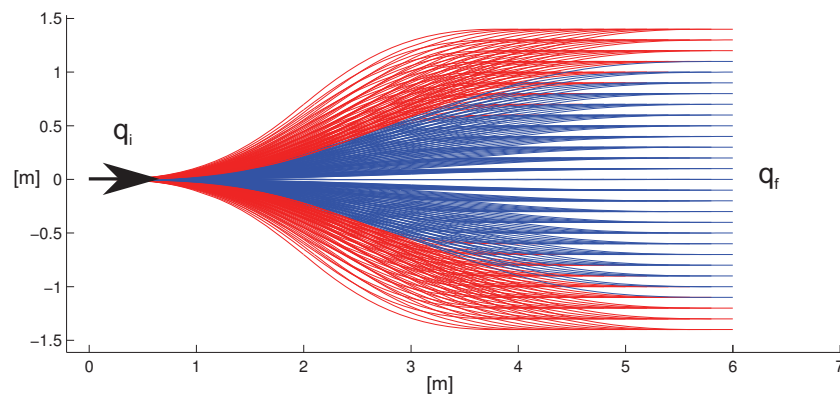
(a) Variation longitudinale de la configuration d'arrivée $q_f(5 \pm \epsilon_x, 1, 0, 0)$, pour $\epsilon_x = 0,2 k_x$; $k_x \in \mathbb{N}$



(b) Variation latérale de la configuration d'arrivée $q_f(5, 1 \pm \epsilon_y, 0, 0)$, pour $\epsilon_y = 0,1 k_y$; $k_y \in \mathbb{N}$



(c) Variation angulaire de la configuration d'arrivée $q_f(5, 1, 0 \pm \epsilon_\theta, 0)$, pour $\epsilon_\theta = \frac{k_\theta \pi}{180}$; $k_\theta \in \mathbb{N}$



(d) Variation latérale et longitudinale de la configuration d'arrivée

FIGURE V.11 – Faisceaux de trajectoires reliant différentes configurations. Dans les figures (a) à (c) en pointés noirs sont représentées les trajectoires initialement souhaitées qui ne sont pas admissibles. En rouge et bleu en sont les trajectoires non admissibles respectivement admissibles du point de vue du respect de la contrainte de courbure maximale

Nous reviendrons sur le sujet avec la solution proposée dans la section §V.4 concernant la problématique de création de la trajectoire à des configurations initiale et finale avec courbure $\kappa \neq 0$. Par la suite nous présentons la stratégie d'évitement d'obstacles que nous employons.

V.3 Stratégie d'évitement d'obstacles

Nous nous intéressons d'abord à la stratégie d'évitement d'obstacle. L'approche de création de trajectoire bi-élémentaire que nous avons exposée dans l'état de l'art présente l'inconvénient de créer des trajectoires entre deux configurations q_A et q_B sous la contrainte que les courbures initiales et finales soient nulles. Si nous voulons utiliser cette approche, il est nécessaire que les roues directrices du véhicule soient droites. Si dans le cas de création de trajectoire de parking latéral cela ne posait pas de problème car en position de départ la contrainte était assurée, dans notre cas, il est fortement probable que, lors de l'apparition de l'obstacle dans le champ de vue du véhicule, les roues soient tournées.

Compte tenu de l'application, nous nous focalisons par la suite au contournement d'un seul obstacle. A cet instant nous supposons disposer d'un algorithme qui fournit une trajectoire admissible entre deux configurations de courbures non nulles (les détails de l'algorithme seront présentés dans la section §V.4.2).

V.3.1 Trajectoire locale d'évitement d'obstacle fixe

Prenons la situation où le véhicule conduit en mode autonome dans un couloir de navigation connu depuis une étape d'apprentissage. Dans ce cas de figure la trajectoire est adaptée à la non-collision entre véhicule et obstacles. Si un obstacle non cartographié bloque l'avancement du véhicule sur la trajectoire de référence alors le véhicule doit construire une autre trajectoire locale adaptée à la situation. Pour la simplicité nous prenons le rectangle englobant de l'obstacle (la projection sur le plan du sol de la boîte englobante de l'obstacle).

Lors de l'évitement d'un obstacle par un véhicule qui suit une trajectoire donnée, il est possible d'utiliser des méthodes basées sur un raisonnement en termes d'écart latéral par rapport à la trajectoire. Notamment le système d'aide à la conduite du type centrage automatique dans la voie (LCA¹) suppose de positionner le véhicule à un écart latéral nul par rapport au centre de la voie. Toujours dans le même cadre d'aide à la conduite, il existe des véhicules, comme la Tesla, qui est capable de quitter le centre de voie pour se rabattre à gauche ou à droite pour laisser passer une moto remontant la file de véhicule ou bien pour éviter un accident par manœuvre d'esquive latérale.

Nous avons décidé de ne pas utiliser cette approche de contrôle/commande en écart latéral car nous avons souhaité maîtriser le profil de la trajectoire empruntée par le véhicule. Par contre, nous avons conçu un module capable de créer une suite de trajectoires admissibles et de suivre ces trajectoires le plus fidèlement possible. Comme généralement le véhicule ne perçoit pas l'intégralité de l'obstacle depuis un point de vue, la création de la trajectoire d'évitement se fera en plusieurs étapes. Par la suite nous présentons une situation typique. Lors de la perception de l'obstacle, le véhicule se situe dans la configuration q_{A1} (voir figure V.12) depuis laquelle il est capable d'observer la largeur de l'obstacle, respectivement le segment rouge continu $[X_1 X_3]$. Si nous avons disposé également de la longueur de l'obstacle $[X_3 X_4]$, alors nous aurions pu utiliser un algorithme capable de nous fournir directement une trajectoire d'évitement entre la configuration q_{A1} et q_{C3} .

1. Lane Centering Assist;

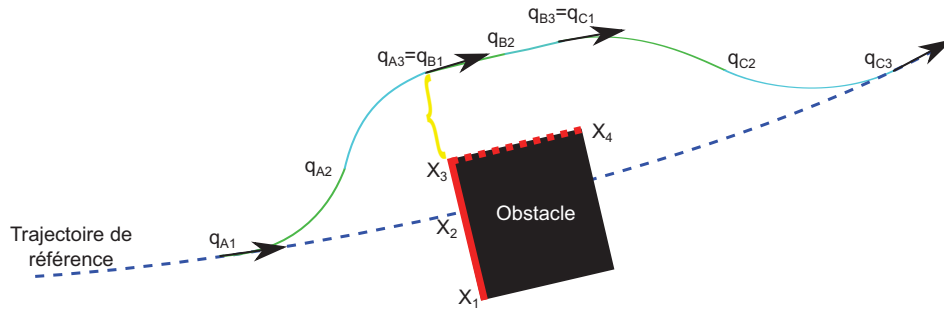


FIGURE V.12 – Principe et séquences de construction de la trajectoire d'évitement de l'obstacle situé sur le trajet du véhicule

Mais en pratique la longueur de l'obstacle $[X_3X_4]$ est mesurée par le système de perception uniquement lorsque le véhicule s'approche de l'obstacle et commence à le contourner. C'est pour cela que la trajectoire sera découpée en plusieurs sous trajectoires :

- $\{q_{A1}, q_{A2}, q_{A3}\}$ - la première sous trajectoire est créée pour conduire le véhicule dans une situation depuis laquelle il sera capable d'observer la longueur de l'obstacle ;
- $\{q_{B1}, q_{B2}, q_{B3}\}$ - le deuxième segment permettra au véhicule de longer l'obstacle ;
- $\{q_{C1}, q_{C2}, q_{C3}\}$ - la dernière partie de la trajectoire d'évitement permettra au véhicule de rejoindre la trajectoire de référence.

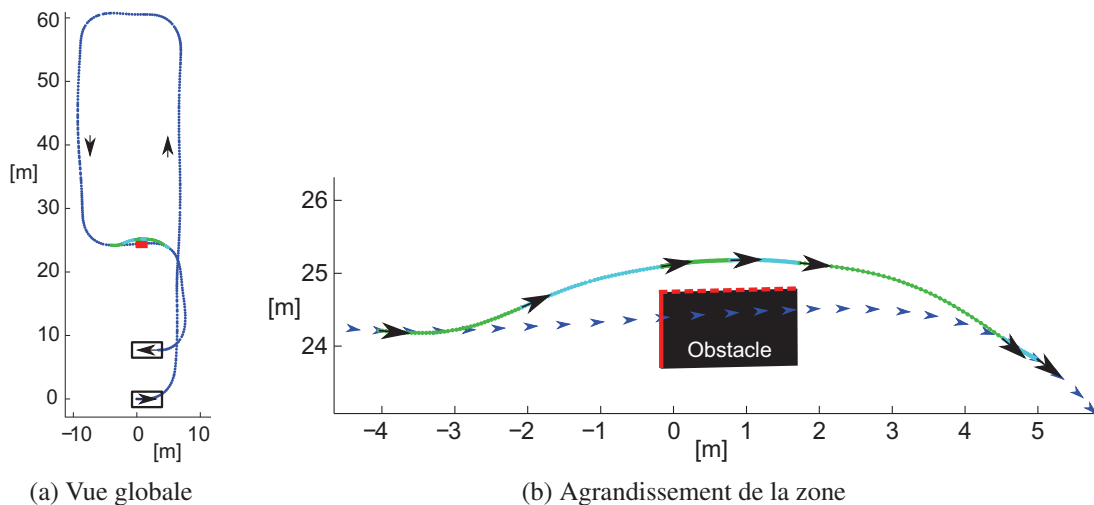


FIGURE V.13 – Trajectoires locales d'évitement d'obstacles - en bleu la trajectoire initiale ; en vert et cyan les séquences de clothoïdes qui forment la trajectoire d'évitement d'obstacle

En utilisant une approche de création de trajectoire bi-élémentaire généralisée entre configurations de courbures non nulles [§V.4.2], nous réalisons les trois morceaux de trajectoires admissibles d'évitement de l'obstacle entre les configurations q_{A1} et q_{C3} . La contrainte d'admissibilité se traduit dans un rayon de courbure tout au long de la trajectoire plus grand que le rayon minimum réalisable par le véhicule Renault Zoe. Pour anticiper, les figures V.13(a) et (b) présentent le résultat obtenu pour une situation donnée. Dans ce premier exemple nous n'avons pas introduit le gabarit du véhicule et les configurations intermédiaires q_{B1} à q_{B3} ont été calculés uniquement en tenant compte d'une distance de sécurité par rapport à l'obstacle (en jaune dans la figure V.12).

V.3.2 Principe de l'approche d'évitement d'obstacles dans un milieu non-cartographié prenant en compte le gabarit du véhicule

Par rapport à l'exemple présenté dans les figures V.13(a) et (b), nous nous intéressons à la création de trajectoire admissible tout en tenant compte du gabarit du véhicule. Par simplicité, dans un espace 2D nous considérons d'abord que le véhicule est représenté par un rectangle englobant. Dans une deuxième phase nous montrerons qu'il est possible de réaliser une estimation de la surface du véhicule au sol par l'utilisation de 4 ou 5 cercles (voir figures V.18(a) et V.18(b)).

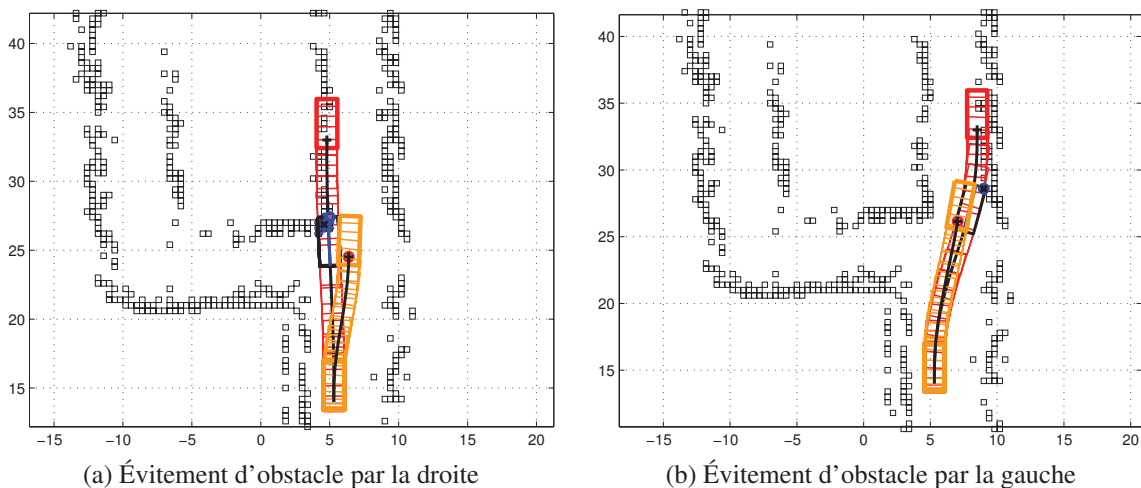


FIGURE V.14 – Calcul de trajectoire d'évitement dans un milieu non cartographié - en rouge la trajectoire initialement planifiée, en orange la trajectoire d'évitement

Prenons volontairement des destinations inatteignables dans les exemples présentés dans les figures V.14(a) et (b). Les rectangles indiquent l'infrastructure et les obstacles pour illustrer la situation, mais pour l'instant nous supposons ne pas disposer d'une carte d'obstacles. Par cet exercice nous souhaitons démontrer les limites d'une approche basée uniquement sur une perception instantanée tout en introduisant les avantages de l'utilisation des informations a priori (comme les cartes). Nous signalerons l'utilisation de la carte dans la stratégie d'évitement d'obstacles en colorant l'intérieur des cellules de la grille (en bleu, jaune, noir), comme par exemple dans la figure V.16(b).

Dans ce contexte, le système de planification de trajectoire est en charge de l'identification des situations de collision. Il est également tenu de calculer une trajectoire alternative en cas de collision. Les trajectoires colorées en orange dans les figures V.14(a) et (b) représentent les solutions d'évitement pour les situations présentées.

La technique que nous utilisons calcule la position de l'obstacle observé par rapport à l'axe de symétrie du véhicule. Dans le cas où le barycentre de l'obstacle est situé uniquement d'un côté de l'axe de symétrie longitudinale du véhicule, la trajectoire d'évitement se fera par le côté opposé. Par exemple les barycentres des obstacles illustrés dans les figures V.15(a) et (b) sont à gauche, par conséquent les trajectoires proposées sont par la droite de l'obstacle (dans le repère du véhicule).

Dans ces deux situations il n'est pas nécessaire de faire appel à la carte. Le système d'évitement d'obstacle fonctionnera sur le principe de *coussin virtuel*. C'est-à-dire, le véhicule évitera les obstacles sur un principe de champ de potentiel (champ répulsif). Ainsi même si la consigne

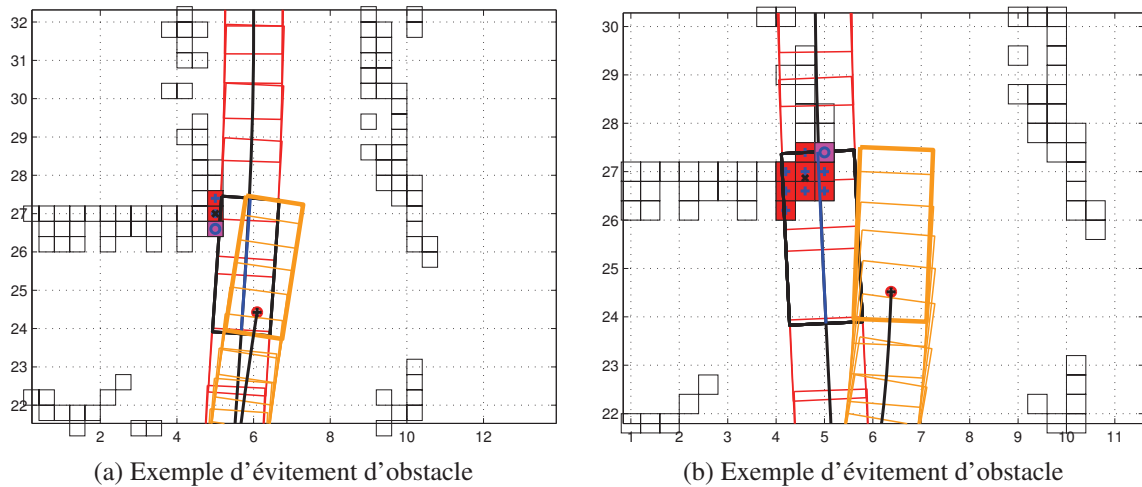


FIGURE V.15 – Exemple de situations de génération d'une nouvelle trajectoire pour des cas où les barycentres des obstacles observés sont situés à gauche de l'axe de symétrie longitudinale du véhicule

initiale est d'amener le véhicule sur une position de collision avec un mur, le comportement du véhicule sera de longer le mur.

Cette approche simple présente des limites. Dans l'exemple illustré dans la figure V.16(a) le barycentre de l'obstacle observé est à droite de l'axe longitudinal de symétrie. Cela conduit à construire une trajectoire vers la gauche. Dans la position initiale du véhicule et jusqu'à au premier contact avec un obstacle sur la trajectoire rouge il y a 11 m d'écart. Mais notre système de perception ne permet pas d'observer l'environnement avec précision depuis cette distance. Nous rappelons que le système de capteurs ultrason détecte dès 4,095 m mais avec une bonne précision dès 3 m et le système de perception par caméra (approche *Structure from Motion*) prend en compte uniquement les obstacles plus proches de 8 m. Dans le cas de figure présenté dans la figure V.16(a) nous observons que l'utilisation de la carte nous permettrait d'avoir une vision d'ensemble de la scène et de l'agencement des obstacles et qu'il serait possible de retrouver le couloir de navigation que le véhicule pourrait potentiellement emprunter.

V.3.3 Évitement d'obstacles dans un milieu cartographié

Comme nous disposons d'une carte d'obstacle réalisée lors d'une étape d'apprentissage, nous nous proposons de l'utiliser pour éteindre l'horizon de perception instantanée des capteurs du véhicule. Les cellules de la grille d'occupation représentant les obstacles cartographiés sont remplies d'une couleur (en bleu, jaune, noir) dans la situation où l'algorithme les utilise comme des a priori. Dans la figure V.16(b) la trajectoire d'évitement a été construite en prenant en compte le macro-obstacle (représenté par les rectangles jaunes) et pas seulement la partie directement visible en temps réel.

Pour réaliser la détection du macro-obstacle d'intérêt, toutes les cellules de la grille d'occupation situées dans le voisinage de la position d'impact et marquées comme occupées ont été regroupées utilisant l'algorithme DBSCAN que nous avons déjà utilisé dans §IV.4.4.1 (pour plus de détails voir également l'annexe A.1). Ainsi deux cellules situées à une distance plus petite que la largeur du véhicule plus une marge de sécurité ont été considérées faisant partie du même obstacle

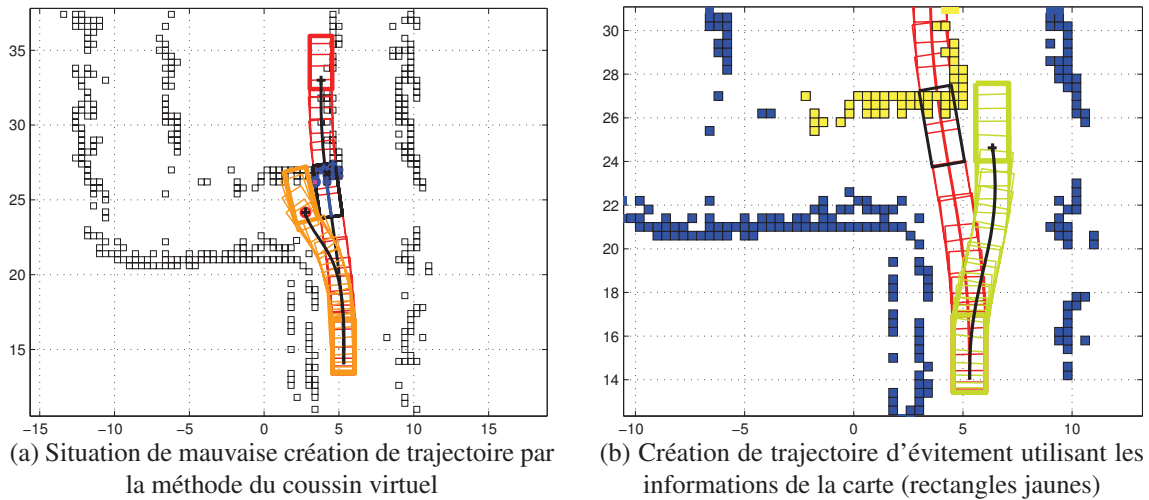


FIGURE V.16 – Résultats de simulation - Calcul écart latéral contournement obstacle

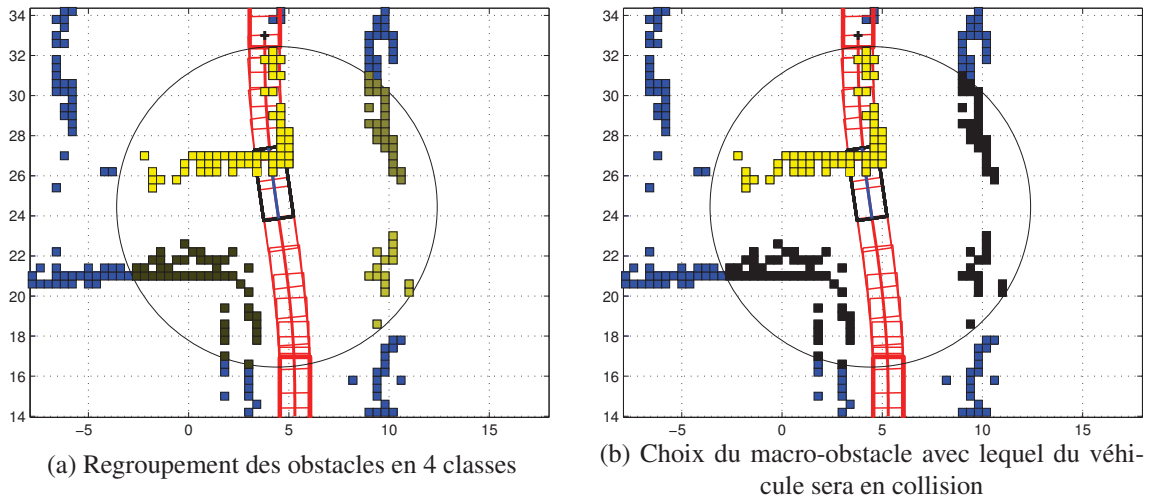


FIGURE V.17 – Regroupement des obstacles situés dans la proximité du véhicule un utilisant l'algorithme de classification DBSCAN

(voir figure V.17(a)). Enfin, parmi les macro-obstacles résultants, uniquement celui impliqué dans l'impact a été retenu dans le calcul de la nouvelle trajectoire d'évitement (voir figure V.17(b)).

Déterminer si le véhicule est en collision avec un obstacle est une opération qui est très utilisée et qui est coûteuse en temps du fait des calculs d'intersection avec chacun des arêtes du rectangle gabarit. C'est pour cela que nous utiliserons une approximation du rectangle englobant du véhicule par 5 cercles comme dans la figure V.18(a). Enfin si nous voulons introduire une distance de sécurité directement dans le gabarit du véhicule, alors les 4 cercles présentés dans la figure V.18(b) sont une bonne estimation.

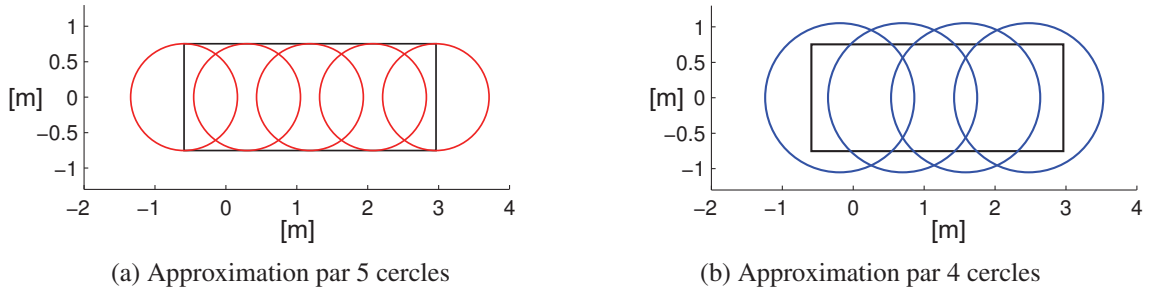
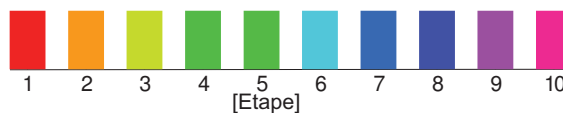
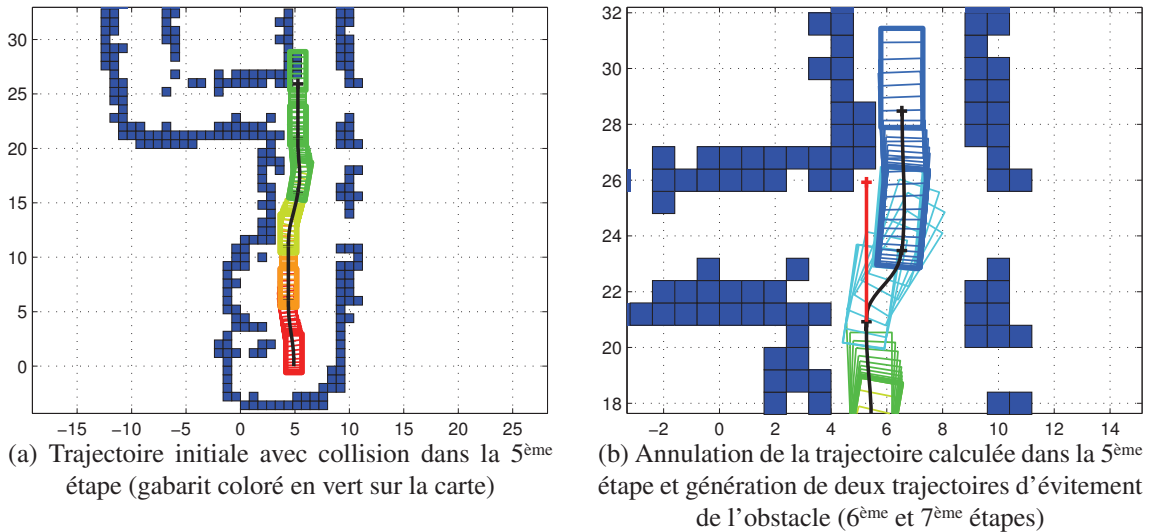


FIGURE V.18 – Approximation du gabarit du véhicule et éventuellement d'une distance de sécurité par des cercles

V.3.4 Déplacement dans un couloir de navigation cartographié avec prise en compte du gabarit du véhicule

Pour une meilleure compréhension de l'algorithme que nous avons implémenté, prenons directement des résultats finaux. Dans ce sens la figure V.19(a) illustre une situation de navigation où le système de planification crée des courtes trajectoires dans le but de naviguer vers le haut de la carte. Nous ne nous trouvons pas dans une approche de suivi de trajectoire de référence mais plutôt dans une approche de navigation vers une destination cible dans un couloir de navigation connu. Lors de la création du 5^{ème} segment de la trajectoire, le véhicule observe que dans son chemin il y a un obstacle qu'il devrait contourner. La figure V.19(b) illustre les trajectoires calculées dans la 6^{ème} et la 7^{ème} étape comme alternative à la trajectoire initiale.



(c) Coloration du gabarit du véhicule en fonction de l'étape de génération de trajectoires

FIGURE V.19 – Résultats de la génération successive de trajectoires pour le déplacement du véhicule dans un couloir de navigation connu. En cas de détection d'obstacles, la dernière trajectoire de collision est annulée et des alternatives sont proposées suivant la situation

Maintenant que nous avons exposé la stratégie de création des trajectoires nous revenons aux détails de la création de chaque segment de trajectoire.

V.4 Génération d'une section de trajectoire

Dans les approches présentées dans l'état de l'art, les courbures initiales et finales étaient nulles. Nous nous proposons à présent de créer une méthode de construction de trajectoires sous optimales à courbure continue également pour des courbures initiales et finales non nulles. Cela nous permettra de construire des trajectoires à tout moment et pour toute configuration du véhicule.

V.4.1 Trajectoire bi-élémentaire avec courbures initiales et finales non nulles

Dans l'état de l'art nous avons présenté qu'une configuration q_s située sur une clothoïde à une distance curviligne s de la configuration $q_0(x = 0, y = 0, \theta = 0, \kappa = 0)$ est égale à :

$$q_s = (x, y, \theta, \kappa) = \left(A\sqrt{\pi} C_f \left(\frac{s}{\pi A} \right), A\sqrt{\pi} S_f \left(\frac{s}{\pi A} \right), s\kappa, \kappa \right) \quad (\text{V.12})$$

avec A le paramètre de la clothoïde ($A^2 = 2s/\kappa$) et $C_f(s)$ et $S_f(s)$ les intégrales de Fresnel. Nous rappelons qu'une clothoïde est une courbe dont la courbure κ varie linéairement avec son abscisse curviligne s ainsi :

$$\kappa(s) = \sigma s + \kappa(0) \quad (\text{V.13})$$

avec σ une constante caractérisant la clothoïde et $\kappa(0)$ la courbure de la trajectoire dans l'origine $s = 0$. Si dans l'état de l'art nous avons présenté des exemples de clothoïdes avec courbure initiale nulle, maintenant nous nous intéressons au cas plus général où $\kappa(0) \neq 0$.

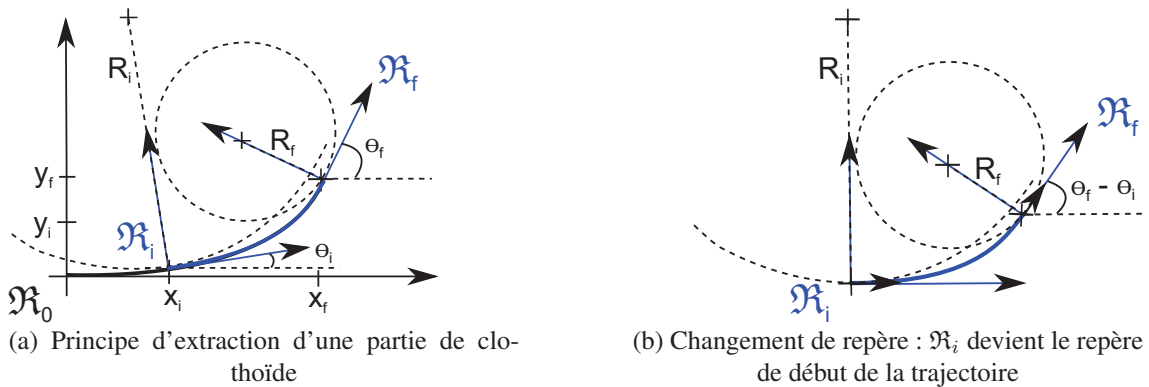


FIGURE V.20 – Construction d'une trajectoire clothoïdale pour une courbure initiale $\kappa(0) \neq 0$

Calculer une clothoïde pour une courbure initiale donnée revient à extraire la partie de la clothoïde qui nous intéresse comme dans la figure V.20(a). Ensuite, il faudrait réaliser un changement de repère prenant l'origine le repère \mathfrak{R}_i tel que l'illustre la figure V.20(b). Pour réaliser cela nous avons besoin des paramètres de la roto-translation (x_i, y_i, θ_i) caractérisant le changement de repère de \mathfrak{R}_0 à \mathfrak{R}_i .

Par s_i et s_f nous notons la longueur du trajet clothoïdal entre de début de la clothoïde où la courbure κ_0 est nulle (courbure $\kappa(0) = 0$) et la position où la courbure est égale à κ_i respectivement κ_f . Utilisant l'équation (V.12) nous pouvons affirmer que θ_i est une fonction qui

dépend des paramètres κ_i et s_i . Connaissant également que $A^2 = 2s_i/\kappa_i$, alors θ_i s'écrit comme une fonction de κ_i et un paramètre A caractéristique à la forme de la clothoïde.

$$\theta_i = f(\kappa_i, A) \quad (\text{V.14})$$

Enfin pour calculer les paramètres de translation x_i et y_i nécessaires dans le changement de repère nous utilisons le développement en série entière donné par l'équation (V.15). Pour le calcul en temps réel nous réalisons une approximation en utilisant un n relativement petit ($6 \leq n \leq 8$ tel que nous l'avons montré par simulation dans le tableau V.1 de la section §V.2.2.1).

$$\begin{cases} x_i = \sum_{n=0}^{\infty} \frac{(-1)^n s_i^{4n+1}}{(2n)!(4n+1)(\sqrt{2}A)^{4n}} \\ y_i = \sum_{n=0}^{\infty} \frac{(-1)^n s_i^{4n+3}}{(2n+1)!(4n+3)(\sqrt{2}A)^{4n+2}} \end{cases} \quad (\text{V.15})$$

Si nous reprenons le même raisonnement utilisé pour déduire les paramètres qui définissent θ_i , nous retrouvons que θ_f peut être écrit comme une fonction $\theta_f = f(\kappa_f, A)$ dans le repère \mathfrak{R}_0 . En tenant compte du fait que nous voulons retrouver un certain θ directement dans le repère \mathfrak{R}_i , alors dans la figure V.20(b) l'angle recherché sera $\theta = \theta_f - \theta_i$. Ainsi θ pourra être écrit comme une fonction $\theta = g(\kappa_i, \kappa_f, A)$. En pratique la courbure κ_i et la position des roues directrices sont liées, par conséquent κ_i sera imposé par l'état du véhicule lors de la création de la trajectoire. Ce qui signifie qu'une portion de clothoïde ne sera caractérisée que par deux variables : κ_f et A .

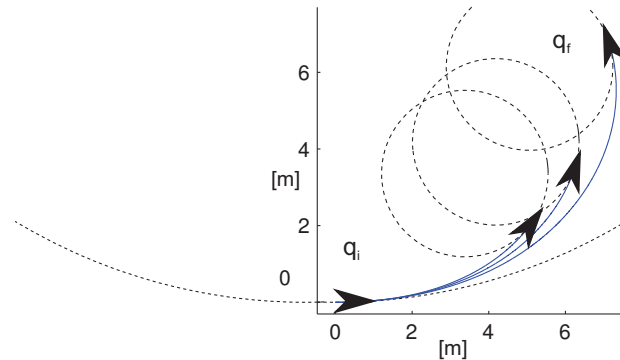


FIGURE V.21 – Trajectoires à dérivée de courbure constante construites en variant le paramètre A caractérisant la forme de la clothoïde et en gardant constantes les courbures initiales et finales non nulles.

Le résultat obtenu peut être observé dans la figure V.21. Nous construisons deux faisceaux de trajectoires clothoïdales en imposant la courbure initiale et finale et en variant uniquement le paramètre A caractérisant la forme de la clothoïde. Il est à remarquer que, utilisant la forme discrète des coordonnées de la clothoïde, nous n'avons pas besoin de calculer tous les points dans la composition de la trajectoire, mais nous pouvons déduire directement la position d'arrivée du véhicule pour un coût calculatoire insignifiant. Une fois la position validée, nous pouvons revenir sur la création des points qui définisse la trajectoire. Cette méthode simple et rapide de création de portions de trajectoires sera utilisée par la suite dans la construction des trajectoires plus complexes.

V.4.2 Généralisation de la méthode de création de trajectoire par séquences VCC

Nous rappelons que dans l'état de l'art nous avons présenté la méthode de création de virages pour des profils de courbures symétriques proposée par [Scheuer, 1998] et [Vorobieva, 2014]. Utilisant deux clothoïdes et optionnellement un arc de cercle (dans des situations où les roues braquent au maximum), un certain type de virage sous optimal à courbure continue est obtenu et désigné par l'acronyme VCC (Virage à courbure continue). La mise en commun de deux séquences VCC peut donner naissance à un virage en «S» tel que la figure V.22 le montre. L'utilisation des séquences VCC peut être vue aussi comme méthode de transformation des 6 chemins proposés par [Dubins, 1957] (LRL, RLR, RSL, RSR, LSR et LSL) dans des virages à courbure continues.

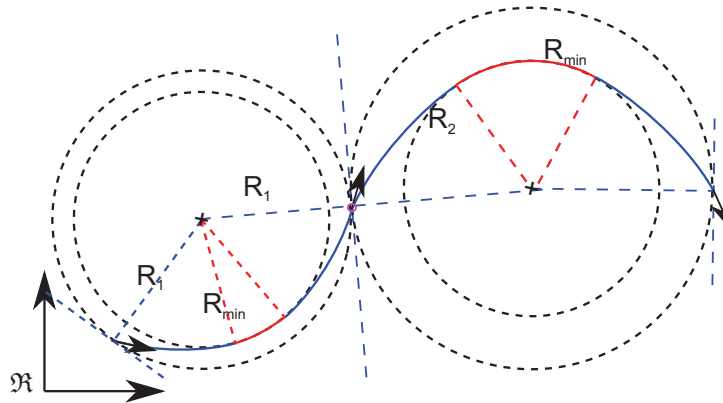


FIGURE V.22 – Deux séquences VCC reliées dans une courbe en «S» [Vorobieva, 2014]

Comme nous l'avons déjà présenté, nous souhaitons étendre l'usage des virages à courbure continue pour le cas où les courbures initiales et finales ne sont pas nulles. À titre d'exemple nous prenons les deux séquences VCC proposées dans la figure V.22 et nous les découpons dans des courbes élémentaires constituées des portions de clothoïdes et d'arcs de cercle (voir figure V.23).

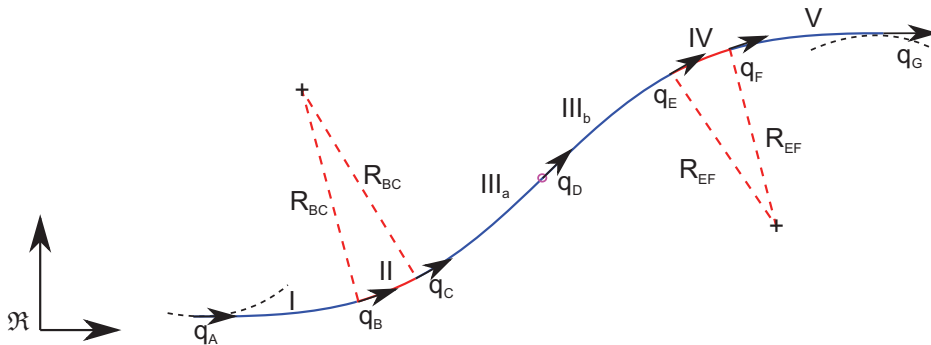


FIGURE V.23 – Généralisation du problème de trajectoire à courbure continue en forme de «S» pour courbures initiales et finales non nulles

Nous nous intéressons maintenant à la complexité du problème. Pour cela nous calculons le nombre de paramètres qui décrivent la configuration finale q_G . Nous rappelons qu'une portion de clothoïde est définie par une fonction $f_{\text{clothoïde}}$ dépendant de trois paramètres κ_i , κ_f et A . En ce qui concerne un arc de cercle, il est exprimé par une fonction $f_{\text{arc de cercle}}$ caractérisée par deux paramètres : la courbure κ et la longueur de l'arc L_{arc} .

Suite à la synthèse réalisée dans le tableau V.2, nous déduisons que la courbe est définie par 10 paramètres dont un imposé par l'état du véhicule dans la configuration q_A : κ_A la courbure

initiale imposée par l'angle de la roue directrice moyenne β_A .

Courbe	Type	Fonction	Contraintes
I	clothoïde	$f_I(A_I, \kappa_A, \kappa_B)$	κ_A imposé
II	arc de cercle	$f_{II}(\kappa_B, L_{BC})$	
III _a	clothoïde	$f_{III_a}(A_{III_a}, \kappa_C, \kappa_D)$	$\kappa_C \equiv \kappa_B$ et $\kappa_D = 0$
III _b	clothoïde	$f_{III_b}(A_{III_b}, \kappa_D, \kappa_E)$	$\kappa_D = 0$
IV	arc de cercle	$f_{IV}(\beta_E, L_{DE})$	
V	clothoïde	$f_V(A_V, \kappa_F, \kappa_G)$	$\kappa_F \equiv \kappa_E$

TABLE V.2 – Calcul de nombre de paramètres définissant la trajectoire illustrée dans la figure V.23. En rouge sont les paramètres inconnus, en bleu sont les paramètres donnés en état initial ou imposés par une contrainte géométrique

Pour comprendre la complexité du problème, regardons les hypothèses réalisées par [Vorobieva, 2014] dans son travail de recherche sur la génération de trajectoires pour le parking latéral utilisant des courbes tel que le montre la figure V.22 :

- courbures initiales et finales nulles : $\kappa_A = 0$ et $\kappa_G = 0$;
- clothoïdes symétriques dans une séquence VCC : $A_I = A_{III_a}$ et $A_{III_b} = A_V$;
- même la vitesse de braquage pour les deux séquences VCC : $A_I = A_{III_a} = A_{III_b} = A_V$;
- si le braquage maximal est atteint : $\kappa_B = \kappa_C = \kappa_{max}$ et $\kappa_E = \kappa_F = \beta_{max}$.

Dans ces conditions, sur les 9 inconnues de la fonction qui décrit la trajectoire présentée dans la figure V.23, 6 seront fixées, restant à déterminer uniquement les 3 paramètres suivants : A_I , L_{BC} et L_{EF} . Dans le cas où le braquage maximal n'est pas atteint pour une des séquences VCC, alors l'arc de cercle sera de longueur nulle et il est nécessaire d'ajouter un paramètre pour caractériser la courbure en fin de la clothoïde qui précède l'arc de cercle. Dans notre cas L_{BC} et L_{EF} seront remplacés par κ_B respectivement κ_E . Même si cette courbe en «S» a l'avantage d'être facile à construire (figure V.22), elle limite trop l'allure de la trajectoire ou dans certains cas n'est pas capable de relier deux configurations. Pour ces deux raisons, nous nous proposons d'utiliser la forme généralisée.

Pour l'implémenter nous choisissons une approche par minimisation. Soit $f_{(I-V)}$ une fonction qui caractérise en ensemble les trajectoires élémentaires notées de I à V. La seule hypothèse réalisée (que le nom le suggère déjà) est que les trajectoires III_a et III_b sont caractérisées par le même paramètre de clothoïde ($A_{III_a} \equiv A_{III_b}$). Cela revient à utiliser sur cette portion une trajectoire bi-élémentaire avec courbure initiale et finale non nulle de signe différent. Dans ces conditions $f_{(I-V)}$ s'écrit ainsi :

$$\begin{cases} (x, y, theta) = f_{(I-V)}(X) \\ X = [\kappa_A, A_I, \kappa_B, L_{BC}, A_{III}, \kappa_E, L_{DE}, A_V, \kappa_G]^T \end{cases} \quad (V.16)$$

Nous gardons la représentation en couleurs bleu et rouge pour mentionner si un paramètre est imposé respectivement s'il doit être déterminé. La méthode que nous utilisons pour déterminer les paramètres s'appelle PSO - *Particle Swarm Optimization* en anglais ou bien *optimisation par essais particuliers*.

PSO est un algorithme d'optimisation qui appartient à la classe d'algorithmes métaheuristiques d'optimisation. Il est inspiré du comportement des insectes (ou bien des animaux) et il est

basé sur le paradigme de l'intelligence des essaims. Même si le principe est ancien, il a été formalisé pour la première fois par J. Kennedy et R. C. Eberhart [Eberhart et al., 1995]. Il suppose d'utiliser plusieurs *particules* qui évoluent dans l'espace de recherche. Chaque particule se déplace avec une certaine vitesse vers une nouvelle position suivant des critères qui tiennent à la fois de l'attitude personnelle (composante cognitive) mais également de l'attitude du groupe (composante sociale). Dans l'annexe A.8 nous détaillons l'algorithme pour un cas unidimensionnel.

Pour la suite nous prenons un critère à minimiser décrit par la fonction de coût composée de trois distances (voir équation (V.17)). Pour contrebalancer l'effet du poids de chacune des distances nous utilisons des pondérations : σ_x^2 , σ_y^2 et respectivement σ_θ^2 . Cela nous permet aussi de pondérer différemment chacune des éléments de la position finale souhaitée (x^* , y^* , θ^*).

$$\min_X \left(\frac{\|x^* - f_{(I-V)}^x(X)\|_2^2}{\sigma_x^2} + \frac{\|y^* - f_{(I-V)}^y(X)\|_2^2}{\sigma_y^2} + \frac{\|\theta^* - f_{(I-V)}^\theta(X)\|_2^2}{\sigma_\theta^2} \right) \quad (\text{V.17})$$

La figure V.24(a) montre la courbe d'évolution de l'erreur de la fonction de coût suivant l'itération du processus d'optimisation. Les paramètres obtenus permettent de reconstruire la trajectoire recherchée pour atteindre la position (x^* , y^* , θ^*) (voir figure V.24(b)).

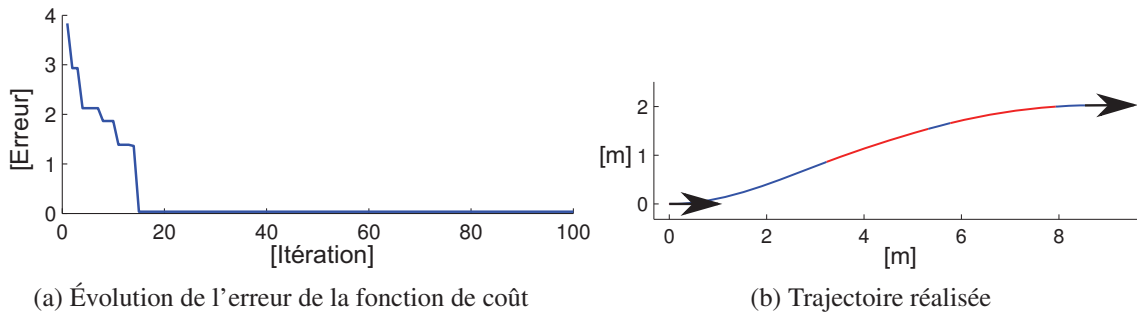


FIGURE V.24 – Création de trajectoire avec un modèle imposé (voir figure V.23) et choix des paramètres utilisant une méthode de minimisation d'une fonction de coût par la méthode PSO

V.4.3 Résultats sur la génération de trajectoire

Les figures V.25(a) et (b) montrent deux faisceaux de trajectoires. Il est intéressant d'observer dans la figure V.25(a) les trajectoires pour rejoindre les positions finales signalées par des flèches en couleur magenta. Comme l'algorithme n'a pas été capable de trouver une solution respectant simultanément les trois critères de distance. En utilisant une pondération différente, nous avons choisi de signaler qu'il est plus important de trouver une trajectoire qui respecte la contrainte latérale et angulaire dans le détriment de la contrainte longitudinale.

Par rapport aux faisceaux de trajectoires présentés dans les figures V.11(a) à (d), ceux présentés dans les figures V.25(a) et (b) sont composés en totalité des trajectoires admissibles et cela est garanti par la méthode de construction de chacune de trajectoires.

Nous concluons cette partie en affirmant que l'utilisation d'un module de création de trajectoire (ou de faisceaux de trajectoires) par une approche généralisée de la méthode basée sur des séquences VCC nous permet d'avoir une gestion claire du déplacement demandé au véhicule.

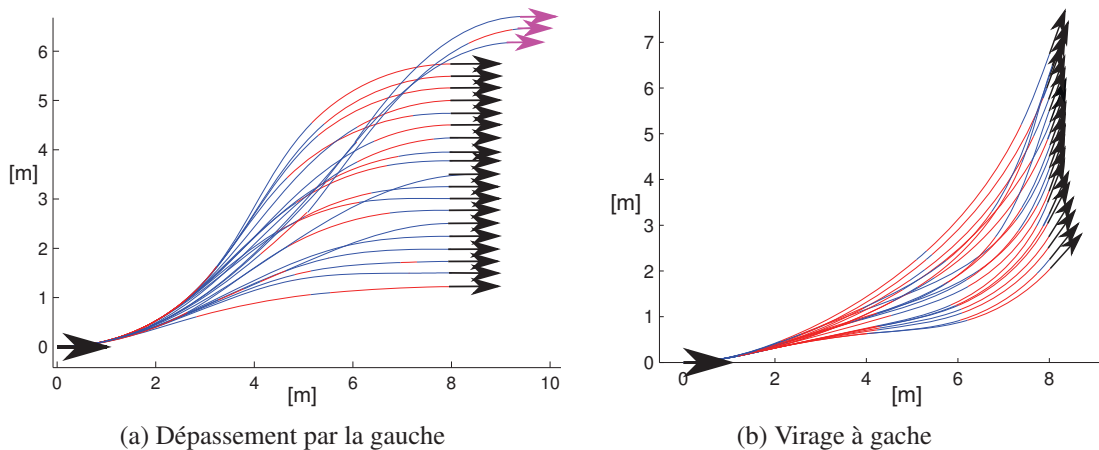


FIGURE V.25 – Faisceaux de trajectoires admissibles construits par la généralisation de la méthode de création de trajectoire par séquences VCC

Ainsi, nous pouvons mettre en pratique la stratégie d'évitement d'obstacle que nous avons énoncée dans la section §V.3 : utiliser la trajectoire de référence créée lors de l'apprentissage plus une technique de planification locale en cas d'évitement d'obstacle.

Le dernier sujet que nous abordons et que nous présenterons dans la section suivante est le contrôle/commande des actionneurs du véhicule pour le suivi de la trajectoire précédemment créée.

V.5 Commande des actionneurs

Le dernier sujet que nous abordons dans ce chapitre est la commande du véhicule. Même si le contrôle des actionneurs ne fait pas l'objet du travail de recherche, nous nous sommes intéressés à ce sujet pour avoir une vision d'ensemble des problématiques liées au véhicule autonome.

V.5.1 Gestion de la distance de sécurité et du freinage d'urgence

Avant de réaliser des manœuvres pilotées en automatique nous nous sommes intéressés à la sécurité du véhicule. Nous avons considéré qu'il est nécessaire de créer une architecture système robuste à la défaillance du module d'évitement d'obstacle. Dans ce sens, nous avons conçu un niveau bas de contrôle qui fonctionne indépendamment des décisions haut niveau du système de voiturier autonome. Ainsi le résultat est un système capable de freiner le véhicule en sécurité pour toute situation de conduite autonome à basse vitesse.

Cette fonctionnalité pourrait être utilisée par le conducteur pour toute manœuvre de conduite à basse vitesse, pas uniquement pour le voiturier à domicile. Il est souvent considéré que le caractère répétitif d'une tâche implique une augmentation des facteurs de risque d'accident par diminution de l'attention [Banovic et al., 2016]. Par conséquent, cette fonctionnalité a toute son importance dans les situations de conduite manuelle comme celle effectuée dans la phase d'apprentissage préalable à la conduite autonome ou toute autre situation de parking.

Pour tout cela, le système de sécurité surveille en permanence les positions des obstacles et actionne les freins automatiquement en cas de détection du franchissement de zone jaune ou rouge pour ralentir ou arrêter le véhicule (voir figure V.26). De plus, ce module de freinage prend le relais si le module de planification de trajectoire ou celui de suivi de trajectoire tombent en panne ou se retrouvent dans un cas de figure non traité.

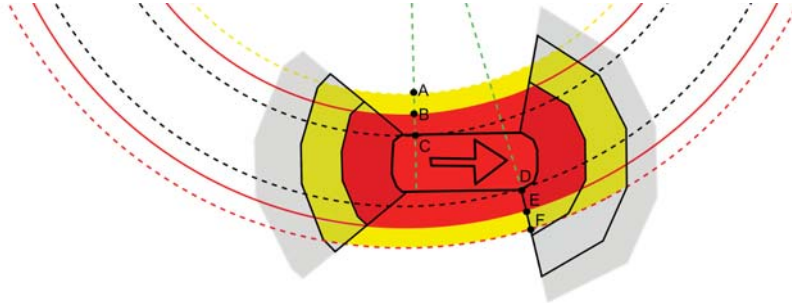


FIGURE V.26 – Représentation détaillée de de la zone de collision (en rouge) et des zones de ralentissement (en jaune)

La figure V.26 représente en détail les zones d'intérêt : en couleur jaune, la zone de ralentissement et en couleur rouge la zone de collision, enfin les zones assombries correspondent aux champs de perception du système de capteurs ultrason. Pour la construction des zones jaunes et rouges nous avons tenu compte de la trajectoire circulaire que le véhicule décrit dans un virage et de la distance curviligne de ralentissement/freinage.

Notons par $CC(O, R_{int}, R_{ext})$ la couronne circulaire de centre O , de rayon intérieur R_{int} et rayon extérieur R_{ext} . En reprenant les notations de la figure V.26, l'espace encombrant est défini par la couronne circulaire $CC(O_{CIR}, \|O_{CIRC}\|, \|O_{CIRD}\|)$. La couronne circulaire dans la composition de la zone rouge a été élargie par la distance de sécurité $\|BC\|$ du côté gauche respectivement $\|DE\|$ du côté gauche du véhicule. Sur le même principe, la zone jaune a été créée en élargissant la zone rouge par la distance $\|AB\|$ respectivement $\|EF\|$.

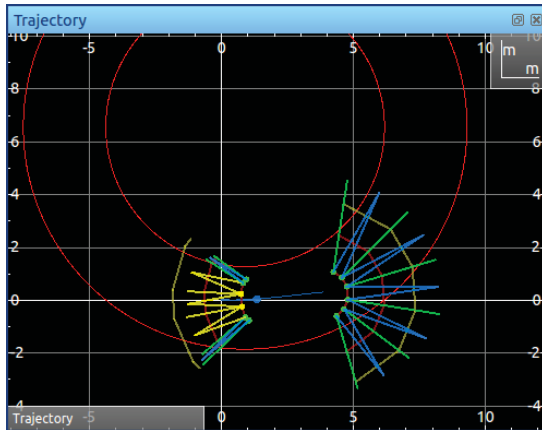
Comme nous l'avons déjà exprimé, la fréquence de mise à jour de l'information issue du capteur ultrason est de 10 Hz.

$$d_{Sécurité} = v_{veh} [(\Delta t_{Détection1} + \Delta t_{Communication1}) + (\Delta t_{Détection2} + \Delta t_{Communication2})] \quad (V.18)$$

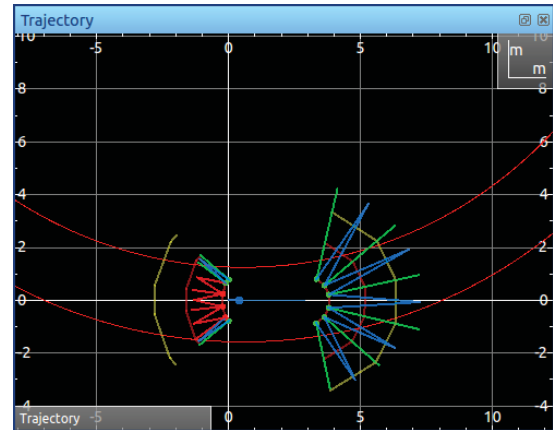
Concernant la distance de freinage d'urgence ($d_{Sécurité}$), elle est donnée par l'équation (V.18), où $\Delta t_{Détection1}$ et $\Delta t_{Détection2}$ représentent les intervalles de temps écoulés jusqu'à la première respectivement à la deuxième détection de l'obstacle. Pour éviter les freinages intempestifs dus à la fausse détection, le système attendra toujours une deuxième trame de confirmation de la détection. En prenant le cas critique, nous constatons que, lors de l'apparition d'un obstacle dans le couloir du véhicule, le système a besoin d'un temps de détection plus un temps de communication qui vaut dans le pire de cas 200 ms.

Comme nous l'avons déjà présenté en bilan du chapitre IV (voir §IV.5.2), il existe des obstacles qui sont invisibles à la détection par une approche *Structure from Motion*, notamment les obstacles monochromes sur un fond de la même couleur. Par conséquent ces obstacles sont détectables par le véhicule uniquement quand ils se trouvent dans le champ de perception des capteurs ultrason (4,095 m). Cela nous a conduit à choisir une vitesse maximale de 1,5 m/s.

Donc pour une vitesse de véhicule de 1,5 m/s ($\approx 5,5$ km/h), le véhicule parcourt théoriquement 0,3 m avant que le système de freinage ne se déclenche. Lors des essais de freinage, nous avons constaté que le système de freinage peut mettre à lui seul jusqu'à 400 ms pour serrer les freins, ce qui nous amène à 600 ms de temps mort. En conséquence nous avons construit un système de gestion de la vitesse de déplacement qui prend en compte le gabarit en ligne droite et en virage du véhicule et la distance à l'obstacle.



(a) Détection d'obstacle et ralentissement automatique en marche arrière



(b) Détection d'obstacle et freinage d'urgence

FIGURE V.27 – Ralentissement automatique (figure (a)) et freinage d'urgence automatique (figure (b)) suite à la détection d'un obstacle lors d'une manœuvre de conduite manuelle du véhicule en marche arrière

Enfin dans les figures V.27(a) et (b) nous présentons une situation de conduite manuelle en marche arrière où le système de surveillance de la distance de sécurité est activé et qui bride la consigne de vitesse pour ralentir le véhicule lors de la détection d'un obstacle dans la zone jaune, enfin de freinage d'urgence lorsque celui-ci franchit la zone rouge.

V.5.2 Commande des actionneurs pour le suivi de trajectoire

Dans ce travail de thèse nous avons utilisé l'idée qu'un robot autonome est simplement un robot commandable qui suit une trajectoire admissible et qui respecte un profil de vitesse adapté. Cette philosophie n'est pas nouvelle et nous la retrouvons également par exemple dans la démonstration de véhicule autonome qu'Audi a faite pour CES 2013 et que nous avons présenté dans le chapitre d'introduction [page 17]. Dans cette démonstration, il a été proposé de réduire le coût du système de voiturier autonome en installant les systèmes de perception et l'intelligence, de manière générale, à l'extérieur du véhicule pour les partager avec toute une flotte de véhicules. Ainsi chaque véhicule ne faisait que respecter une trajectoire et une vitesse envoyées par l'unité centrale.

Nous avons expliqué dans les sections précédentes la technique de création d'une trajectoire admissible entre deux poses (ou configurations) du véhicule. Un des grands avantages dans l'utilisation de clothoïdes et des arcs de cercle est la simplicité de la création de la consigne de braquage des roues directrices. Par conséquent, si nous voulons que le véhicule suive une trajectoire clothoïdale, il suffit de commander les roues directrices suivant un angle de braquage β (angle de la roue directrice du modèle bicyclette) qui est déterminé simplement dans l'équation de la clothoïde. Bien entendu, l'admissibilité de la trajectoire générée est garantie si et seulement si la

vitesse d'avancement du véhicule est compatible avec le temps de réponse des actionneurs. Par la suite nous nous intéressons au calcul de cet angle de braquage pour le suivi de trajectoire.

Nous rappelons que la courbure κ dans chaque point de la trajectoire est définie comme l'inverse du rayon du cercle osculateur. Par conséquent une trajectoire circulaire sera décrite par le véhicule simplement en gardant une courbure constante ($\kappa(s) = \kappa_{cst}$) et une trajectoire clothoïdale sera réalisée en variant la courbure suivant la loi $\kappa(s) = \sigma s + \kappa(0)$. Enfin l'angle de braquage de la roue moyenne directrice pour un modèle bicyclette est donné par l'équation suivante :

$$\beta(s) = \chi \left| \arctan \left(\frac{L}{R} \right) \right| \quad (\text{V.19})$$

avec $L = a_1 + a_2$ la distance entre l'axe des roues avant et arrières et $\chi = 1$ correspondant à un braquage à gauche et $\chi = -1$ pour un braquage à droite.

Dans ses travaux de recherche [Vorobieva, 2014] a montré que pour un parking latéral, donc pour des trajectoires très courtes, un véhicule peut être commandé en boucle ouverte au long de chaque manœuvre. Éventuellement dans le cas d'une grande dérive, une nouvelle trajectoire était régénérée pour prendre en compte la correction souhaitée. Pour commander le véhicule, l'auteur propose de réaliser en avance le profil de vitesse à respecter et parallèlement la consigne de braquage suivant l'abscisse curviligne s .

Dans notre cas, la trajectoire à suivre est plus longue que dans le cas de parking latéral et l'utilisation d'une boucle de contrôle continu s'impose. La problématique du suivi de trajectoire par un robot est un sujet largement détaillé dans la littérature scientifique [Lenain et al., 2004, 2006; Samson et al., 2016]. Nous nous proposons uniquement d'appliquer une technique adaptée à notre problème et de montrer des résultats préliminaires.

Une méthode est d'utiliser la commande par linéarisation exacte. Cette méthode assure une bonne précision à courbure constante et présente l'avantage de découpler l'angle de braquage de la vitesse du véhicule. Mais en pratique des instabilités ou des dépassements sont constatés car les hypothèses fortes assumées ne sont pas satisfaites : les actionneurs sont idéaux (vitesse de braquage infini et pas de temps mort), pas de latence pour l'observation de la dérive et enfin courbure lentement variable.

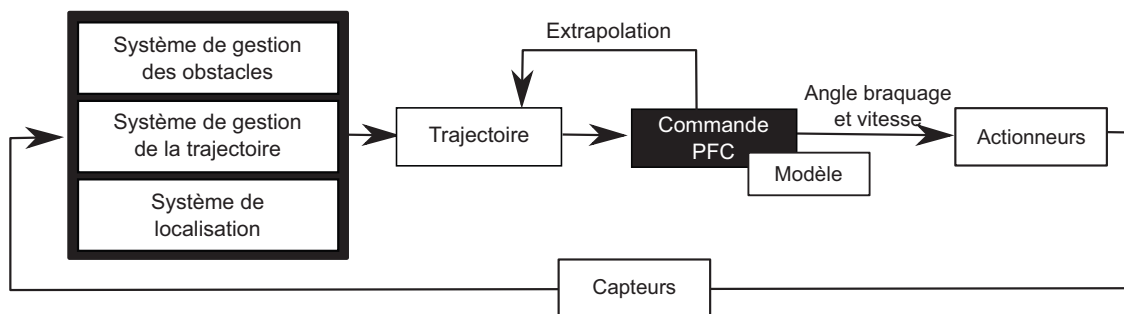


FIGURE V.28 – Schéma de principe du fonctionnement du système de suivi de trajectoire par approche predictive (PFC - Predictive Functional Control)

Si l'on avait disposé d'actionneurs idéaux, alors il aurait pu être possible de créer une loi de commande permettant de trouver à un instant donnée la commande idéale à appliquer aux roues directrices de façon à ce que le véhicule suive la trajectoire de référence. Mais en pratique le temps de réponse non nul des actionneurs implique qu'en entrée et en sortie des virages l'écart latéral augmente. Afin de corriger ces écarts enregistrés pendant les phases transitoires, il est possible d'utiliser une anticipation de la courbure de trajectoire et de créer une commande dite prédictive.

Même s'il est difficile de créer une commande idéale, il existe des méthodes capables d'intégrer des a priori. Par exemple, connaissant l'état actuel, il est possible d'anticiper la commande utilisant le temps de réponse des actionneurs et la courbure future de la trajectoire. Dans ce sens, notre choix a été d'utiliser la technique de *commande prédictive à modèle* ou PFC - Predictive Functional Control en anglais que nous présentons schématiquement dans la figure V.28 et que nous détaillerons par la suite.

V.5.2.1 Commande prédictive à modèle

Comme le nom l'indique, le principe de la méthode repose sur l'utilisation d'un modèle du système pour prédire son comportement futur. Ainsi, il est possible de construire une consigne ou une séquence de consignes pour que le système atteigne l'état souhaité dans un horizon de temps (voir figure V.30).

A un instant donné, l'angle de la roue moyenne directrice du véhicule est caractérisé par une composante liée à la trajectoire β^{traj} et une deuxième composante appelée déviation par rapport à la trajectoire $\beta^{déviation}$. Dans ce contexte β s'exprime ainsi :

$$\beta = \beta^{traj} + \beta^{déviation} \quad (V.20)$$

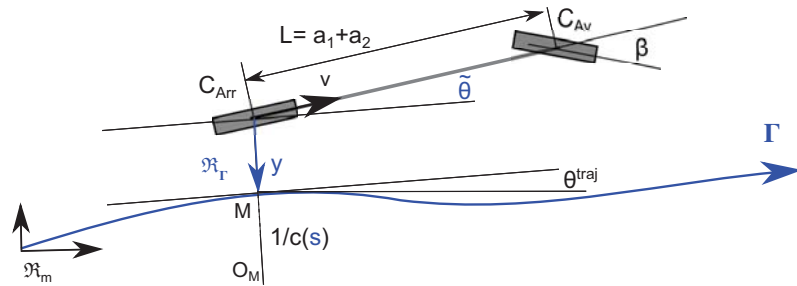


FIGURE V.29 – Modèle bicyclette simple

Connaissant que de manière générale le rayon décrit par le centre de l'essieu arrière du véhicule autour d'un centre instantané de rotation (CIR) est égale à $R_{CIR} = L / \tan \beta$, alors dans notre cas β^{traj} est définie ainsi (voir figure V.29) :

$$\beta^{traj} = \arctan \left(\frac{L \cos \tilde{\theta}}{\frac{1}{c(s)} - y} \right) = \arctan \left(\frac{L c(s) \cos \tilde{\theta}}{\alpha} \right) = \arctan(\lambda_1) \quad (V.21)$$

Par la méthode de commande par linéarisation exacte nous savons que l'angle de braquage β s'écrit sous la forme suivante :

$$\beta = \arctan \left(\frac{L \Lambda \cos^3 \tilde{\theta}}{\alpha^2} + \frac{L c(s) \cos \tilde{\theta}}{\alpha} \right) \quad (V.22)$$

avec $\Lambda = -K_p y - K_d \alpha \tan \tilde{\theta} + c(s) \alpha \tan^2 \tilde{\theta}$ et $\alpha = 1 - c(s)y$. Utilisant les équations (V.20) à (V.22) nous déduisons que la composante de l'angle de braquage liée à la déviation par rapport à la trajectoire s'exprime ainsi :

$$\beta^{déviation} = \arctan \left(\frac{\lambda_2}{1 + \lambda_1 \lambda_2 + \lambda_1^2} \right) \quad (V.23)$$

où, pour la simplicité des équations, nous avons noté :

$$\lambda_1 = \frac{L c(s) \cos \tilde{\theta}}{\alpha} \quad \text{et} \quad \lambda_2 = \frac{L \Lambda \cos^3 \tilde{\theta}}{\alpha^2} \quad (\text{V.24})$$

Notre objectif est de calculer l'angle de braquage β de tel façon que le centre de l'essieu arrière du robot suit la trajectoire, cela se traduit par distance nulle ($y \rightarrow 0$) et angle de dérive nul ($\tilde{\theta} \rightarrow 0$) par rapport à la trajectoire dans le point d'abscisse curviligne s .

Dans la section §V.3.1 sur la problématique d'évitement d'un obstacle nous avons fait référence à la possibilité d'utiliser un raisonnement en termes d'écart latéral par rapport à la trajectoire. Typiquement $y \rightarrow \text{écartLat}$ où $\text{écartLat} \neq 0$. Mais nous n'avons pas retenu cette approche car nous avons souhaité utiliser une trajectoire préalablement construite avant de demander au véhicule de la suivre. Par la suite nous restons sur l'hypothèse que l'écart latéral est nul.

Supposons qu'à l'instant présent la consigne est à l'itération n . Pour atteindre une certaine consigne objectif β^{trajObj} dans un horizon $n + n_H$, la modalité n'est pas unique. Par exemple, une méthode simple est d'utiliser une extrapolation par un polynôme de premier ordre pour déterminer l'angle de braquage β^{trajRef} à tout moment $n + i$:

$$\beta_{[n+i]}^{\text{trajRef}} = \beta_{[n+n_H]}^{\text{trajObj}} - \gamma^i \left(\beta_{[n]}^{\text{trajObj}} - \beta_{[n]}^R \right) \quad (\text{V.25})$$

ou $\beta_{[n]}^R$ représente la sortie du processus à l'instant présent.

Utilisant l'équation (V.21) pour les conditions d'atteinte de objectif $y \rightarrow 0$ et $\tilde{\theta} \rightarrow 0$, nous avons :

$$\begin{cases} \beta_{[n]}^{\text{trajObj}} = \arctan(L c(s)) & , \text{à l'instant présent } [n] \\ \beta_{[n+n_H]}^{\text{trajObj}} = \arctan(L c(s + s_H)) & , \text{à l'instant } [n + n_H] \end{cases} \quad (\text{V.26})$$

où par définition $s_H = \dot{s}H$. Utilisant cette écriture, nous disposons de l'angle de braquage idéal à suivre que nous appelons par la suite angle de braquage de référence : $\beta_{[n+i]}^{\text{trajRef}}$.

Si nous souhaitons réaliser un algorithme de commande adapté à notre véhicule alors, il faut inévitablement passer par une étape de modélisation de l'actionneur. Dans ce sens il est possible d'exprimer son comportement sous la forme du système d'équations (V.27) :

$$\begin{cases} \vec{X}_{[n]} = \mathbf{A} \vec{X}_{[n-1]} + \vec{B} \beta_{[n-1]}^C \\ Y_{[n]} = \vec{C}^T \vec{X}_{[n]} \end{cases} \quad (\text{V.27})$$

où $\vec{X}_{[n]} = \begin{bmatrix} \beta_{[n]}^M \\ \beta_{[n-1]}^M \\ \beta_{[n-1]}^C \end{bmatrix}$, $\mathbf{A} = \begin{bmatrix} a_1 & a_2 & b_2 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\vec{B} = \begin{bmatrix} b_1 \\ 0 \\ 1 \end{bmatrix}$, $\vec{C} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $Y_{[n]} = \beta_{[n]}^M$. Enfin

a_1 , a_2 , b_1 , b_2 sont des variables représentant le comportement spécifique de l'actionneur. Dans ce contexte β^M représente la *mesure* en sortie du processus et β^C représente la *commande*.

Mais la réponse du système à l'instant $n + i$ est la somme entre la *réponse libre du système* (qui dépend de valeurs précédentes et du bas niveau) et la *réponse forcée du système* (qui dépend uniquement de la commande) et cela s'exprime sous forme de l'équation suivante :

$$\hat{\beta}_{[n+i]}^M = \vec{C}^T \mathbf{A}^i \vec{X}_{[n]} + \mu(n)^T \hat{\beta}_B^M(i) \quad (\text{V.28})$$

où $\hat{\beta}_B^M(i)$ est la réponse mesurée à une *commande de base* (ou commande élémentaire).

L'objectif devient alors de trouver la réponse forcée en fonction d'une commande du type $\mu(n)^T \hat{\beta}_B^M(i)$. Mais, la commande à chaque instant $n + i$ dans le futur est une combinaison linéaire

de i^{k-1} commandes de base :

$$\beta_{[n+i]}^C = \sum_{k=1}^{n_B} \mu_k(n) \beta_{Bk}^C(i), \quad n_B = i^{k-1} \quad (\text{V.29})$$

Reprenant l'écriture matricielle du modèle de l'actionneur, nous obtenons :

$$\begin{aligned} \vec{X}_{[n+i]} &= \mathbf{A} \vec{X}_{[n+i-1]} + \vec{B} \beta_{[n+i-1]}^C = \\ &= \mathbf{A}^2 \vec{X}_{[n+i-2]} + \mathbf{A} \vec{B} \beta_{[n+i-2]}^C + \vec{B} \beta_{[n+i-1]}^C \end{aligned} \quad (\text{V.30})$$

$$Y_{[n+i]} = \hat{\beta}_B^M(i) = \left[\sum_{j=0}^i \vec{C}^T \mathbf{A}^{i-j} \vec{B} \beta_{B1}^C[n+j-1] \dots \sum_{j=0}^i \vec{C}^T \mathbf{A}^{i-j} \vec{B} \beta_{Bn_B}^C[n+j-1] \right]^T \quad (\text{V.31})$$

Un algorithme de minimisation est alors utilisé pour trouver $\mu(n)$ les coefficients de la combinaison linéaire à l'instant présent n (ou $i = 0$) qui minimise un critère sur l'horizon de prédiction. Le critère à minimiser exprimé sous forme d'une distance sera noté avec $D(\mu(n))$:

$$D(\mu(n)) = \sum_{i=0}^{n_H} \left(\hat{\beta}_{[n+i]}^M - \beta_{[n+i]}^{trajRef} \right)^2 = \quad (\text{V.32})$$

$$= \sum_{i=0}^{n_H} \left(\vec{C}^T \mathbf{A} \vec{X}_{[n]} + \mu(n)^T \hat{\beta}_B^M(i) - \beta_{[n+i]}^{trajRef} \right)^2 = \quad (\text{V.33})$$

$$= \sum_{i=0}^{n_H} \left(\mu(n)^T \hat{\beta}_B^M(i) - d(n+i) \right)^2 \quad (\text{V.34})$$

où $d(n+i) = \beta_{[n+i]}^{trajRef} - \vec{C}^T \mathbf{A} \vec{X}_{[n]}$

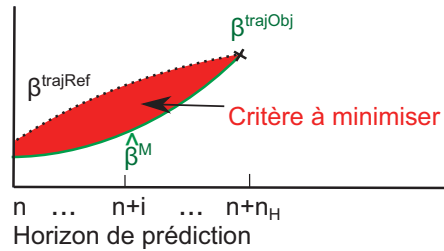


FIGURE V.30 – Connaissant la consigne de braquage β^{ref} correspondant à un suivi idéal de la trajectoire de référence et connaissant également l'erreur de position du véhicule à l'instant i , il est possible d'estimer une séquence de consignes β^* qui minimise un critère sur l'horizon de prédiction $[i, i + H]$

Comme nous cherchons les coefficients $\mu_k(n)$ qui minimisent $D(\mu(n))$, alors en dérivant $D(\mu(n))$ par rapport aux coefficients et en sachant que à l'extrême la dérivée est égale à zéro, alors nous obtenons :

$$D'(\mu(n)) = \sum_{i=0}^{n_H} \left(\mu^*(n)^T \hat{\beta}_B^M(i) - d(n+i) \right) \hat{\beta}_B^M(i) = \quad (\text{V.35})$$

$$= \left[\sum_{i=0}^{n_H} \left(\hat{\beta}_B^M(i) \hat{\beta}_B^M(i)^T \right) \right] \mu^*(n) - \sum_{i=0}^{n_H} \left[d(n+i) \hat{\beta}_B^M(i) \right] = \quad (\text{V.36})$$

$$= R_2 \mu^*(n) - R_1 = 0 \quad (\text{V.37})$$

Cela revient à réécrire

$$\mu^*(n) = \min_{\mu(n)} [D(\mu(n))] = \frac{R_1}{R_2} \quad (\text{V.38})$$

avec :

$$R_1 = \sum_{i=0}^{n_H} [d(n+i) \hat{\beta}_B^M(i)] \quad (\text{V.39})$$

$$R_2 = \sum_{i=0}^{n_H} (\hat{\beta}_B^M(i) \hat{\beta}_B^M(i)^T) \quad (\text{V.40})$$

Comme la structure de la commande s'exprime par définition sous la forme :

$$\beta_{[n+i]}^C = \sum_{k=1}^{n_B} \mu_k(n) \beta_{Bk}^C(i), \quad n_B = i^{k-1} \quad (\text{V.41})$$

alors le terme de l'asservissement à l'instant présent n ($i = 0$) s'écrit donc :

$$\beta^{*traj} = \beta_{[n]}^C = \mu^*(n) \beta_B^C(0) \quad (\text{V.42})$$

Reprenant l'équation (V.20) nous trouvons la forme finale de l'angle de braquage avec tous les variables connues :

$$\beta = \beta^{*traj} + \beta^{déviation} \quad (\text{V.43})$$

Comme nous l'avons déjà présenté, l'angle de la roue moyenne directrice du véhicule β est caractérisé par une composante liée à la trajectoire β^{*traj} que nous estimons utilisant l'approche de commande prédictive par modèle et une deuxième composante appelée déviation par rapport à la trajectoire $\beta^{déviation}$ que nous avons estimée dans les équations (V.23) et (V.24).

Par la suite nous présentons des résultats préliminaires obtenus utilisant cette méthode pour le suivi de trois trajectoires différentes.

V.5.3 Résultats du suivi de trajectoire

Implémentant le principe de la commande PFC, nous réalisons alors des tests préliminaires sur une plateforme de simulation qui reproduit le comportement des actionneurs et de manière globale du véhicule à basse vitesse.

Pour une première validation des résultats nous utilisons la trajectoire présentée dans la figure V.31(a). Les positions de départ et d'arrivée sont illustrées par des rectangles et le sens de suivi de la trajectoire par des flèches. Pour les tests nous avons imposé au véhicule de suivre la trajectoire à une vitesse constante. Le choix de la vitesse maximale a été réalisé en fonction des performances du système de perception des obstacles.

Les indicateurs de qualité que nous suivons sont les erreurs latérales et angulaires du véhicule par rapport à la trajectoire tout au long du parcours. Dans ces sens, les figures V.31(b) et (c) présentent les résultats obtenus. Nous pouvons constater que l'erreur latérale reste dans une enveloppe de $\pm 0,1$ m à l'exception d'un changement de trajectoire au dernier moment lors de l'évitement de l'obstacle (en rouge dans la figure V.31(a)) où l'écart latéral atteint un pic de 0,12 m. Dans les mêmes conditions l'écart angulaire atteint un pic de 9° .

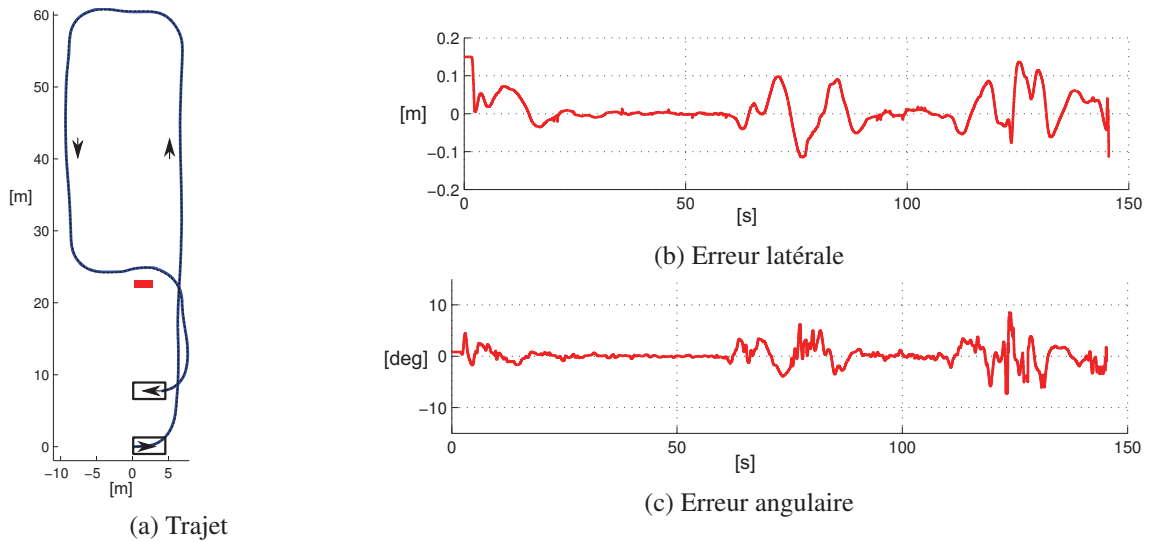


FIGURE V.31 – Contrôle/commande pour le suivi de trajectoire de référence d'un véhicule qui roule à une vitesse de 3m/s. Dans la figure (a) la trajectoire de référence est représentée par une courbe noire pointillée et trajectoire réalisée par le véhicule par une courbe en bleu continu. Le rectangle rouge représente l'obstacle évité et les rectangles noirs les deux places de parking de début et de fin de la trajectoire. Les figures (b) et (c) illustrent les erreurs latérales respectivement angulaires entre la position du véhicule et la trajectoire à suivre

Nous répétons les mêmes tests pour deux autres trajectoires présentées dans les figures V.32(a) et V.33(a) et nous obtenons des erreurs de même ordre de grandeur.

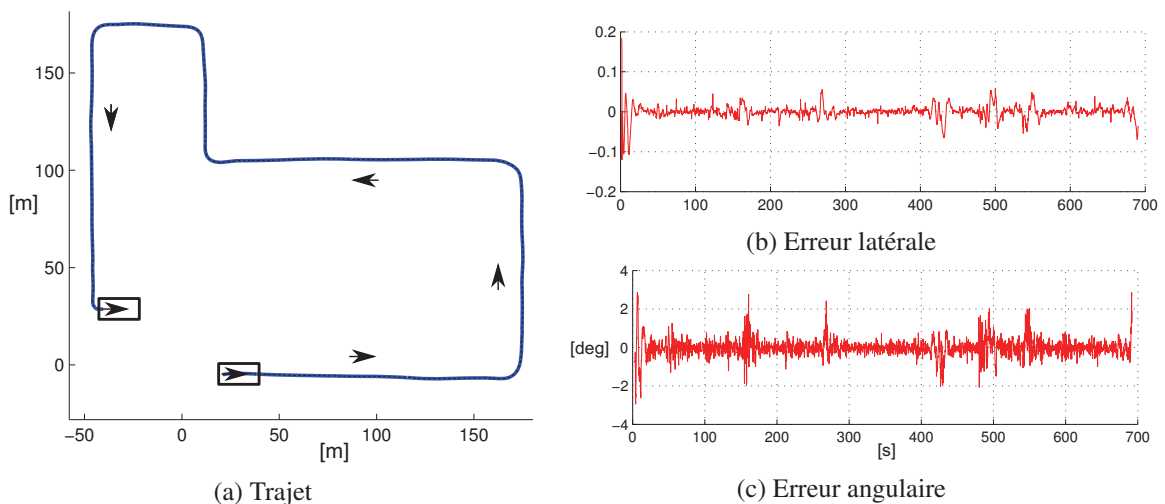


FIGURE V.32 – Contrôle/commande pour le suivi de trajectoire de référence d'un véhicule qui roule à une vitesse de 1,5m/s

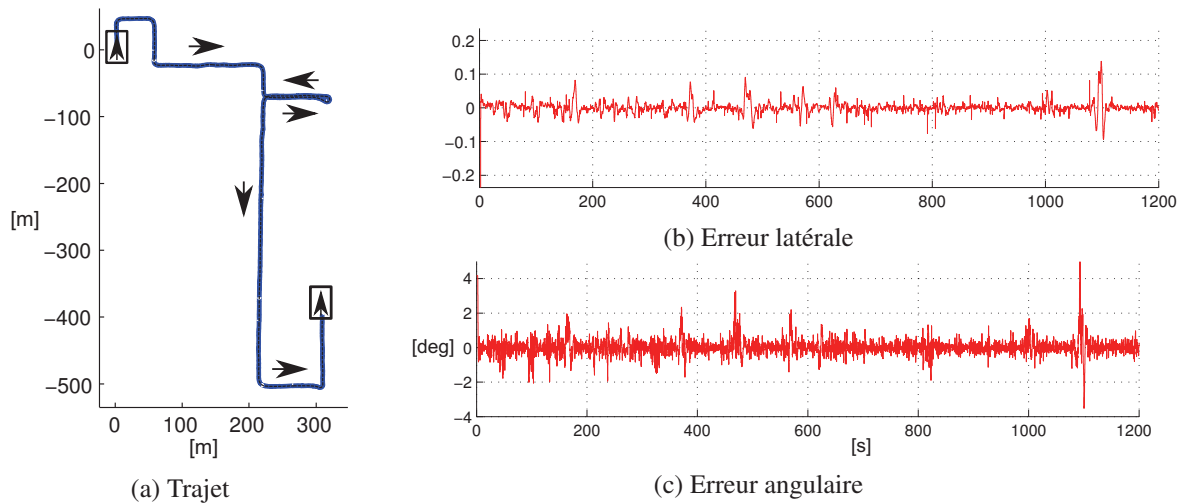


FIGURE V.33 – Contrôle/commande pour le suivi de trajectoire de référence d'un véhicule qui roule à une vitesse de 1,5m/s

Comme nous l'avons déjà exprimé, même si le contrôle des actionneurs ne fait pas l'objet du travail de recherche, nous nous sommes intéressés à ce sujet pour avoir une vision d'ensemble des problématiques liées au véhicule autonome.

Nous avons montré qu'une approche du type *commande prédictive à modèle* (PFC - Predictive Functional Control) peut répondre de manière satisfaisante à notre problème de suivi de trajectoire. Nous avons montré également qu'il est possible d'obtenir des erreurs latérales inférieures à 0,1 m pour un déplacement du véhicule à basse vitesse (1,5 m/s maximum dans notre cas) sur des trajectoires présentant des virages serrés mais dans des bonnes conditions d'adhérence. Même si dans notre modèle de déplacement du véhicule nous n'avons pas intégré les dérives dues au roulement avec glissement, il existe la possibilité ou de réaliser une estimation de ces dérives ou de les mesurer utilisant un capteur du type centrale inertielle, capteur qui n'était pas disponible sur notre véhicule prototype.

Par la suite nous présentons le bilan de ce chapitre.

V.6 Bilan sur la navigation

Dans ce chapitre nous avons présenté l'état de la problématique de navigation d'un véhicule liée à notre système de voiturier autonome. Comme le véhicule n'a pas le droit d'explorer des zones inconnues le problème de la planification globale n'apparaît pas. Dans notre cas il s'agit plutôt de la navigation dans un couloir connu depuis une étape d'apprentissage et de l'évitement des obstacles non cartographiés situés sur la trajectoire de référence.

Nous avons vu que lors de l'évitement d'un obstacle, le début du mouvement du véhicule doit l'amener à une configuration où ses capteurs pourront décider de la longueur de l'obstacle et de la suite du mouvement. Nous avons montré que grâce à l'utilisation d'une carte d'obstacles nous pouvons prédire suffisamment à l'avance de la possibilité d'emprunter un certain couloir

de navigation. Pour la construction d'une partie de trajectoire nous avons utilisé des courbes à géométrie connue. Nous avons montré qu'en utilisant trois clothoïdes et éventuellement deux arcs de cercle (si le braquage maximal est atteint) il est possible de créer des trajectoires admissibles à courbure continue adaptées à notre situation et compatibles avec la vitesse d'avancement et les temps de réponse des actionneurs.

Enfin concernant la commande des actionneurs, nous avons exprimé notre vision que le véhicule autonome peut être vu comme un robot commandable qui suit une trajectoire et un profil de vitesse et que toute l'intelligence est mise dans la création continue d'une trajectoire adaptée à la situation. Abordant le problème ainsi nous a permis d'utiliser directement une approche d'estimation de la commande pour un horizon donné. Les avantages de la méthode utilisée sont multiples. D'abord, nous avons pris en compte de manière explicite la réponse des actionneurs et nous avons utilisé un critère à minimiser pour construire une commande optimale. Puis cette méthode nous donne aussi une certaine flexibilité dans le choix de la réponse souhaitée, de l'horizon de prédiction ou bien de la structure de la commande. Mais il n'y a pas que des avantages et l'approche présente aussi des limitations. D'abord, toute modélisation de l'actionneur est inexacte et cela se traduit dans une réponse différente du système. Enfin, cette inexactitude est transférée dans le critère de minimisation et il est difficile d'apporter une preuve formelle de stabilité du système en toute condition.

Par la suite nous présentons un bilan général.

Bilan général, questions ouvertes, conclusions et perspectives

Bilan général

Dans ce travail de thèse nous avons étudié des algorithmes de perception, de localisation et de contrôle d'un véhicule autonome se déplaçant à basse vitesse pour une application du type voiturier autonome à domicile (ou *autonomous home valet parking*, en anglais).

Après avoir posé le problème et présenté les avancées techniques sur le véhicule autonome avec un focus sur le parking autonome, nous nous sommes concentrés sur les principales fonctions d'un système d'automatisation de conduite : la détection des obstacles, la localisation et la navigation. Partant de l'existant et connaissant les performances attendues, une architecture système ainsi que la logique de fonctionnement du voiturier autonome ont été tracées. Cela s'est concrétisé dans la création d'une architecture fonctionnelle (constituée d'un ensemble de fonctions interconnectées, de scénarios et de modes opératoires), dans la création d'une architecture software modulaire ainsi que dans la création des interfaces pour la connexion au véhicule prototype.

Nous avons exploré la problématique de la détection d'obstacles, les moyens de perception et de l'acquisition de l'information. Nous nous sommes attardés pour réaliser un bilan des capteurs de distance tout en justifiant les avantages et les limitations de nos choix d'utiliser un système de 12 capteurs ultrason (SCU) et d'une caméra monoculaire avec objectif fish-eye. Partant d'un système propriétaire fermé de capteurs ultrason, nous avons réussi à réaliser une carte d'obstacle à un niveau de précision supérieur au produit d'origine. Une augmentation de la limite de détection des capteurs ultrason a été réalisée utilisant une détection d'obstacle via une technique *Structure from Motion* tout en gardant l'aspect temps réel. Suivant les performances obtenues et nous avons conclu que l'utilisation duale caméra - SCU permet d'un côté de pallier le problème de non-détection des obstacles non texturés (comme le mur blanc) et en même temps d'avoir un horizon de perception supérieur à celui des capteurs ultrason. Enfin, ces informations d'occupation ont été exploitées pour traiter la problématique de détection du couloir de navigation libre dans la proximité de la trajectoire de référence. L'information de la carte a été stockée dans un arbre permettant d'économiser de la mémoire et de créer de cartes multirésolutions.

Concernant la localisation du véhicule, nous nous sommes focalisés sur les techniques adaptées à notre application. Sur des fils d'exécution indépendants les techniques de localisation en temps réel collaborent pour une robustesse dans le fonctionnement continu : la localisation odométrique, la localisation par appariement des grilles d'occupation et la localisation par appariement entre une image actuelle et une base d'images adaptée à notre besoin et améliorée en termes de temps de calcul. Dans les exemples présentés nous avons conclu que le système de localisation par appariement de grilles d'occupation est capable de se localiser de plus en plus précisément avec l'enrichissement de la grille. Nous avons conclu également qu'il est possible d'intégrer des a priori

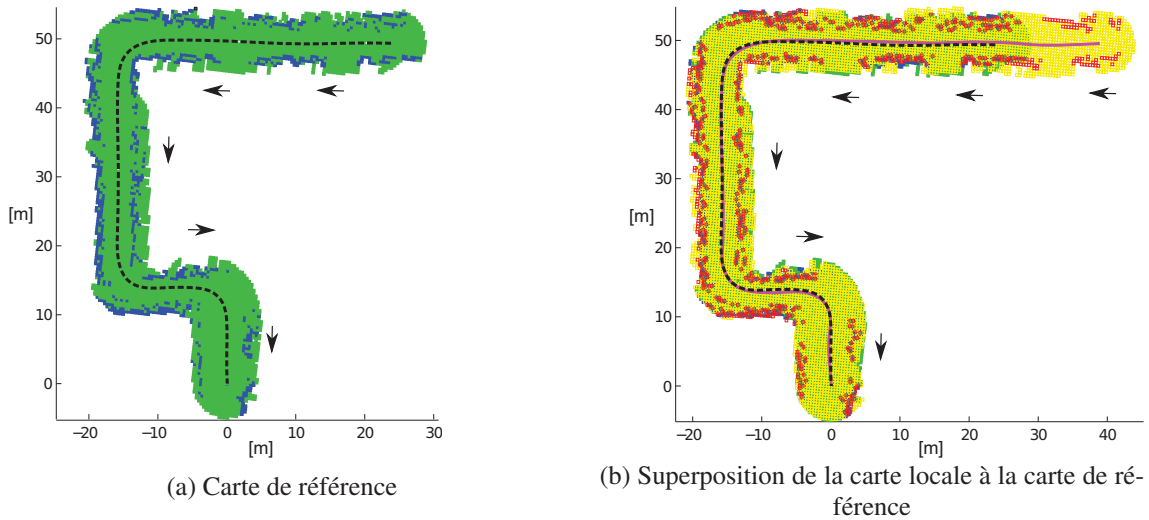
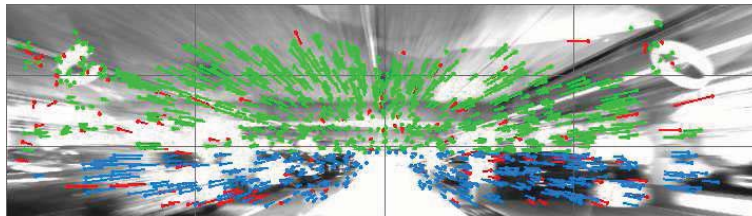
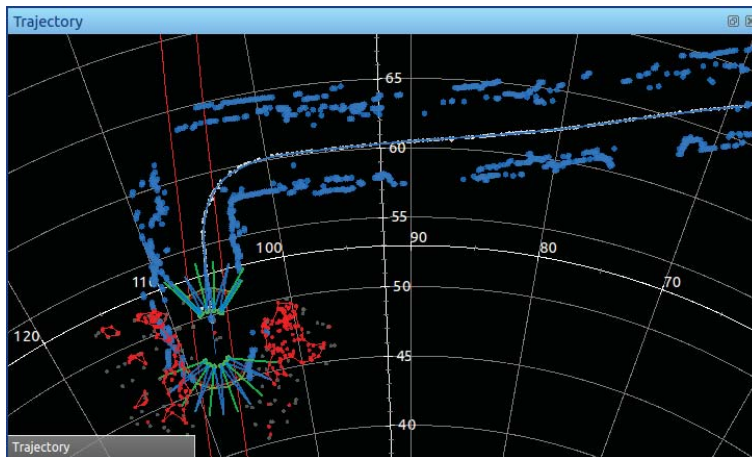


FIGURE V.34 – Cartes d'occupation



(a) Appariements des points intéressés dans des images successives dans une approche SfM pour la détection des obstacles

(b) Calcul des obstacles par approche SfM dans un rayon de 8 m autour de la caméra. Les points rouges correspondant aux obstacles ont été accumulés par une technique de *fenêtre glissante* au long de la trajectoireFIGURE V.35 – Exemple d'extension de la zone observée par les capteurs ultrasons par une approche *Structure from Motion*

de localisation diminuant ainsi l'incertitude et augmentant la vitesse de convergence. Ces a priori de localisation sont donnés par le résultat de la méthode de localisation globale par appariement d'images. Nous avons montré qu'il existe des situations où la forte ressemblance de l'environnement perturbe cette méthode et que le résultat présente des *sauts*. Ces inconvénients ont été réglés par l'ajout d'un modèle de déplacement qui a pour rôle de filtrer les données aberrantes et

d'estimer la nouvelle position suivant l'historique du déplacement.

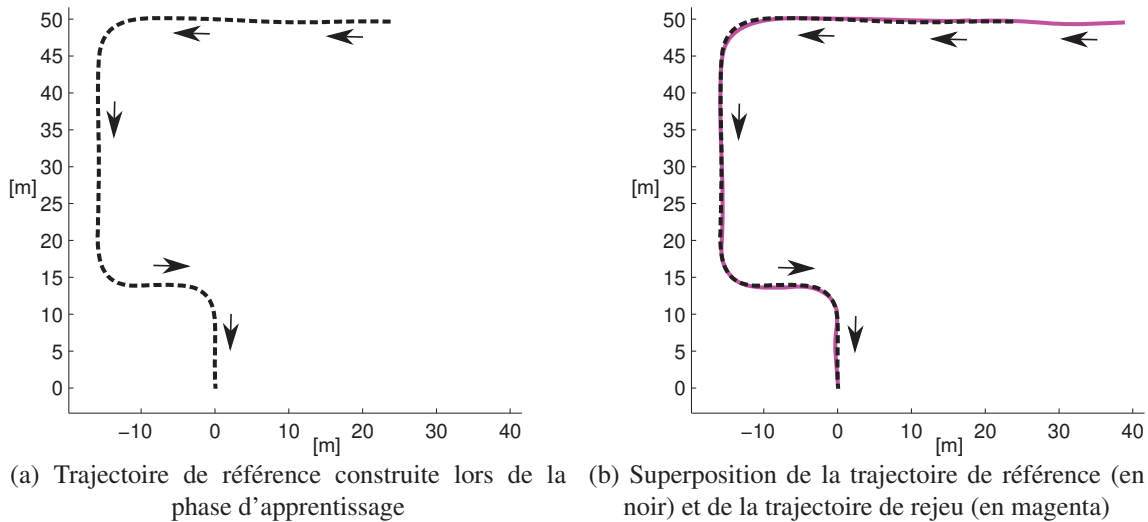


FIGURE V.36

Enfin, nous avons dédié le dernier chapitre du mémoire à la problématique de navigation du véhicule. Nous avons considéré résolue la problématique de contrôle des actionneurs du véhicule pour le suivi d'une trajectoire donnée et nous nous sommes concentrés sur la création d'une trajectoire admissible. Utilisant les fonctionnalités de gestion du couloir traversable et de localisation construites dans les chapitres II à IV, nous avons développé une technique de planification locale pour le guidage du véhicule lors de la présence d'obstacles non cartographiés sur la trajectoire habituelle. Pour la construction d'une partie de trajectoire nous avons utilisé des courbes à géométrie connue et nous avons montré qu'en utilisant trois clothoïdes et éventuellement deux arcs de cercle (si le braquage maximal est atteint) il est possible de créer des trajectoires à courbure continue adaptées à notre situation. Nous avons montré que l'utilisation d'une carte d'obstacles nous permet de prédire plus en avance de la possibilité d'emprunter un certain couloir de navigation.

Au final, il y reste des questions ouvertes auxquelles il faudrait répondre pour pouvoir atteindre un niveau de 5 sur 5 d'autonomie de conduite du système de conduite autonome.

Questions ouvertes

- Quel est le niveau de surveillance humaine ?
Concernant le système de voiturier autonome, les défis techniques sont de différents niveaux de complexité et dépendent fortement du degré de surveillance humaine. Même si l'application cible est de n'avoir ni intervention ni surveillance humaine, il faudrait passer par des étapes intermédiaires car le manque de surveillance du véhicule lors de la manœuvre dans un environnement ouvert (accessible aux personnes) rend le problème comparable à la conduite urbaine à faible vitesse. Par conséquent, dans cette étape de transition il faudrait définir quel est le rôle de l'humain ou il faudrait décider et assumer la responsabilité en tant que constructeur si le véhicule est le seul responsable de sa sécurité. Sinon il faudrait que le conducteur dispose d'un bouton d'arrêt d'urgence et qu'il reste à proximité du véhicule ou qu'il le surveille en continue depuis un dispositif connecté.
- Est-ce que la validité du scénario est laissée à l'appréciation du conducteur ? Si oui, est-ce qu'on peut être sûr que le conducteur a bien compris les limites du système ?

Nous avons donné le cadre de l'utilisation du voiturier autonome mais il faudrait s'interroger également sur la manière de vérification que le scénario d'utilisation est conforme aux limites techniques du système avant la mise en conduite autonome ;

- Si dans des conditions d'utilisation nominales la coexistence avec d'autres éléments dynamiques est acceptée, alors est-ce qu'il y a des limites de taille et des vitesses à respecter par les autres objets/êtres ? Par exemple les enfants et les animaux domestiques peuvent surgir devant le véhicule.
- Est-ce qu'il y a des limites géométriques pour l'espace de stationnement ou pour la zone de navigation ? Est-ce que ces limites peuvent être quantifiées et un diagnostic peut être réalisé automatiquement ?
- Quelles sont les conditions nominales d'éclairage et les conditions météorologiques ? Qu'est-ce qu'il faudrait faire en cas de non-respect de ces contraintes ? Est-ce qu'il est suffisant de prendre toutes les décisions en fonction de l'instant où le conducteur demande le démarrage en autonome du véhicule ?
- Quelles sont les signalisations et comment délimiter la zone de navigation ? Comment gérer le problème des obstacles réglementaires ?
- Est-ce qu'il est nécessaire/souhaitable d'utiliser une communication I2V ?
- Combien d'heures de tests sont nécessaires pour démontrer le bon fonctionnement du système dans sa globalité ou au moins des fonctions sécuritaires ? Est-ce qu'il est possible de définir un nombre de combinaisons de conditions que véhicule est susceptible de rencontrer de telle façon qu'on puisse décider de la représentativité et de la suffisance des tests ?

Enfin les dernières questions concernent les objectifs de sécurité à atteindre par le système conformément aux normes SdF (Sûreté de Fonctionnement). Malheureusement, l'histoire montre que tous les systèmes (semi)autonomes sont inévitablement impliqués, tôt ou tard, dans des accidents soldés avec des blessés ou morts. Ainsi les normes SdF cherchent à encadrer et à assurer une certaine probabilité d'occurrence des accidents. Voici quelques questions auxquelles il faudrait répondre :

- Quelle est la fréquence acceptable de perte de contrôle du véhicule ?
- Quel est le niveau de verrouillage et comment limiter les mouvements ou la vitesse de véhicules (mécanique, électronique, logiciel) ?
- Quel est le niveau de protection contre les cyber-attaques ? Comment on limite la prise de contrôle à distance ou l'inhibition des fonctions de sécurité ?
- Quelle est la probabilité d'occurrence que :
 - le véhicule soit impliqué dans un accident et de blesser/tuer un piéton adulte, un enfant ou un animal de compagnie ;
 - le véhicule tombe en panne et qu'il crée un blocage d'accès pour d'autres véhicules et/ou piétons ; ou bien que la reprise en manuel ne puisse pas se réaliser suite à la mise en hors-service du système de conduite autonome ;

Nous terminons notre mémoire en présentant nos conclusions et les perspectives de travail.

Conclusions et perspectives

Dans ce travail nous avons montré qu'il est possible de concevoir un système de parking automatique beaucoup plus performant et polyvalent que ceux actuellement disponibles qui ne permettent que le parking latéral. Tout cela en respectant la contrainte imposée sur la définition technique des capteurs extéroceptifs limités en prix et en gabarit. Même si le contexte dans lequel

le véhicule va évoluer est moins contraignant que la conduite sur route ouverte, toutefois le sujet est complexe par la diversité de l'environnement et des situations que le système devrait gérer.

Même si nous avons prouvé de la faisabilité du projet, il y reste beaucoup de travail à réaliser pour atteindre des niveaux de performances et la robustesse qui rendront l'application prête à la commercialisation. Plusieurs aspects techniques ou applicatifs restent à améliorer ou à développer :

- Détection en temps réel d'une défaillance partielle ou totale d'un sous-système du véhicule et action en conséquence pour assurer la sécurité du véhicule ;
- Amélioration du fonctionnement du système de perception pour la localisation et la détection d'obstacles dans une gamme plus large de conditions de luminosité y compris dans des conditions d'éclairage nocturne grâce aux feux du véhicule ;
- Conduite par temps de légère pluie et décision automatique de la mise en marche ou non du système autonome ;
- Gestion des obstacles conventionnels : pelouse, signes de circulation (s'il s'agit de la conduite dans une copropriété où il peut y avoir du trafic) ;
- Alerte du propriétaire par communication à distance en cas de non-aboutissement de la manœuvre ou en cas de présence d'un obstacle bloquant ;
- Élargissement de l'utilisation également à des parkings fermés : gestion de la conduite sur les rampes incurvées entre les niveaux, développement d'un système de détection et discrimination d'une place de parking disponible ;
- Gestion adaptative de la vitesse du véhicule par rapport à la précision de localisation ou par rapport aux conditions de détection d'obstacles
- Prise en compte du niveau d'adhérence entre les pneus et le sol ;
- Adaptation des paramètres des actionneurs ;
- Exploration des capteurs communicants en dehors du véhicule.

Nous concluons notre mémoire en affirmant qu'aujourd'hui, la commercialisation de voitures 100% autonomes n'est pas à l'ordre du jour et reste un projet d'avenir. Une forte incitation à acheter des véhicules équipés des ADAS vient aussi de la part de certains assureurs qui offrent des réductions sur la prime d'assurance des voitures semi-autonomes afin d'anticiper et d'encourager l'évolution technologique. Ainsi nous trouvons des offres allant jusqu'à 25% de réduction de la prime d'assurance pour les véhicules équipés des trois technologies suivantes : système de freinage d'urgence autonome, dispositif de stationnement automatique (capable de gérer la direction, l'accélération et le freinage) et enfin régulateur de vitesse adaptatif avec fonction automatisée de freinage. Nous constatons que la fonction de parking autonome n'est pas négligée car pour les assureurs le coût total des dégâts réalisés lors des manœuvres de parking est très élevé.

Mais, au-delà des théories, la technologie répond encore mal aux attentes de l'industrie et aux souhaits des consommateurs. En divisant le problème global pour se concentrer uniquement sur le cas d'usage du parking permet d'avancer sur la recherche des solutions pratiques, implémentables et fiables de demain. De plus, commencer à commercialiser une voiture partiellement autonome permettra aux organismes législatifs d'avoir les bases des discussions sur les problèmes de responsabilité juridiques en cas d'accident, sur les problèmes de piratage informatique ou, le plus important, permettra aux gens de s'habituer à laisser le contrôle de la voiture à des systèmes informatiques.

Annexes

Annexe A.1

DBSCAN - Algorithme de partitionnement de données

L'algorithme DBSCAN¹ est proposé par Ester et al. [1996] et a pour but de partitionner des ensembles de données sur une logique de proximité des éléments constituant pour un critère de distance donnée. De plus cet algorithme se veut résistant au bruit et avec une capacité à le détecter. Le principe de fonctionnement est détaillé dans l'algorithme 4.

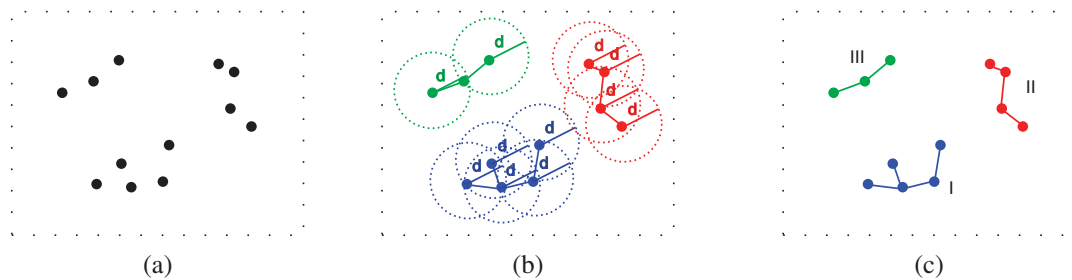


FIGURE A.1 – Résultat de l'algorithme 4 pour une distance de voisinage d et $min_{Elem} = 1$

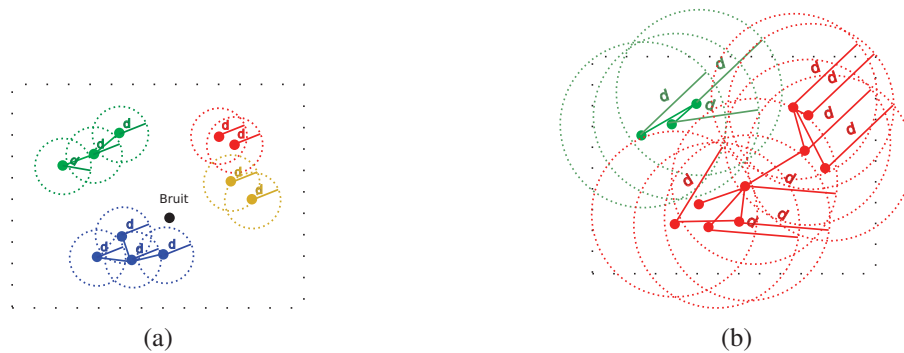


FIGURE A.2 – Résultat de l'algorithme DBSCAN pour différentes distance de voisinage d

1. Density-Based Spatial Clustering of Applications with Noise

Algorithme 4 DBSCAN

nécessite : E - Ensemble d'éléments; min_{Elem} - Taille minimale d'une classe;

nécessite : d - Distance maximale entre éléments

$ClasseCourante \leftarrow 1$

tant que il existe des éléments non visités dans E **faire**

 Choisir un élément X

$X.visité \leftarrow \text{OUI}$

$X.classe \leftarrow ClasseCourante$

$listeVoisins = \text{RECHERCHE_VOISINS}(X, E, d)$

si $tailleDe(listeVoisins) < min_{Elem}$ **alors**

pour tout X dans $listeVoisins$ **faire**

$X.visité \leftarrow \text{OUI}$

$X.classe \leftarrow \text{Bruit}$

fin pour

sinon

$ClasseCourante \leftarrow ClasseCourante + 1$

$\text{ETENDRE_VOISINAGE}(X, ClasseCourante, E, listeVoisins, d, min_{Elem})$

fin si

fin tant que

fonction $\text{ETENDRE_VOISINAGE}(X, ClasseCourante, E, listeVoisins, d, min_{Elem})$

$X.visité \leftarrow \text{OUI}$

$X.classe \leftarrow ClasseCourante$

pour tout X' de $listeVoisins$ **faire**

si $X'.visité == \text{NON}$ **alors**

$X'.visité \leftarrow \text{OUI}$

$listeVoisins' = \text{RECHERCHE_VOISINS}(E, X', d)$

si $tailleDe(listeVoisins') \geq min_{Elem}$ **alors**

$listeVoisins = listeVoisins \cup listeVoisins'$

fin si

fin si

si $X'.classe == 0$ **alors**

$X'.classe \leftarrow ClasseCourante$

fin si

fin pour

fin fonction

fonction $\text{RECHERCHE_VOISINS}(X, E, d)$

retourne les éléments de l'ensemble E qui sont à une distance inférieure à d de X

fin fonction

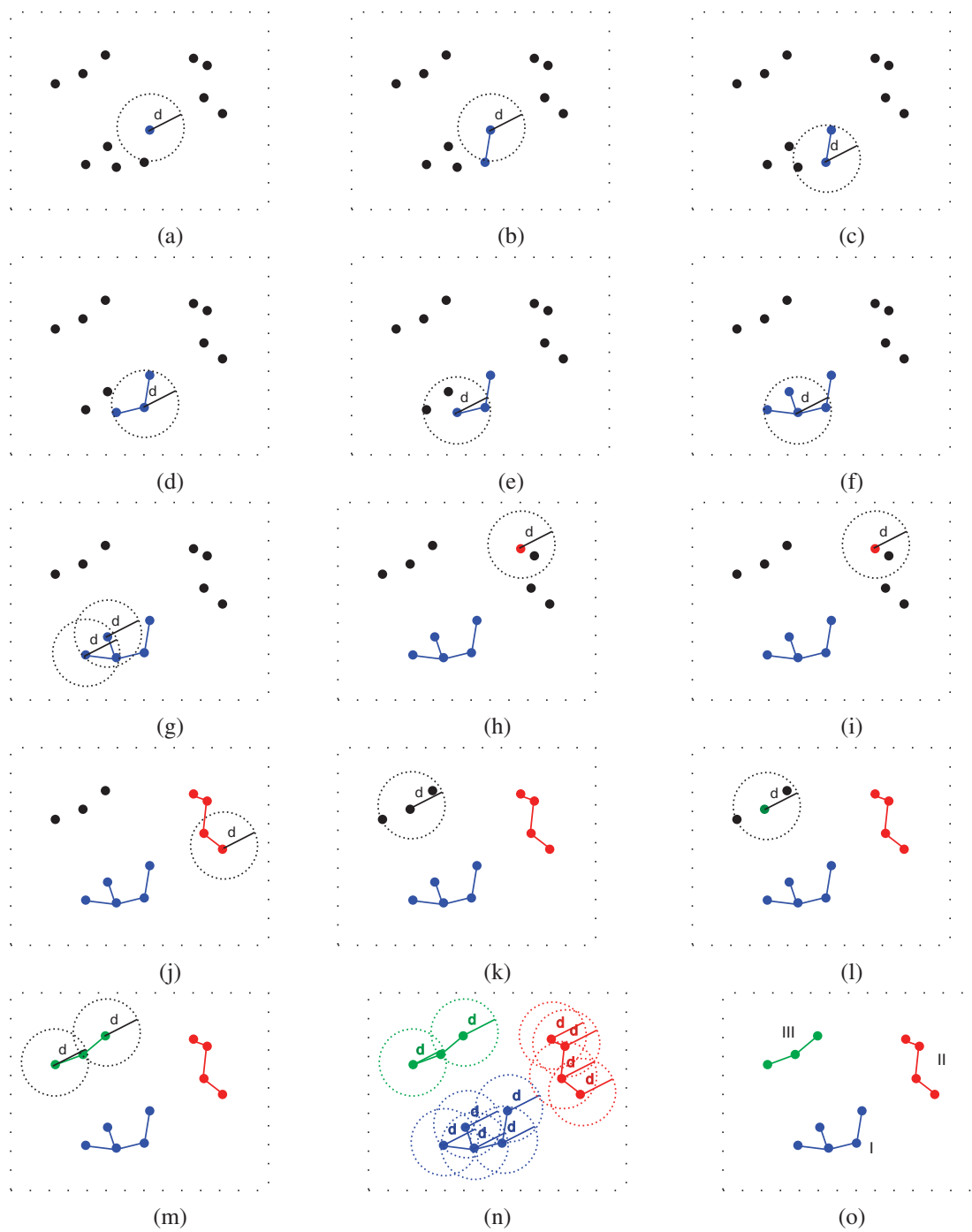


FIGURE A.3 – Détail de fonctionnement de l'algorithme DBSCAN pour une distance de voisinage d et un paramétrage $min_{Elem} = 1$

Annexe A.2

 χ -Shape - Algorithme pour décèlement de formes

L'algorithme *Algorithme χ Shape* [Duckham et al., 2008] pour la construction des polygones englobants (potentiellement non convexes) pour un ensemble des points donnés.

Algorithme 5 Algorithme χ - Shape

nécessite : E Ensemble de points $P \subset \mathbb{R}^2$

nécessite : Paramètre d

$\Delta_{Delaunay} = \text{TRIANGULATION_DELAUNAY}(E)$

$S_{Bord} \leftarrow \text{SORT}(\text{EXTRACTION_CONTOUR}(\Delta_{Delaunay}), \text{'décroissant'})$

$Polygone \leftarrow \text{vide}$

tant que $\text{taille}(S) > 0$ **faire**

$S_{Bord} \leftarrow S_{Bord} - s$ où $s \leftarrow$ segment en haut de la liste

si ($\|s\| > d$) && (s est **régulier** dans $\Delta_{Delaunay}$) **alors**

$\Delta_{Delaunay} \leftarrow \Delta_{Delaunay} - s$: suppression de s du $\Delta_{Delaunay}$

$S_{Bord} \leftarrow S_{Bord} \cap$ **révélés** : rajout des deux côtés **révélés** par la suppression de s

$S_{Bord} \leftarrow \text{SORT}(S_{Bord}, \text{'décroissant'})$

sinon

$Polygone \leftarrow Polygone \cup s$: rajout du s dans le polygôme résultant

fin si

fin tant que

retourne $Polygone$

fonction $\text{EXTRACTION_CONTOUR}(\Delta_{Delaunay})$

retourne S_{Bord} : l'ensemble des segments au bord de la triangulation $\Delta_{Delaunay}$

fin fonction

fonction $\text{TRIANGULATION_DELAUNAY}(E)$

retourne $\Delta_{Delaunay}$: l'ensemble des triangles qui respectent les contraintes de la *Triangulation Delaunay*

fin fonction

La *triangulation de Delaunay* [Delaunay, 1934] représente une décomposition de l'enveloppe convexe dans un ensemble fini des triangles de tel façon qu'aucun point P ne soit pas à l'intérieur du cercle circonscrit d'un des triangles. Dans les figures A.4(a) à (c) on retrouve, pour un ensemble de points, une configuration de triangles (non unique) qui respecte bien ces contraintes. Dans l'algorithme 5 on nomme cet ensemble $\Delta_{Delaunay}$.

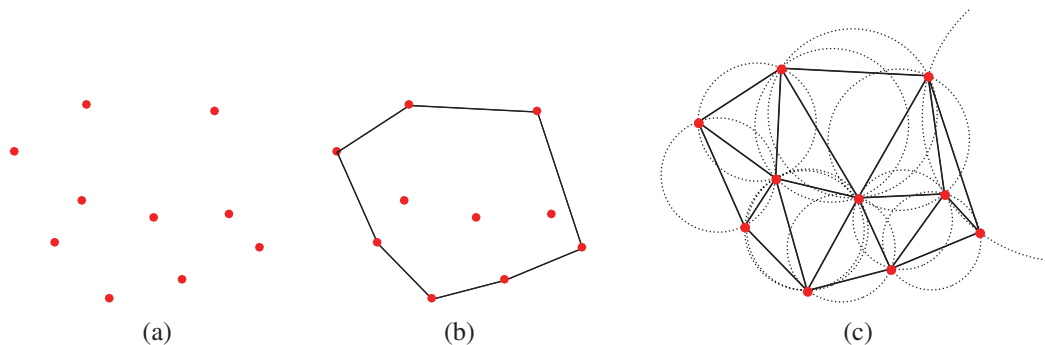


FIGURE A.4 – Résultat de la triangulation Delaunay

Dans la figure A.5 on retrouve une première étape de trie en ordre décroissante des tailles des bords de l'ensemble $\Delta_{Delaunay}$ et reconstruction du contour après l'enlèvement du segment le plus grand.

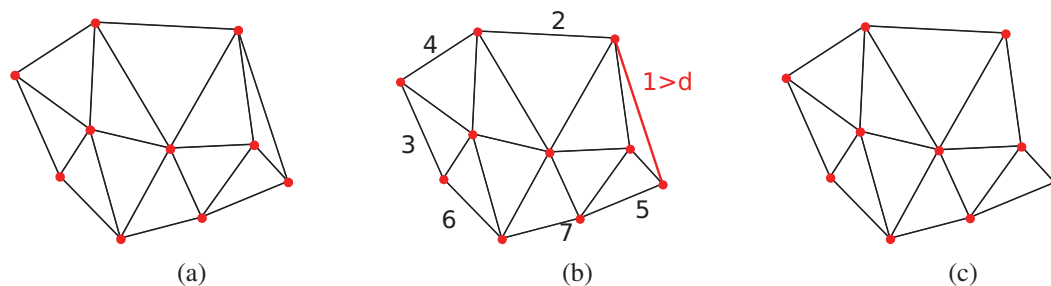


FIGURE A.5 – Algorithme χ Shape - élimination d'un segment situé sur le bord

Dans les figures A.6(a) à (h) on retrouve la succession des étapes de l'algorithme 5, en répétant le même raisonnement d'élimination des segments jusqu'à ce que tous les segments au bord de l'ensemble $\Delta_{Delaunay}$ soient d'une dimension supérieure à un d imposé.

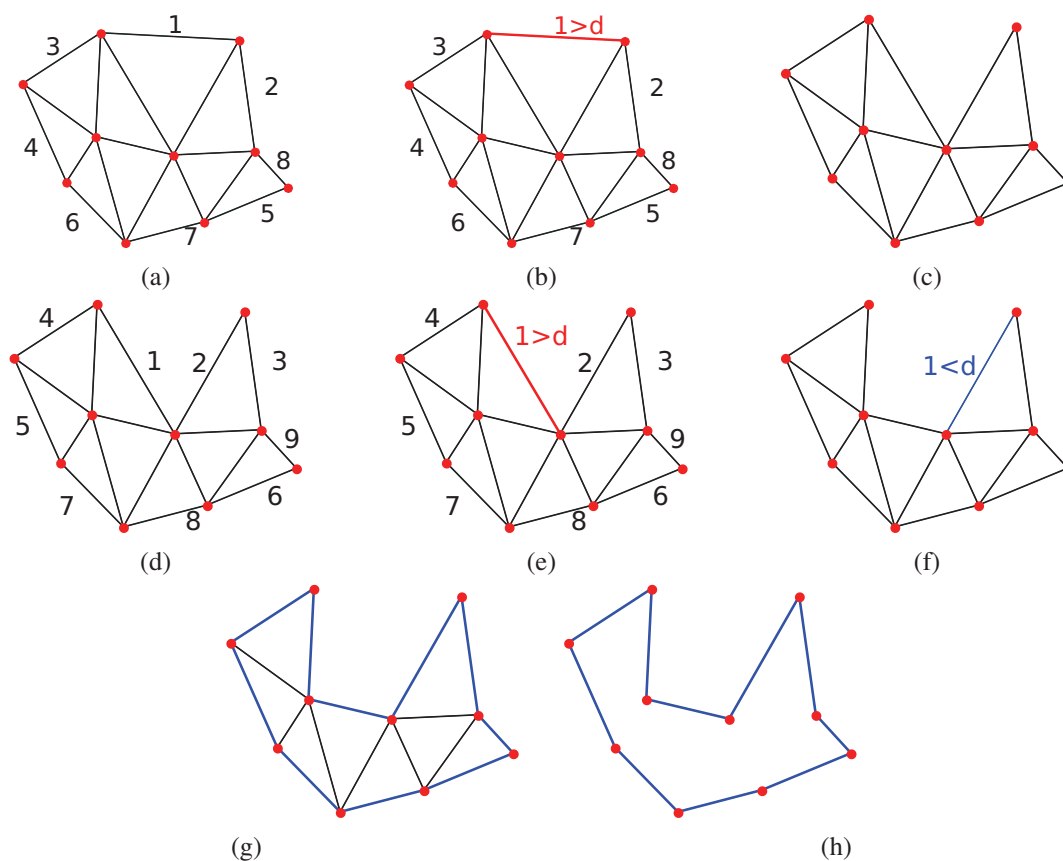


FIGURE A.6 – Algorithme χ – Shape - élimination d'un segment situé sur le bord

Annexe A.3

Comparaison des algorithmes *Bresenham* et *Raycast*

Dans les figures A.7(a) et (b) nous présentons le résultat de l'algorithme *Bresenham* et *Raycast* pour la sauvegarde d'un segment dans une grille d'occupation. On utilise les mêmes deux algorithmes pour réaliser la zone de visibilité de 360° autour d'un point et dans les figures A.8(a) et (b) on peut observer le résultat pour trois situations différentes.

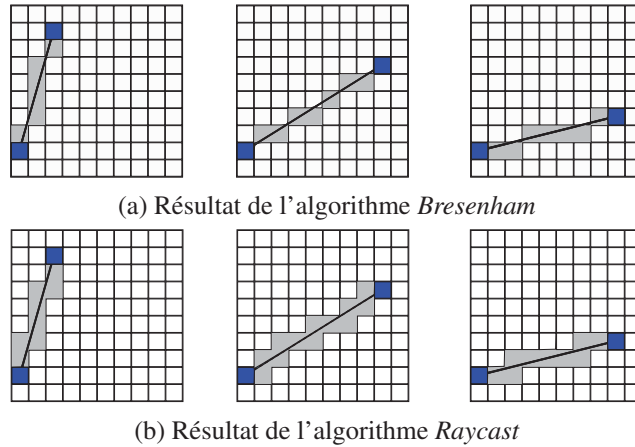


FIGURE A.7 – Comparaison des algorithmes *Bresenham* et *Raycast* pour trois segments

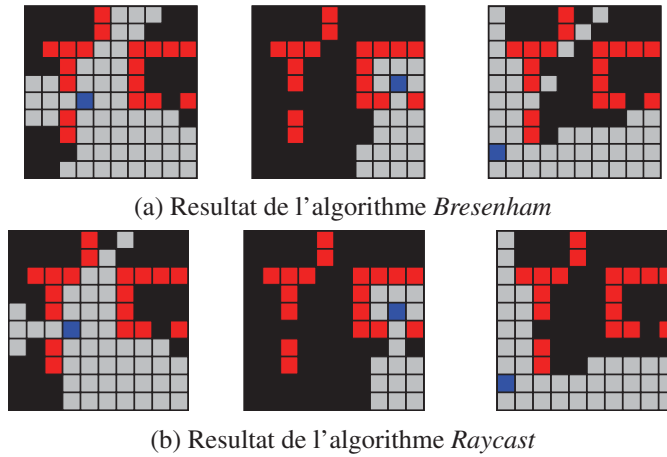


FIGURE A.8 – Comparaison des zones de visibilité de 360° autour d'un point pour trois situations différentes. En rouge - les obstacles, en noir - zone non observable, en gris - zone observable, en bleu - le point d'observation

Annexe A.4

Zone de recherche de la projection d'un point 3D dans l'image

[Féraud, 2011] développe une approche géométrique pour la définition d'une zone de recherche d'un point 3D $\vec{P}t_0 = [x_0, y_0, z_0]^T$ dans l'image. La zone est définie comme une enveloppe rectangulaire de la projection de l'ellipsoïde dans le plan de l'image. La méthode proposée extrait les points de l'ellipsoïde dont le plan tangent en ce point contient l'origine et conduit à des tangentes horizontales ou verticales dans l'image.

Pour la simplicité, notons $P_{\vec{P}_0}^{-1} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}$ est l'inverse de la matrice d'incertitude $\mathbf{P}_{\vec{P}_0}$ associé au point $\vec{P}t_0$. Les quatre points qui définissent la zone sont notés $\vec{P}_{vi} = [x_{vi}, y_{vi}, z_{vi}]^T$ et $\vec{P}_{hi} = [x_{hi}, y_{hi}, z_{hi}]^T$ pour $i \in \{1, 2\}$, les points conduisant à des tangentes verticales respectivement horizontales de l'enveloppe et seront calculés conformément à l'équation suivante :

$$\begin{cases} x_{vi} = \frac{-\beta + \sqrt{\Delta}}{p_0} \\ y_{vi} = -\frac{p_0}{p_1} x_{vi} - \frac{r}{p_1} \\ z_{vi} = z_0 - \frac{(x_{vi} - x_0)C + (y_{vi} - y_0)E}{F} \end{cases} \quad \text{et} \quad \begin{cases} x_{hi} = \frac{-\beta - \sqrt{\Delta}}{p_0} \\ y_{hi} = -\frac{p_0}{p_1} x_{hi} - \frac{r}{p_1} \\ z_{hi} = y_0 - \frac{(x_{hi} - x_0)B + (z_{hi} - z_0)E}{D} \end{cases} \quad (\text{A.1})$$

$$\text{où} \quad \begin{cases} p_0 = -[x_0, y_0] \vec{M}_1 \\ p_1 = -[x_0, y_0] \vec{M}_2 \\ r = -[x_0, y_0] \mathbf{M} [x_0, y_0]^T - 1 \end{cases} \quad \text{et} \quad \begin{cases} \alpha = \left[1, -\frac{p_0}{p_1} \right] \mathbf{M} \left[1, -\frac{p_0}{p_1} \right]^T \\ \beta = -2 \left(\frac{r}{p_1} \right) \left[1, -\frac{p_0}{p_1} \right] \vec{M}_2 \\ \gamma = M_{22} \left(\frac{r}{p_1} \right)^2 - r \\ \Delta = \beta^2 - 4\alpha\gamma \end{cases} \quad (\text{A.2})$$

avec \vec{M}_i est la $i^{\text{ème}}$ colonne de la matrice \mathbf{M} qui est définie dans le cas des tangentes verticales respectivement horizontales par :

$$\mathbf{M} = \frac{1}{F} \begin{bmatrix} AF - C^2 & BF - CE \\ BF - CE & DF - E^2 \end{bmatrix} \quad \text{ou} \quad \mathbf{M} = \frac{1}{D} \begin{bmatrix} AD - B^2 & CD - BE \\ BF - CE & DF - E^2 \end{bmatrix} \quad (\text{A.3})$$

Les points $\vec{P}_{v1}, \vec{P}_{v2}, \vec{P}_{h1}$ et \vec{P}_{h2} ainsi calculés, seront projetés dans l'image et notés p_{v1}, p_{v2}, p_{h1} et p_{h2} . Ces derniers points se situent sur l'ellipse d'incertitude et permettent de retrouver les limites de la zone rectangulaire englobant la projection de l'incertitude, comme illustré dans la figure III.12 de la section III.3.2.1.

Annexe A.5

Algorithmes d'appariements de grilles d'occupation

Algorithme 6 Calcul du coefficient d'appariement entre deux cartes pour déterminer la roto-translation nécessaire à appliquer à une carte pour se superposer à la deuxième

nécessite : Carte globale $CarteG$ et carte locale $CarteL$

tant que Système ultrason en actif **faire**

$CarteL \leftarrow$ mise à jour

$aprioriPosition : \{r_{min}, r_{max}, c_{min}, c_{max}, \theta_{min}, \theta_{max}\} \leftarrow$ mise à jour

pour tout $\theta_i \in [\theta_{min}, \theta_{max}]$ **faire**

$corrScore_{r \times c}(\theta_i) \leftarrow$ SCORE_CORRELATION($CarteG_{r \times c}, CarteL_{m \times n}, aprioriPosition$)

$maxCorrScore_i \leftarrow$ max($corrScore_{r \times c}(\theta_i)(:)$)

fin pour

$[\Delta_{theta}] \leftarrow$ Index (max($maxCorrScore_i(:)$))

$[\Delta_x, \Delta_y] \leftarrow$ Index (max($corrScore_{r \times c}(\theta_{Idx})(:)$))

fin tant que

fonction SCORE_CORRELATION($CarteG_{r \times c}, CarteL_{m \times n}, aprioriPosition$)

si premier appel **alors**

// Les variables globales suivantes sont définies :

$N \leftarrow m \times n$: nombre d'éléments de la $CarteL_{m \times n}$

$\overline{CarteL}_{1 \times 1} \leftarrow \left(\sum_{i,j} CarteL_{m \times n}(i,j) \right) \frac{1}{N}$

$stdL_{1 \times 1} \leftarrow std(CarteL_{m \times n}(:))$

fin si

$corrScore \leftarrow 0_{r \times c}$

$1_{m \times n} \leftarrow ones(m, n)$

$\overline{CarteG}_{r \times c} \leftarrow convolve(CarteG_{r \times c}, 1_{m \times n} \circ \frac{1}{N})$

$corrI_{r \times c} \leftarrow convolve(CarteG_{r \times c}, flipplr(flipud(CarteL_{m \times n} - \overline{CarteL}_{1 \times 1}))) \frac{1}{N}$

$corrII_{r \times c} \leftarrow \overline{CarteG}_{r \times c} * sum(CarteL(:) - \overline{CarteL}_{1 \times 1})_{1 \times 1}$

$stdG_{r \times c} \leftarrow sqrt\left(convolve\left(CarteG_{r \times c}^{\circ 2}, 1_{m \times n} \circ \frac{1}{N} \right) - Moy_{CarteG_{r \times c}}^{\circ 2} \right)_{r \times c}$

$corrScore \leftarrow (corrI_{r \times c} - corrII_{r \times c}) \circ \frac{1}{(stdG_{r \times c} \circ stdL_{1 \times 1})}$

retourne $corrScore$

fin fonction

fonction SCORE_CORRELATION_OPTIM(*CarteG_{r×c}*, *CarteL_{m×n}*, *aprioriPosition*)

// Fonction avec résultats qualitatifs semblables mais moins coûteuse en termes de temps de calcul

N ← $m \times n$: nombre d'éléments de la *CarteL_{m×n}*
 $corrScore$ ← $\mathbf{O}_{r \times c}$
 $1_{m \times n}$ ← $ones(m, n)$
 $\overline{CarteG_{r \times c}}$ ← $convolve\left(CarteG_{r \times c}, 1_{m \times n} \circ \frac{1}{N}\right)$
 $corrI_{r \times c}$ ← $convolve\left(CarteG_{r \times c}, fliplr\left(flipud\left(CarteL_{m \times n} - \overline{CarteL_{1 \times 1}}\right)\right)\right) \frac{1}{N}$
 $stdG_{r \times c}$ ← $sqrt\left(convolve\left(CarteG_{r \times c}^{\circ 2}, 1_{m \times n} \circ \frac{1}{N}\right) - Moy_{CarteG_{r \times c}}^{\circ 2}\right)_{r \times c}$
 $corrScore$ ← $corrI_{r \times c} \circ \frac{1}{stdG_{r \times c} \circ stdL_{1 \times 1}}$

retourne *corrScore*

fin fonction

Algorithme 7 Calcul de la roto-translation nécessaire à appliquer à une carte pour la superposer à une deuxième en exploitant la particularité d'existence des lignes droites de l'environnement intérieur

//

Etape I**nécessite :** Carte globale $CarteG$ et carte locale $CarteL$ $\phi_{alignement} \leftarrow \text{ALLIGNEMENT_CARTE}(CarteG_{r \times c})$ **si** $\phi_{alignement} \neq 0$ **alors** $CarteGa_{r' \times c'} \leftarrow \text{rotation}(CarteG_{r \times c}, \phi_{alignement})$ **fin si** $TH_{CarteGa} \leftarrow$ transformation de Hough de la $CarteGa_{r' \times c'}$ $SH_{CarteGa}^{Rot} \leftarrow$ spectre de Hough de la $CarteGa_{r' \times c'}$ **tant que** Système ultrason en fonctionnement **faire** $CarteL \leftarrow$ mise à jour $TH_{CarteL} \leftarrow$ transformation de Hough de la $CarteL_{m \times n}$ $SH_{CarteGL}^{Rot} \leftarrow$ spectre de Hough de la $CarteL_{m \times n}$ $CorrAngulaire \leftarrow \text{corrélation}(SH_{CarteGa}^{Rot}, SH_{CarteL}^{Rot})$ $[\psi_A \dots \psi_Z] \leftarrow \text{argmax}_{1 \dots k}(\max(CorrAngulaire))$: angles de rot. de la $CarteL_{m \times n}$ $[CarteL_A \dots CarteL_Z] \leftarrow \text{rotation}(CarteL_{m \times n}, [\psi_A \dots \psi_Z])$ **pour tout** $[CarteL_A, \dots, CarteL_Z]$ **faire** $SH_{CarteL_i}^U \leftarrow$ spectre U de la carte locale $CarteL_{A \dots Z}$ $SH_{CarteL_i}^V \leftarrow$ spectre V de la carte locale $CarteL_{A \dots Z}$ $\Delta_{A \dots Z}^U \leftarrow \text{argmax}(\text{corrélation}(SH_{CarteL_{A \dots Z}}^U, SH_{CarteG}^V))$ $\Delta_{A \dots Z}^V \leftarrow \text{argmax}(\text{corrélation}(SH_{CarteL_{A \dots Z}}^V, SH_{CarteG}^U))$ **fin pour**.....
// Etape II - Affiner la position utilisant l'algorithme précédent de corrélation maximale $\text{aprioriPosition} : \{r_{min}, r_{max}, c_{min}, c_{max}, \theta_{min}, \theta_{max}\} \leftarrow$ mise à jour $\text{corrScore}_{r \times c}(\theta_i) \leftarrow \text{SCORE_CORRELATION}(CarteG_{r \times c}, CarteL_{m \times n}, \text{aprioriPosition})$ $\text{maxCorrScore}_i \leftarrow \max(\text{corrScore}_{r \times c}(\theta_i)(:))$ $[\Delta_{theta}] \leftarrow \text{index}(\max(\text{maxCorrScore}_i(:)))$ $[\Delta_U, \Delta_V] \leftarrow \text{index}(\max(\text{corrScore}_{r \times c}(\theta_{Idx})(:)))$ **fin tant que**

fonction ALLIGNEMENT_CARTE($CarteG_{r \times c}$) $TH_{CarteG} \leftarrow$ transformation de Hough de la $CarteG_{r \times c}$ $STH_{CarteG} \leftarrow$ spectre de Hough de la $CarteG_{r \times c}$ $\phi_{alignement} \leftarrow$ angle minimal pour lequel ST_{CarteG} atteint sa valeur maximale**retourne** $\phi_{alignement}$ **fin fonction**

Annexe A.6

Détecteur et descripteur de points d'intérêts

Dans cette annexe nous décrivons brièvement l'approche de détection et d'appariement de points d'intérêt que nous utilisons dans l'algorithme *Structure from Motion* dans la section §IV.4.2 (Appariement et suivi des points).

Soit $\mathbf{Blob}_{5 \times 5}$ et $\mathbf{Checkerboard}_{5 \times 5}$ deux matrices de taille 5×5 tel que détaillé dans les figures A.9(a) et (b). \mathbf{Im}_{Blob} et $\mathbf{Im}_{Checkerboard}$ sont les deux images résultantes après l'application des deux opérateurs en vue de détection des zones et les points d'intérêt dans l'image \mathbf{Im} :

$$\mathbf{Im}_{Blob} = \mathbf{Blob}_{5 \times 5} \circ \mathbf{Im} \quad (\text{A.4})$$

$$\mathbf{Im}_{Checkerboard} = \mathbf{Checkerboard}_{5 \times 5} \circ \mathbf{Im} \quad (\text{A.5})$$

où l'opération matricielle $\mathbf{A} = \mathbf{B}_{5 \times 5} \circ \mathbf{C}$ est définie ainsi :

$$\mathbf{A}_{uv} = \mathbf{B}_{5 \times 5} \cdot \begin{bmatrix} \mathbf{C}_{(u-2)(v-2)} & \cdots & \mathbf{C}_{(u+2)(v-2)} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{(u-2)(v+2)} & \cdots & \mathbf{C}_{(u+2)(v+2)} \end{bmatrix}, \text{ pour } \forall (u, v) \in \mathbf{C} \quad (\text{A.6})$$

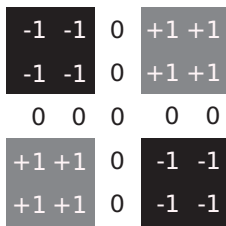
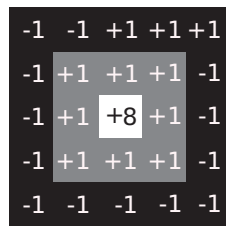
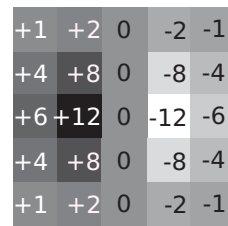
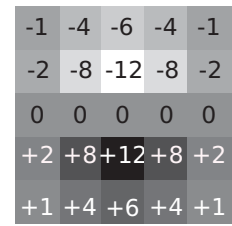
(a) $\mathbf{Checkerboard}_{5 \times 5}$ (b) $\mathbf{Blob}_{5 \times 5}$ (c) $\mathbf{Sob}_{5 \times 5, h}$ (d) $\mathbf{Sob}_{5 \times 5, v}$

FIGURE A.9 – Utilisation du détecteur de zones *Blob* et du détecteur de coins *Checkerboard* pour la détection des points d'intérêts et des opérateurs Sobel pour le calcul du gradient de l'intensité de chaque pixel

L'opérateur Sobel utilisé calcule le gradient de l'intensité de chaque pixel, ce qui indique la direction de la plus forte variation du clair au sombre ainsi que le taux de changement dans cette direction. Notons par $\mathbf{G}_{horizontal}$ et $\mathbf{G}_{vertical}$ les deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point de l'image \mathbf{Im} , voir équations (A.7) et (A.8). En chaque point, les approximations des gradients horizontaux et verticaux peuvent être combinées en \mathbf{G} pour donner une approximation de la norme du gradient ou \mathbf{G}_{uv} dans chaque point (u, v) se calcule suivant l'équation (A.9).

$$\mathbf{G}_h = \mathbf{Sob}_{5 \times 5, h} \circ \mathbf{Im} \quad (\text{A.7})$$

$$\mathbf{G}_v = \mathbf{Sob}_{5 \times 5, v} \circ \mathbf{Im} \quad (\text{A.8})$$

$$\mathbf{G}_{uv} = \sqrt{\mathbf{G}_h(u, v)^2 + \mathbf{G}_v(u, v)^2} \quad (\text{A.9})$$

Alors les opérateurs Sobel de dimensions 5×5 illustrées dans les figures A.9(c) et (d) sont

calculés par les expressions suivantes :

$$\mathbf{Sob}_{5 \times 5, h} = \text{conv2}([1, 2, 1]^T \cdot [1, 2, 1], \mathbf{Sob}_{3 \times 3, h}) \quad (\text{A.10})$$

$$\mathbf{Sob}_{3 \times 3, h} = [1, 2, 1]^T \cdot [1, 0, -1] = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (\text{A.11})$$

$$\mathbf{Sob}_{5 \times 5, v} = \mathbf{Sob}_{5 \times 5, h}^T \quad (\text{A.12})$$

où les opérateurs Sobel de dimensions 3×3 sont définis dans l'équation (A.13).

$$\mathbf{Sob}_{3 \times 3, h} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{et} \quad \mathbf{Sob}_{3 \times 3, v} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (\text{A.13})$$

$$\mathbf{Sob}_{5 \times 5, h} = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} \quad \text{et} \quad \mathbf{Sob}_{5 \times 5, v} = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{bmatrix} \quad (\text{A.14})$$

Les résultats du filtre de Sobel nous sont utiles dans la construction du descripteur de chaque point d'intérêt. Ainsi pour des raisons d'optimisation du temps de calcul, un descripteur de taille 32 pixels est utilisé. Celui-ci est composé des valeurs choisies dans le voisinage du point d'intérêt, 16 valeurs étant choisies dans la matrice $\mathbf{G}_{horizontal}$ et d'autres 16 valeurs dans la matrice $\mathbf{G}_{vertical}$ suivant un masquage comme celui présenté dans la figure A.10.

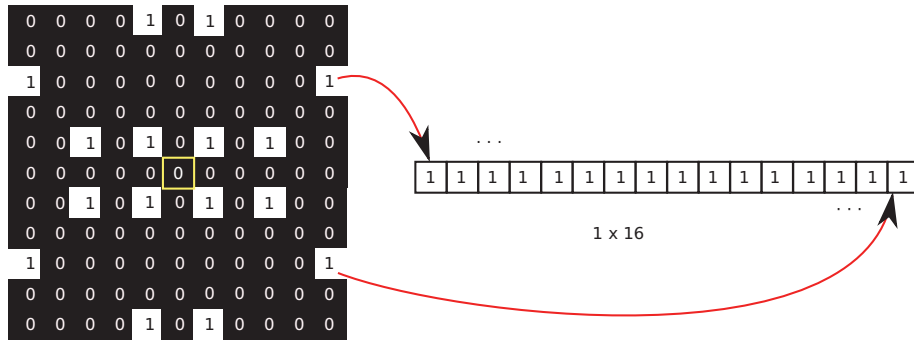


FIGURE A.10 – Masque utilisé dans l'extraction des 16 composants \mathbf{G}_h et 16 composants \mathbf{G}_v pour la création du descripteur de chaque point d'intérêt. En couleur jaune nous représentons le centre du masque qui sera superposé sur le point d'intérêt, en blanc et en noir sont respectivement les cellules indiquant les valeurs utilisées et inutilisées suite à l'application du masque

L'utilisation d'un descripteur de taille 32 à la place de descripteurs qui sont généralement de taille 10×10 à 20×20 présente une plus grande sensibilité et un taux plus élevé de faux appariement mais le problème est résolu par l'appariement avec vérification croisée double comme détaillé dans la figure IV.28. Le critère d'appariement utilisé est un critère de distance entre deux vecteurs qui est défini comme la distance à la norme L_1 également appelée distance de Manhattan (équation (A.15)).

$$d_{Manhattan}(Im_1_{-}D_i, Im_2_{-}D_i) = \sum_{k=1}^{32} |Im_1_{-}D_i(k) - Im_2_{-}D_i(k)| \quad (\text{A.15})$$

Annexe A.7

Estimation des paramètres d'un modèle par l'algorithme RANSAC

Dans cette annexe nous détaillons une méthode d'estimation des paramètres d'un modèle par une approche stochastique du type RANSAC (RANDOM SAMPLE CONSENSUS).

Pour mieux expliquer le concept, prenons l'exemple illustré dans les figures A.11(a) à (c). Initialement nous disposons l'un ensemble d'observations sous la forme d'un nuage de N points 2D (figure A.11(a)). Nous disposons également d'un modèle qui peut être ajusté à ces données. Dans notre cas, nous cherchons à trouver la droite qui interpole le mieux ce nuage de points. Pour cela, 2 points sont tirés aléatoirement et la droite passant par ces points est construite. Ensuite, pour tous les autres $N - 2$ points les distances $d_i = \text{dist}(P_i, \text{Droite})$ sont calculées. Les points se trouvant plus près qu'une distance d_c imposée seront considérés comme des inliers, les autres points comme des outliers. Sur ce principe, nous observons dans la figure A.11(b) qu'uniquement 5 points sur 30 sont des inliers, en échange dans la figure A.11(c) 22 points sur 30 sont des inliers. La solution finale sera celle avec le plus d'inliers.

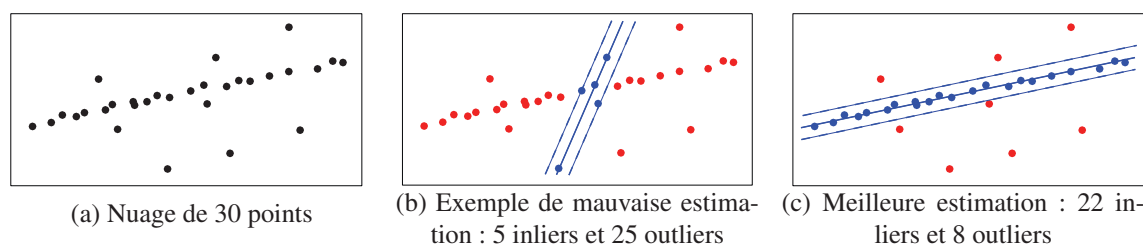


FIGURE A.11 – Estimation de la droite qui interpole le mieux le nuage de 30 points (figure (a)) par une approche stochastique du type RANSAC; les points bleus sont considérés comme des inliers par rapport à la droite choisie et les points rouges des outliers

Nous prenons les notations suivantes :

- n le nombre minimum de données nécessaires pour ajuster le modèle (2 points dans notre cas);
 - k le nombre d'itérations de l'algorithme;
 - p la probabilité que l'algorithme RANSAC produise un résultat correct lorsqu'il sélectionne les n points à une itération
 - w la probabilité de choisir un point pertinent à chaque fois qu'un seul point est sélectionné
- Cela se réécrit sous la forme :

$$w = \frac{\text{nombre de points pertinents dans les données}}{\text{nombre de points dans les données}} \quad (\text{A.16})$$

Cependant, le nombre d'itérations k peut être déterminé à partir d'un résultat théorique. En supposant que les n points nécessaires pour l'estimation d'un modèle sont sélectionnées de manière indépendante, w^n correspond à la probabilité que l'ensemble de n points correspond à des valeurs pertinentes. Le complément $1 - w^n$ correspond à la probabilité qu'au moins un des n points soit un cas aberrant. Sachant que l'algorithme RANSAC exécutera k itérations indépendantes, alors $(1 - w^n)^k$ est la probabilité que l'algorithme ne choisisse jamais un ensemble de n points

qui seraient tous pertinents. Par conséquent,

$$(1 - w^n)^k = 1 - p \quad (\text{A.17})$$

Cela nous permet d'extraire k comme suit :

$$k = \frac{\log(1 - p)}{\log(1 - w^n)} \quad (\text{A.18})$$

avec p imposé généralement à 99% k se calcule ainsi :

$$k = \frac{\log(1 - 0,99)}{\log\left(1 - \left(\frac{22}{30}\right)^2\right)} \approx 6 \text{ itérations} \quad (\text{A.19})$$

ou bien pour un $p = 99,99999\%$ le nombre k dévient

$$k = \frac{\log(1 - 0,9999999)}{\log\left(1 - \left(\frac{22}{30}\right)^2\right)} \approx 15 \text{ itérations} \quad (\text{A.20})$$

Annexe A.8

Algorithme PSO - Particle Swarm Optimization

PSO est un algorithme d'optimisation qui appartient à la classe d'algorithmes métaheuristiques d'optimisation. Il est inspiré du comportement des insectes (ou bien des animaux) et il est basé sur le paradigme de l'intelligence des essaims. Même si le principe est ancien, il a été formalisé pour la première fois par J. Kennedy et R. C. Eberhart [Eberhart et al., 1995].

L'algorithme très simple à implémenter qui s'applique à une très large gamme de problèmes d'optimisation ce qui lui a assuré le succès. Par la suite nous prenons un l'exemple unidimensionnel (voir les figures A.12(a) à (d)).

Supposons que quatre abeilles volent à la même hauteur par rapport à un référentiel fixé (niveau des océans), que chaque abeille est capable de mesurer la distance au sol et que leur but est de déterminer l'endroit où la hauteur est la plus basse. En plus les abeilles sont capables de communiquer entre elles à distance.

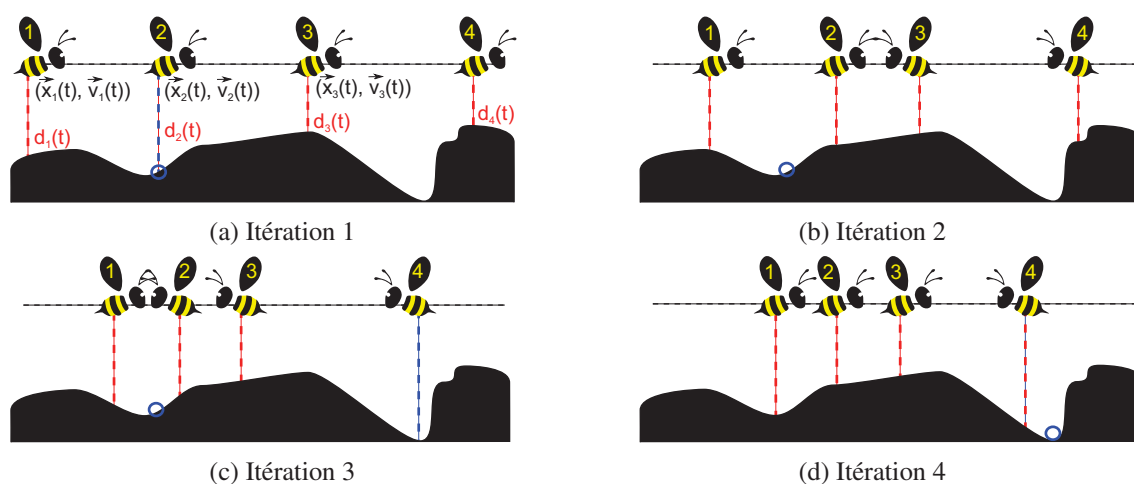


FIGURE A.12 – Illustration du fonctionnement l'algorithme PSO pour un cas unidimensionnel. Les lignes pointillées colorées en rouge et bleu indiquent les distances mesurées par les abeilles. La couleur bleue indique qu'une abeille a mesuré un nouveau minimum global qui est signalé par un cercle toujours de couleur bleue

Même si nous pouvons facilement déterminer le minimum global, car nous avons une vue d'ensemble sur l'espace de recherche, pour les abeilles ce n'est pas le cas. Initialement quatre abeilles sont disposées uniformément (ou aléatoirement) dans l'espace et volent avec une vitesse \vec{v}_i , $i = \{1, 2, 3, 4\}$. A la première itération, la deuxième abeille trouve le minimum et le signale à tout l'essaim. Aux étapes suivantes chaque abeille va décider de continuer sa recherche suivant trois critères : l'inertie, l'attitude personnelle (composante cognitive), l'attitude sociale (composante sociale).

Cela se formalise suivant l'équation (A.21) où l'état de chaque particule (abeille dans notre cas) est décrit par sa position $x_i(t)$ et par sa vitesse $v_i(t)$:

$$\begin{cases} v_i(t+1) = w v_i(t) + r_1 c_1 [MinP_i(t) - x_i(t)] + r_2 c_2 [MinG(t) - x_i(t)] \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (\text{A.21})$$

avec :

- $I v_i(t)$: composante d'inertie de la particule i et w un coefficient d'inertie ;
- $r_1 c_1 [P_i(t) - x_i(t)]$: composante cognitive ;
- $r_2 c_2 [g_i(t) - x_i(t)]$: composante sociale ;
- c_1, c_2 : coefficients d'accélération ;
- r_1, r_2 : aléatoirement choisis suivant une loi uniforme, $r_1 \sim u(0, \phi_1)$ et $r_2 \sim u(0, \phi_2)$;
- $\text{Min}P_i$: la meilleure solution atteinte par la particule i ;
- $\text{Min}G$: la meilleure solution globale.

Partant de ce principe, beaucoup de versions ont été développées pour mieux explorer l'espace utilisant plusieurs essaims comme dans les travaux de [Yazdani et al., 2013]. Nous citons également les travaux de [Xue et al., 2013] sur l'optimisation multi-objective.

Bibliographie

- K. Adriantavinson, P. Chevrel, and G. Martin. Projet de fin d'études. developpement d'algorithmes liés à la localisation d'un véhicule autonome lors de manoeuvre de parking. Renault et Ecole des mines de Nantes, 2015. (page 162).
- S. Agarwal, K. Mierle, and Others. Ceres solver. <http://ceres-solver.org>. (page 132).
- S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79. IEEE, 2009. (page 132).
- S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *Computer Vision–ECCV 2010*, pages 29–42. Springer, 2010. (page 83).
- D. Agoiz Badia. Simulador de aplicaciones openmp basado en dimemas. 2016. (page 107).
- O. Ait Aider. *Model Based Localization for an Indoor Mobile Robot*. Theses, Université d'Evry-Val d'Essonne, Dec. 2002. (page 80, 80).
- A. Alam. Fuel-efficient distributed control for heavy duty vehicle platooning. 2011. (page 4).
- C. Alia, T. Gilles, T. Reine, and C. Ali. Local trajectory planning and tracking of autonomous vehicles, using clothoid tentacles method. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 674–679. IEEE, 2015. (page 172).
- J. Altmann, P. Asaro, N. Sharkey, and R. Sparrow. Armed military robots : editorial. *Ethics and Information Technology*, 15(2) :73, 2013. (page 2).
- M. Aly. Real time detection of lane markers in urban streets. In *IEEE Intelligent Vehicles Symposium*, pages 7–12. IEEE, 2008. (page 43).
- R. Ambrus, A. Vibhute, G. Dubey, and A. Kapuria. Unmanned vehicles conversion kits : a minimally invasive system for the conversion of regular vehicles into tele-operable and autonomous platforms. In *Proceedings of Conference on Advances In Robotics*, pages 1–6. ACM, 2013. (page 13, 13).
- J.-Y. Antoine. Systèmes d'information géographique. *cours SIG, Université François Rabelais de Tours*, [tp ://www.info.univtours.fr/~antoine/documents_enseignement/SIG_CM_I.pdf](http://www.info.univtours.fr/~antoine/documents_enseignement/SIG_CM_I.pdf) (uploaded on 08/25/2014), 2014. (page 37).
- S. Aronoff. Geographic information systems : a management perspective. 1989. (pages 38, 77).
- I. Ashokaraj, A. Tsourdos, P. Silson, and B. White. Sensor based robot localisation and navigation : Using interval analysis and extended kalman filter. In *Control Conference, 2004. 5th Asian*, volume 2, pages 1086–1093. IEEE, 2004. (page 80).
- A. Asvadi, P. Peixoto, and U. Nunes. Two-stage static/dynamic environment modeling using voxel representation. In *Robot 2015 : Second Iberian Robotics Conference*, pages 465–476. Springer, 2016. (page 27).
- D. Aubert, F. Nashashibi, R. Chapuis, and F. MARNIER. Projet arcoss, ameliorer la perception de l'environnement du vehicule. *Revue Generale des Routes*, 2005. (page 8).
- Audi piloted driving. Fascinating performance : the audi rs 7 piloted driving concept at the hockenheimring, 2016. URL http://www.audi.com/com/brand/en/vorsprung_durch_technik/content/2014/10/piloted-driving.htm. (page 11, 11).
- C. Audras, A. Comport, M. Meilland, and P. Rives. Real-time dense rgb-d localisation and mapping. In *Australian Conference on Robotics and Automation*, 2011. (page 35).
- R. Aufrère, F. Marmoiton, R. Chapuis, F. Collange, and J.-P. Dérutin. Road detection and vehicle tracking by vision for adaptive cruise control. *The International Journal of Robotics Research*, 20(4) :267–286, 2001. (page 43).
- Automotive Sensors and Electronics Expo. <http://www.automotivesensors2015.com/>, 2015. No citations.
- C. Aynaud. *Localisation précise et fiable de véhicules par approche multisensorielle*. PhD thesis, Clermont-Ferrand 2, 2015. (page 81).
- C. Aynaud, C. Bernay-Angeletti, R. Chapuis, R. Aufrere, and C. Debain. Approche top-down de perception multisensorielle : application à la localisation de véhicules. In *Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014*, 2014. (page 81).
- A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, et al. Odin : Team victortango's entry in the darpa urban challenge. *Journal of Field Robotics*, 25(8) :467–492, 2008. (page 45).
- D. Ball, S. Heath, J. Wiles, G. Wyeth, P. Corke, and M. Milford. Openratslam : an open source brain-based slam system. *Autonomous Robots*, 34(3) :149–176, 2013. No citations.
- Balyo. Nouvelle génération des chariots de manutention automatiques, 2016. URL <http://www.balyo.com/>. (pages 2, 3, 4, 41).
- J. Bammert, K. Hoffsommer, S. Kossmann, and V. Vovkushevsky. Method for operating a driver assistance system when parking a vehicle in a parking space, Sept. 10 2013. URL <https://www.google.com/patents/US8531313>. US Patent 8,531,313. (page 17).
- T. Bandyopadhyay, Y. Li, M. H. Ang, and D. Hsu. A greedy strategy for tracking a locally predictable target among obstacles. In *ICRA 2006. Proceedings 2006 IEEE International Conference on Robotics and Automation*, pages 2342–2347. IEEE, 2006. (page 27).
- N. Banovic, T. Buzali, F. Chevalier, J. Mankoff, and A. K. Dey. Modeling and understanding human routine behavior. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 248–260. ACM, 2016. (page 189).

-
- J. P. Barreto. A unifying geometric representation for central projection systems. *Computer Vision and Image Understanding*, 103(3) : 208–217, 2006. (page 64).
- J. P. d. A. Barreto. General central projection systems : modeling, calibration and visual servoing. 2004. (page 65).
- B. Barshan and H. F. Durrant-Whyte. Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3) :328–342, 1995. (page 78).
- H. Barth and N. Jecker. Method and device for planning a path when parking a vehicle, July 30 2013. URL <https://www.google.com/patents/US8497782>. US Patent 8,497,782. (page 17).
- A. Bartoli. *Reconstruction et alignement en vision 3D : points, droites, plans et caméras*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2003. (page 148).
- Baumer. Passion for sensors, 2016. URL <http://www.baumer.com/>. (page 62, 62).
- H. Bay, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. (page 98).
- H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3) :346–359, 2008. (page 98).
- E. Bazulin. The calibration of an ultrasonic antenna array installed on a wedge. *Russian Journal of Nondestructive Testing*, 50, 2014. (page 49).
- T. Becherrawy. *Optique géométrique*. De Boeck Supérieur, 2005. (page 59).
- P. Beeson, N. K. Jong, and B. Kuipers. Towards autonomous topological place detection using the extended voronoi graph. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4373–4379. IEEE, 2005. (page 169).
- A. Ben-Afia, L. Deambrogio, D. Salós, A.-C. Escher, C. Macabiau, L. Soulier, and V. Gay-Bellile. Review and classification of vision-based localisation techniques in unknown environments. *IET Radar, Sonar & Navigation*, 8(9) :1059–1072, 2014. (page 79).
- R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *Computer Vision-ECCV 2014 Workshops*, pages 613–627. Springer, 2014. (page 44).
- J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9) :509–517, 1975. (page 115).
- C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa. Overview of platooning systems. In *Proceedings of the 19th ITS World Congress*, 2012. (page 4).
- C. Bernay-Angeletti, F. Chabot, C. Aynaud, R. Aufrere, and R. Chapuis. A top-down perception approach for vehicle pose estimation. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2240–2245. IEEE, 2015. (page 81).
- M. Bertozzi and A. Broggi. Gold : A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, 7(1) :62–81, 1998. (page 43).
- M. Bertozzi, A. Broggi, and A. Fascioli. Vision-based intelligent vehicles : State of the art and perspectives. *Robotics and Autonomous systems*, 32(1) :1–16, 2000. (page 7).
- M. Bevis, S. Businger, T. A. Herring, C. Rocken, R. A. Anthes, and R. H. Ware. Gps meteorology : Remote sensing of atmospheric water vapor using the global positioning system. *Journal of Geophysical Research : Atmospheres*, 97, 1992. (page 77).
- A. Bishko, A. A. Samokrutov, and V. G. Shevaldykin. Ultrasonic echo-pulse tomography of concrete using shear waves low-frequency phased antenna arrays. In *Proceedings of the 17th World Conference on Nondestructive Testing*, volume 25, page 2008, 2008. (page 49).
- G. Blanc, T. Chateau, Y. Mezouar, O. A. Aider, P. Martinet, L. Eck, V. Moreau, and A. Nadim. Implementation of a vision-based navigation framework on a house mobile robot prototype. In *International Conference on Advances in Intelligent Systems-Theory and Applications, AISTA*, volume 4, pages 15–18, 2004. (page 84).
- G. Blanc, Y. Mezouar, and P. Martinet. Indoor navigation of a wheeled mobile robot along visual routes. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 3354–3359. IEEE, 2005. (page 84).
- J.-L. Blanco, J. Gonzalez, and J.-A. Fernandez-Madrigal. A new method for robust and efficient occupancy grid-map matching. In *Pattern Recognition and Image Analysis*, pages 194–201. Springer, 2007. (page 81).
- J.-L. Blanco, J.-A. Fernández-Madrigal, and J. González. Efficient probabilistic range-only slam. In *IROS 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1017–1022. IEEE, 2008. No citations.
- Blue Frog Robotics. Buddy, le robot compagnon, 2016. URL <http://www.bluefrogrobotics.com/>. (pages 2, 3).
- A. Bobtsov, A. Borgul, K. Zimenko, and A. Pyrkin. Control approaches for complicated self-unstable plants with applications for two-wheel mobile system. In *Manufacturing Modelling, Management, and Control*, volume 7, pages 609–613, 2013. (page 13, 13).
- J.-D. Boissonnat, A. Cerezo, and J. Leblond. *A note on shortest paths in the plane subject to a constraint on the derivative of the curvature*. PhD thesis, INRIA, 1994. (page 171).
- L. Bombini, P. Cerri, P. Medici, and G. Alessandretti. Radar-vision fusion for vehicle detection. In *Proceedings of International Workshop on Intelligent Transportation*, pages 65–70, 2006. (page 36).
- P. Bonnifait, P. Bouron, P. Crubillé, and D. Meizel. Data fusion of four abs sensors and gps for an enhanced localization of car-like vehicles. In *2001 ICRA. IEEE International Conference on Robotics and Automation*, volume 2, pages 1597–1602. IEEE, 2001. (pages 76, 90, 91).
- P. Bonnifait, P. Bouron, D. Meizel, and P. Crubille. Dynamic localization of car-like vehicles using data fusion of redundant abs sensors. *The Journal of Navigation*, 56 :429–441, 2003. (page 76).
- J. Borenstein and Y. Koren. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 7(4) :535–539, 1991a. (page 129).

-
- J. Borenstein and Y. Koren. Real-time map building for fast mobile robot obstacle avoidance. In *Fibers' 91, Boston, MA*, pages 74–81. International Society for Optics and Photonics, 1991b. (page 129).
- A. Borgul, K. Zimenko, A. Bobtsov, and A. Pyrkin. Control approaches for complicated self-unstable plants with applications for two-wheel mobile robot motobot in educational purposes. In *Advances in Control Education*, volume 9, pages 107–111, 2012. (page 13, 13).
- J.-Y. Bouguet. Camera calibration toolbox for matlab. 2004. (page 66).
- W. Boyle and G. Smith. Charge-coupled devices-a new approach to mis device structures. *Spectrum, IEEE*, 8(7) :18–27, 1971. (page 55).
- W. Boyle and G. Smith. Three dimensional charge coupled devices, Mar. 12 1974. US Patent 3,796,927. (page 55).
- W. S. Boyle. pumped solid state maser, Sept. 26 1961. US Patent 3,002,156. (page 55).
- G. Bradski and A. Kaehler. *Learning OpenCV : Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008. (page 66).
- K. A. Braitman, A. T. McCartt, D. S. Zuby, and J. Singer. Volvo and infiniti drivers' experiences with select crash avoidance technologies. *Traffic injury prevention*, 11(3) :270–278, 2010. (page 5).
- P. Brand. *reconstruction tridimensionnelle à partir d'une caméra en mouvement : de l'influence de la précision*. PhD thesis, Université Claude Bernard-Lyon I, 1995. (page 66).
- S. Brand, E. C. Weiss, R. M. Lemor, and M. C. Kolios. High frequency ultrasound tissue characterization and acoustic microscopy of intracellular changes. *Ultrasound in medicine & biology*, 34(9) :1396–1407, 2008. (page 49).
- J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1) :25–30, 1965. (page 129).
- G. Bresson. *Localisation d'une flotte de véhicules communicants par approche de type SLAM visuel décentralisé*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2014. (page 42).
- G. Bresson, T. Féraud, R. Aufrère, P. Checchin, and R. Chapuis. Parsimonious real time monocular slam. In *2012 IEEE Intelligent Vehicles Symposium*, pages 511–516. IEEE, 2012. (page 32).
- G. Bresson, T. Féraud, R. Aufrère, P. Checchin, and R. Chapuis. Real-time monocular slam with low memory requirements. *IEEE Transactions on Intelligent Transportation Systems*, 16 :1827–1839, 2015. (pages 32, 82).
- K. A. Brookhuis, C. J. van Driel, T. Hof, B. van Arend, and M. Hoedemaeker. Driving with a congestion assistant; mental workload and acceptance. *Applied ergonomics*, 40(6) :1019–1025, 2009. (page 4).
- H. Bruyninckx. Bayesian probability. *CiteSeer, não publicado em con*, 2002. (page 81).
- J. Bryner. *Applications of ghz ultrasound : Material characterization and wave propagation in microstructures*. PhD thesis, ETH, 2009. (page 49).
- M. Buehler, K. Iagnemma, and S. Singh. *The DARPA urban challenge : autonomous vehicles in city traffic*, volume 56. springer, 2009. (page 45).
- X.-N. Bui, J.-D. Boissonnat, P. Soueres, and J.-P. Laumond. Shortest path synthesis for dubins non-holonomic robot. In *1994. Proceedings., IEEE International Conference on Robotics and Automation*, pages 2–7. IEEE, 1994. (page 170).
- W. Burgard, D. Fox, and S. Thrun. Markov localization for mobile robots in dynamic environments. *arXiv preprint arXiv :1106.0222*, 2011. (page 81).
- M. Cakmak and L. Takayama. Towards a comprehensive chore list for domestic robots. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 93–94. IEEE Press, 2013. (page 2).
- R. Campan and F. Scapini. *Ethologie : approche systémique du comportement*. De Boeck Supérieur, 2002. (page 38).
- K. Čapek and P. Selver. *R.U.R. (Rossum's Universal Robots) : A Play in Three Acts and an Epilogue*. H. Milford, 1925. (page 2).
- L. Carlone, P. F. Alcantarilla, H.-P. Chiu, Z. Kira, and F. Dellaert. Mining structure fragments for smart bundle adjustment. In *Proceedings of British Machine Vision Conference (BMVC)*, 2014. (page 83).
- S. Carpin. Merging maps via hough transform. In *IROS 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1878–1883. IEEE, 2008. (page 107).
- E. Casapietra, T. H. Weisswange, C. Goerick, F. Kummert, and J. Fritsch. Building a probabilistic grid-based road representation from direct and indirect visual cues. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 273–279. IEEE, 2015. (page 43).
- V. G. Cerf. What's a robot? *Communications of the ACM*, 56 :7–7, 2013. No citations.
- M. Chafkin. Udacity's sebastian thrun, godfather of free online education, changes course. *Fast Company*, 14, 2013. (page 19).
- E. Chan. Sartre automated platooning vehicles. In *Transport Research Arena (TRA) 5th Conference : Transport Solutions from Research to Deployment*, 2014. (page 42, 42).
- B. Chapman, G. Jost, and R. Van Der Pas. *Using OpenMP : portable shared memory parallel programming*, volume 10. MIT press, 2008. (page 107).
- C. Chauvel, Y. Page, B. Fildes, and J. Lahaussé. Automatic emergency braking for pedestrians effective target population and expected safety benefits. In *23rd international technical conference on the enhanced safety of vehicles (ESV 2013)*, 2013. (page 5).
- A. Chebly, R. Talj, and G. Tagne. Planification des trajectoires pour la navigation des véhicules autonomes par la méthode des tentacules en forme de clothoïdes. Technical report, Technical report, Université de technologie de Compiègne, 2014. (page 172).
- F. Chemat. *Éco-extraction du végétal : procédés innovants et solvants alternatifs*. Dunod, 2011. (page 49).
- J. Chen, D. Zeng, C. Kang, and Z. J. Zhang. A triangle division based point matching for image registration. In *Advanced Materials Research*, volume 765, pages 726–729. Trans Tech Publ, 2013. (page 155).
- M. Chirca. Dispositif et procédé de détection de place de stationnement libre pour un véhicule automobile, 2015a. FR Patent App. 15-57,491. (page 23).
- M. Chirca. Procédé et dispositif de réglage automatique d'éléments du poste de conduite, 2015b. FR Patent App. 15-63,083. (page 23).

-
- M. Chirca and G. Martin. Procédé et dispositif de détermination de positions spatiales et de paramètres cinématiques d'objets dans l'environnement d'un véhicule, 2015. FR Patent App. 15-63,078. (page 23).
- M. Chirca and N. Minoiu-Enache. Dispositif et procédé d'aide au stationnement pour un véhicule automobile, 2015. FR Patent App. 15-57,593. (page 23).
- M. Chirca, R. Chapuis, G. Martin, C. Debain, and R. Lenain. Autonomous valet parking system architecture. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2619–2624, Sept 2015a. (page 23).
- M. Chirca, G. Martin, N. Minoiu-Enache, H. Vorobieva, and S. Guegan. Device and method for automatically parking a motor vehicle (dispositif et procédé de stationnement automatique d'un véhicule automobile), 2015b. Patent WO2016128654A1 (FR3032410A1). (page 23).
- K. S. Chong and L. Kleeman. Accurate odometry and error modelling for a mobile robot. In *1997. Proceedings., 1997 IEEE International Conference on Robotics and Automation*, volume 4, pages 2783–2788. IEEE, 1997. (page 78).
- Z. Chong, B. Qin, T. Bandyopadhyay, T. Wongpiromsarn, B. Rebsamen, P. Dai, E. Rankin, and M. H. Ang Jr. Autonomy for mobility on demand. In *Intelligent Autonomous Systems 12*, pages 671–682. Springer, 2013. (page 14).
- C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14 :462–467, 1968. (page 116).
- R. D. Christ and R. L. Wernli Sr. *The ROV manual : a user guide for remotely operated vehicles*. Butterworth-Heinemann, 2013. (page 3).
- K. Chu, M. Lee, and M. Sunwoo. Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE Transactions on Intelligent Transportation Systems*, 13(4) :1599–1616, 2012. (page 172).
- J. Civera, A. J. Davison, and J. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24 (5) :932–945, 2008. (page 82).
- J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel. 1-point ransac for extended kalman filtering : Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5) :609–631, 2010. (page 82).
- A. Cohen, C. Zach, S. N. Sinha, and M. Pollefeys. Discovering and exploiting 3d symmetries in structure from motion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1514–1521. IEEE, 2012. (page 32, 32).
- P. Corcoran and P. Mooney. Characterising the metric and topological evolution of openstreetmap network representations. *The European Physical Journal Special Topics*, 215(1) :109–122, 2013. (page 39).
- C. Coué. *Modèle bayésien pour l'analyse multimodale d'environnements dynamiques et encombrés : Application à l'assistance à la conduite en milieu urbain*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2003. (pages 129, 130).
- J. Courbon. *Navigation de robots mobiles par mémoire sensorielle*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2009. (page 65).
- J. Courbon, Y. Mezouar, L. Eck, and P. Martinet. A generic fisheye camera model for robotic applications. In *IROS 2007. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1683–1688. IEEE, 2007. (page 65).
- D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3001–3008. IEEE, 2011. No citations.
- M. Cummins and P. Newman. Fab-map : Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6) :647–665, 2008. (pages 84, 116, 117).
- M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9) :1100–1123, 2011. (pages 84, 116, 117, 118).
- D. Cunningham. The oldest maps of the world : Deciphering the hand paintings of cueva de el castillo caves in spain and lascaux france. *Update*, 2011. (page 37).
- T. Czaplak and J. Wrona. Technology development of military applications of unmanned ground vehicles. In *Vision Based Systems for UAV Applications*, pages 293–309. Springer, 2013. (page 9, 9).
- L. Dagum and R. Enon. Openmp : an industry standard api for shared-memory programming. *IEEE Computational Science & Engineering*, 5 :46–55, 1998. (page 107).
- DARPA. Creating breakthrough technologies for national security. <http://www.darpa.mil/>, 2016. (page 8).
- P. Davidson and A. Spinoulas. Autonomous vehicles : what could this mean for the future of transport? In *Australian Institute of Traffic Planning and Management (AITPM) National Conference, 2015, Brisbane, Queensland, Australia*, 2015. (page 7).
- A. Dávila and M. Nombela. Sartre : Safe road trains for the environment. In *Conference on Personal Rapid Transit PRT@ LHR*, volume 3, pages 2–3, 2010. (page 42, 42).
- J. Davins-Valldaura, S. Moussaoui, F. Plestan, and G. P. Gil. Parego extensions for multi-objective optimization of expensive evaluation function. In *World congress on global optimization*, 2015. (page 6).
- J. Davins-Valldaura, S. Moussaoui, G. Pita-Gil, and F. Plestan. Parego extensions for multi-objective optimization of expensive evaluation functions. *Journal of Global Optimization*, pages 1–18, 2016. (page 6).
- J. Davis, T. Herring, I. Shapiro, A. Rogers, and G. Elgered. Geodesy by radio interferometry : Effects of atmospheric modeling errors on estimates of baseline length. *Radio science*, 20, 1985. (page 77).
- J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012. (page 46).
- E. Degtariova-Kostova and V. Kostov. *Irregularity of optimal trajectories in a control problem for a car-like robot*. PhD thesis, INRIA, 1998. (page 171).
- B. Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennykh Nauk*, 7(793-800), 1934. (pages 159, 210).
- F. Dellaert. Visual slam tutorial : Bundle adjustment. 2014. (page 83).

-
- P. Delmas. *Génération active des déplacements d'un véhicule agricole dans son environnement*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2011. (pages 97, 172).
- Delphi Drive. Us coast-to-coast automated drive, 2016. URL <http://www.delphi.com/delphi-drive>. (pages 11, 12).
- K. G. Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 4 :2–3, 2010. (page 155).
- M. Devy, D. L. A. Ojeda, and A. Herbulot. Mobile object detection from an embedded camera : how to minimize the latency time ? *IFAC Proceedings Volumes*, 44(1) :2220–2225, 2011. (page 8).
- J. Diard. *La carte bayésienne : un modèle probabiliste hiérarchique pour la navigation en robotique mobile*. PhD thesis, Grenoble, INPG, 2003. (page 81).
- E. D. Dickmanns. The development of machine vision for road vehicles in the last decade. In *2002. IEEE Intelligent Vehicle Symposium*, volume 1, pages 268–281. IEEE, 2002. (page 8).
- M. Dickow, A. Dahlmann, C. Alwardt, F. Sauer, and N. Schörmig. First steps towards a multidimensional autonomy risk assessment (mara) in weapons systems. 2015. (page 9, 9).
- M. Diftler, J. Badger, C. Joyce, E. Potter, and L. Pike. Robonaut 2-building a robot on the international space station. 2015. (page 2).
- E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1) :269–271, 1959. (page 170).
- H. Dong and T. D. Barfoot. Lighting-invariant visual odometry using lidar intensity imagery and pose interpolation. In *Field and Service Robotics*, pages 327–342. Springer, 2014. (page 79).
- J. Dou and J. Li. Robust image matching based on sift and delaunay triangulation. *Chinese Optics Letters*, 10(s1) :11001, 2012. (page 155).
- B. Douillard, A. Quadros, P. Morton, J. P. Underwood, M. De Deuge, S. Hugosson, M. Hallström, and T. Bailey. Scan segments matching for pairwise 3d alignment. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3033–3040. IEEE, 2012. (page 31).
- G. Dubbelman and B. Browning. Closed-form online pose-chain slam. In *IEEE International Conference on Robotics and Automation*, pages 5190–5197. IEEE, 2013. No citations.
- L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3) :497–516, 1957. (pages 170, 171, 186).
- F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica. Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, 96 :59–69, 2014. (page 169).
- M. Duckham, L. Kulik, M. Worboys, and A. Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10) :3224–3236, 2008. (pages 159, 210).
- R. C. Eberhart, J. Kennedy, et al. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pages 39–43. New York, NY, 1995. (pages 188, 221).
- A. H. Eichelberger and A. T. McCartt. Toyota drivers' experiences with dynamic radar cruise control, pre-collision system, and lane-keeping assist. *Journal of Safety Research*, 2015. (page 5).
- E. Einhorn, C. Schröter, and H.-M. Gross. Can't take my eye off you : Attention-driven monocular obstacle detection and 3d mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 816–821. IEEE, 2010. (page 45).
- S. El Fkihî. *Modèles probabilistes indexés par les arbres : application à la détection de la peau dans les images couleur*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I, 2008. (page 116).
- A. Eliazar and R. Parr. Dp-slam : Fast, robust simultaneous localization and mapping without predetermined landmarks. In *IJCAI*, volume 3, pages 1135–1142, 2003. (page 83).
- J. Engel, T. Schöps, and D. Cremers. Lsd-slam : Large-scale direct monocular slam. In *Computer Vision—ECCV 2014*, pages 834–849. Springer, 2014. (page 132).
- J. F. Engelberger. *Robotics in practice : management and applications of industrial robots*. Springer Science & Business Media, 2012. (page 2).
- F. Erbs, B. Schwarz, and U. Franke. Stixmentation-probabilistic stixel based traffic scene labeling. In *BMVC*, pages 1–12, 2012. (page 33).
- T. Ertl. *Computer graphics—principles and practice*. Springer, 1996. (page 129).
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. (pages 158, 207).
- J. S. Esteves, A. Carvalho, and C. Couto. Generalized geometric triangulation algorithm for mobile robot absolute self-localization. In *2003. ISIE'03. 2003 IEEE International Symposium on Industrial Electronics*, volume 1, pages 346–351. IEEE, 2003. (page 77).
- C. Estrada, J. Neira, and J. D. Tardós. Hierarchical slam : Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4) :588–596, 2005. (page 84).
- N. Euro and A. T. Protocol. Euro ncap, 2013. (page 5).
- M. F. Fallon, H. Johannsson, J. Brookshire, S. Teller, and J. J. Leonard. Sensor fusion for flexible human-portable building-scale mapping. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4405–4412. IEEE, 2012. (page 79).
- T. Féraud. *Rejeu de chemin et localisation monoculaire : Application du Visual SLAM sur carte peu dense en environnement extérieur contraint*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2011. (pages 82, 95, 97, 98, 99, 102, 213).
- T. Féraud, P. Checchin, R. Aufrère, and R. Chapuis. Communicating vehicles in convoy and monocular vision-based localization. In *Intelligent Autonomous Vehicles*, volume 7, pages 73–78, 2010a. (pages 8, 96).
- T. Féraud, P. Checchin, and R. Chapuis. Véhicules communicants en convoi et localisation par vision monoculaire. *Caen-FRANCE. 17ème congrès francophone AFRIF-AFIA, Reconnaissance des Formes et Intelligence Artificielle. RFIA*, 2010b. (page 8).

-
- T. Féraud, R. Chapuis, R. Aufrère, and P. Checchin. Improving results of rational non-linear observation functions using a kalman filter correction. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–7. IEEE, 2011a. (page 8).
- T. Féraud, R. Chapuis, R. Aufrère, and P. Checchin. Kalman filter correction with rational non-linear functions : Application to visual-slam. In *ECMR*, pages 232–238, 2011b. (page 32).
- C. Fernandez, R. Izquierdo, D. Fernandez Llorca, and M. A. Sotelo. A comparative analysis of decision trees based classifiers for road detection in urban environments. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, pages 719–724. IEEE, 2015. (page 43).
- D. Filliat. Robotique mobile. *ENSTA ParisTech, Paris France*, 2011. (pages 39, 40, 74, 81).
- R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4 :1–9, 1974. (page 130).
- M. A. Fischler and R. C. Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981. (pages 142, 155).
- A. Fix, R. Matthey, A. Amediek, G. Ehret, F. Gruet, C. Kiemle, V. Klein, G. Mileti, J. P. do Carmo, and M. Quatrevalet. Investigations on frequency and energy references for a space-borne integrated path differential absorption lidar. In *Proceedings of the International Conference on Space Optics*, 2014. (page 35).
- W. J. Fleming. Overview of automotive sensors. *Sensors Journal, IEEE*, 1, 2001. (page 27).
- W. J. Fleming. New automotive sensors—a review. *Sensors Journal, IEEE*, 8, 2008. (page 27).
- T. W. Foresman. *The history of geographic information systems : perspectives from the pioneers*. Prentice Hall, 1998. (pages 38, 77).
- C. Forney, J. Forrester, B. Bagley, W. McVicker, J. White, T. Smith, J. Batryn, A. Gonzalez, J. Lehr, T. Gambin, et al. Surface reconstruction of maltese cisterns using rov sonar data for archeological study. In *Advances in Visual Computing*, pages 461–471. Springer, 2011. (page 3).
- M. Forster, R. Frank, M. Gerla, and T. Engel. Improving highway traffic through partial velocity synchronization. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 5573–5578. IEEE, 2012. (page 4).
- D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, pages 391–427, 1999. (page 81).
- D. H. D. Fox, W. Burgard, and S. Thrun. A highly efficient fastslam algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proceedings IEEE RSJ International Conference on Intelligent Robots and Systems*, 2003. (page 83).
- J. Fraden. *Handbook of modern sensors : physics, designs, and applications*. Springer Science & Business Media, 2004. (page 27).
- T. Fraichard and A. Scheuer. From reeds and shepp’s to continuous-curvature paths. *IEEE Transactions on Robotics*, 20 :1025–1035, 2004. (page 172).
- U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6d-vision : Fusion of stereo and motion for robust environment perception. In *Pattern Recognition*, pages 216–223. Springer, 2005. (page 33, 33).
- U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. Herrtwich. Making berthä see. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 214–221, 2013. (page 33, 33).
- F. Fraundorfer, D. Scaramuzza, and M. Pollefeys. A constricted bundle adjustment parameterization for relative scale estimation in visual odometry. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1899–1904. IEEE, 2010. (page 79).
- S. Fuhrmann, F. Langguth, and M. Goesele. Mve—a multiview reconstruction environment. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage (GCH)*, volume 6, page 8. Citeseer, 2014. (page 132).
- T. Furukawa, K. Takami, X. Tong, D. Watman, A. Hamed, R. Ranasinghe, and G. Dissanayake. Map-based navigation of an autonomous car using grid-based scan-to-map matching. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V003T01A005–V003T01A005. American Society of Mechanical Engineers, 2015. (page 81).
- D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel. Real-time monocular object slam. *Robotics and Autonomous Systems*, 75 : 435–449, 2016. (page 84).
- T. Garcia-Valverde, A. Garcia-Sola, H. Hagra, J. A. Dooley, V. Callaghan, and J. A. Botia. A fuzzy logic-based system for indoor localization using wifi in ambient intelligent environments. *IEEE Transactions on Fuzzy Systems*, 21(4) :702–718, 2013. (page 77).
- M. Gariépy, J. Trépanier, K. Szilder, S. McIlwain, C. Pilcher, J. Chung, M. Ringshandl, M. N. Ensan, D. Zimcik, G. Shepherd, et al. *Journal aéronautique et spatiale du canada*. 2015. (page 3).
- S. K. Gehrig, F. Eberli, and T. Meyer. A real-time low-power stereo vision engine using semi-global matching. In *Computer Vision Systems*, pages 134–143. Springer, 2009. (page 33).
- A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Computer Vision—ACCV 2010*, pages 25–38. Springer, 2010. (pages 115, 146, 147).
- A. Geiger, J. Ziegler, and C. Stiller. Stereoscan : Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. IEEE, 2011. (pages 115, 146).
- C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical implications. In *Computer Vision—ECCV 2000*, pages 445–461. Springer, 2000. (page 64).
- S. Giese, D. Carr, and J. Chahl. Implications for unmanned systems research of military uav mishap statistics. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1191–1196, 2013. (page 3).

-
- A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth. Openfabmap : An open source toolbox for appearance-based loop closure detection. In *IEEE International Conference on Robotics and Automation*, pages 4730–4735. IEEE, 2012. (pages 84, 116).
- S. Golodetz, M. Sapienza, J. P. Valentin, V. Vineet, M.-M. Cheng, V. A. Prisacariu, O. Kähler, C. Y. Ren, A. Arnab, S. L. Hicks, et al. Semanticpaint : interactive segmentation and learning of 3d worlds. In *ACM SIGGRAPH 2015 Emerging Technologies*, page 22. ACM, 2015. (page 46).
- G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B : Numerical Analysis*, 2 :205–224, 1965. (pages 145, 153).
- GoogleCar. Google self-driving car project, 2016. URL <https://www.google.com/selfdrivingcar>. (page 10).
- D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. O. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier. Mechatronic design of nao humanoid. In *ICRA'09. IEEE International Conference on Robotics and Automation*, pages 769–774. IEEE, 2009. (page 4).
- C. Grana, D. Borghesani, M. Manfredi, and R. Cucchiara. A fast approach for integrating orb descriptors in the bag of words model. In *IS&T/SPIE Electronic Imaging*, pages 866709–866709. International Society for Optics and Photonics, 2013. (page 146).
- M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007. (page 78).
- G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2432–2437. IEEE, 2005. (page 83).
- G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1) :34–46, 2007. (page 83).
- M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science and Business Media, 2012. (page 142).
- K. Grove, J. Atwood, P. Hill, G. Fitch, A. DiFonzo, M. Marchese, and M. Blanco. Commercial motor vehicle driver performance with adaptive cruise control in adverse weather. *Procedia Manufacturing*, 3 :2777–2783, 2015. (page 43).
- J. Guivant and E. Nebot. Improving computational and memory requirements of simultaneous localization and map building algorithms. In *2002. Proceedings. ICRA'02. IEEE International Conference on Robotics and Automation*, volume 3, pages 2731–2736. IEEE, 2002. (page 82, 82).
- J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3) :242–257, 2001. (page 82).
- J. E. Guivant and E. M. Nebot. Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. *IEEE Transactions on Robotics and Automation*, 19(4) :749–755, 2003. (page 82).
- M. Gunther, T. Wiemann, S. Albrecht, and J. Hertzberg. Building semantic object maps from sparse and noisy 3d data. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2228–2233. IEEE, 2013. (page 31).
- P. Gurdjos, P. Sturm, and Y. Wu. Euclidean structure from $n \geq 2$ parallel circles : theory and algorithms. In *Computer Vision—ECCV 2006*, pages 238–252. Springer, 2006. (page 66).
- N. Haala, M. Peter, J. Kremer, and G. Hunter. Mobile lidar mapping for 3d point cloud collection in urban areas—a performance test. *The international archives of the photogrammetry, remote sensing and spatial information sciences*, 37 :1119–1127, 2008. (page 30).
- M. Hachm. Ces 2013 : Audi demonstrates its self-driving car. *Popular Science*, pages 2013–01, 2013. (page 18).
- R. Halterman and M. Bruch. Velodyne hdl-64e lidar for unmanned surface vehicle obstacle detection. In *SPIE Defense, Security, and Sensing*, pages 76920D–76920D. International Society for Optics and Photonics, 2010. (page 31).
- K. Hamada, Z. Hu, M. Fan, and H. Chen. Surround view based parking lot detection and tracking. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1106–1111. IEEE, 2015. (page 17).
- C. Hane, L. Heng, G. H. Lee, A. Sizov, and M. Pollefeys. Real-time direct dense matching on fisheye images using plane-sweeping stereo. In *2nd International Conference on 3D Vision (3DV)*, volume 1, pages 57–64. IEEE, 2014. (page 45).
- C. Hane, T. Sattler, and M. Pollefeys. Obstacle detection for self-driving cars using only monocular cameras and wheel odometry. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5101–5108. IEEE, 2015. (page 45).
- C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15. Citeseer, 1988. (page 97).
- R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. (page 144).
- R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6) : 580–593, 1997. (page 142).
- A. Hedrick, H. Marshall, A. Winstral, K. Elder, S. Yueh, and D. Cline. Independent evaluation of the snodas snow depth product using regional scale lidar-derived measurements. *Cryosphere Discussions*, 2014. (page 30).
- A. M. Hernandez. *Vision dynamique pour la navigation d'un robot mobile*. PhD thesis, Institut National Polytechnique de Toulouse, 2004. (page 8).
- J. Hildermeier and A. Villareal. Two ways of defining sustainable mobility : Autolib' and bemobility. *Journal of Environmental Policy & Planning*, 16(3) :321–336, 2014. (page 14).
- A. B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection : a survey. *Machine Vision and Applications*, 25(3) :727–745, 2014. (page 43).
- H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814. IEEE, 2005. (page 33).

-
- HLDI. Mercedes-benz collision avoidance features : initial results. *Bulletion of Highway Loss Data Institute*, 29(7), April 2012. (page 5).
- HLDI. Honda accord collision avoidance features : initial results. *Bulletion of Highway Loss Data Institute*, 31(2), April 2014. (page 5).
- HLDI. Honda accord collision avoidance features. *Bulletion of Highway Loss Data Institute*, 32(7), April 2015a. (page 5).
- HLDI. 2013–15 honda accord collision avoidance features. *Bulletion of Highway Loss Data Institute*, 32(33), December 2015b. (page 5).
- HLDI. Mazda collision avoidance features. *Bulletion of Highway Loss Data Institute*, 32(22), September 2015c. (page 5).
- HLDI. Subaru collision avoidance features : an update. *Bulletion of Highway Loss Data Institute*, 32(8), April 2015d. (page 5).
- HLDI. Volvo city safety loss experience – a long-term update. *Bulletion of Highway Loss Data Institute*, 32(1), April 2015e. (page 5).
- B. W. Hobson, A. D. Sherman, and P. R. McGill. Imaging and sampling beneath free-drifting icebergs with a remotely operated vehicle. *Deep Sea Research Part II : Topical Studies in Oceanography*, 58(11) :1311–1317, 2011. (page 3).
- K. Hojjati-Emami, B. Dhillon, and K. Jenab. Reliability prediction for the vehicles equipped with advanced driver assistance systems (adas) and passive safety systems (pss). *International Journal of Industrial Engineering Computations*, 3 :731–742, 2012. (page 5).
- Hokuyo. Laser distance measurement devices, 2016. URL <https://www.hokuyo-aut.jp>. (page 31).
- A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *2008. IROS 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3946–3952. IEEE, 2008. (page 78).
- A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *International Symposium on Robotics Research (ISRR)*, volume 2, 2011. (page 79).
- G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. On the complexity and consistency of ukf-based slam. In *ICRA'09. IEEE International Conference on Robotics and Automation*, pages 4401–4408. IEEE, 2009. (page 82).
- H.-P. Huang and S.-Y. Chung. Dynamic visibility graph for path planning. In *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2813–2818. IEEE, 2004. (page 169).
- S. Huang, Z. Wang, and G. Dissanayake. Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 24(5) :1121–1130, 2008. (page 81).
- T. Hunter, P. Abbeel, and A. Bayen. The path inference filter : model-based low-latency map matching of probe vehicle data. *IEEE Transactions on Intelligent Transportation Systems*, 15(2) :507–529, 2014. (page 81).
- IBEO ADAS. Intelligent scanners for smart drivers, 2016. URL <http://www.ibeo-as.com/>. (page 31).
- IEEE Sensors. <http://ieee-sensors2016.org/>, 2016. No citations.
- iRobot. Robots ménagers, 2016. URL <http://www.irobot.fr/>. (page 2, 2).
- B. Jaforte. Recent applications of the clothoid equation in highway design. *Rivista della Strada*, 45(Analytic), 1976. (page 171).
- A. K. Jain. Data clustering : 50 years beyond k-means. *Pattern recognition letters*, 31 :651–666, 2010. (page 115).
- J. A. Jakubiec and C. F. Reinhart. A method for predicting city-wide electricity gains from photovoltaic panels based on lidar and gis data combined with hourly daysim simulations. *Solar Energy*, 93 :127–143, 2013. (pages 38, 77).
- A. A. Janbakhsh, M. B. Khaknejad, and R. Kazemi. Simultaneous vehicle-handling and path-tracking improvement using adaptive dynamic surface control via a steer-by-wire system. *Proceedings of the Institution of Mechanical Engineers, Part D : Journal of automobile engineering*, 227 :345–360, 2013. (page 5).
- H. Jang, S. Cho, and B. Yong. The safety evaluation method of advanced emergency braking system. *Transactions of the Korean Society of Automotive Engineers*, 21(5) :162–168, 2013. (page 43).
- Jean-Pierre Brossard. *Dynamique du véhicule : modélisation des systèmes complexes*. PPUR presses polytechniques, 2006. (pages 86, 86, 88).
- B. S. Jensen, M. B. Skov, and N. Thiruvachandran. Studying driver attention and behaviour for three configurations of gps navigation in real traffic driving. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1271–1280. ACM, 2010. (page 6).
- P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5) :748–760, 2001. (page 77).
- Y. Jeong, D. Nister, D. Steedly, R. Szeliski, and I.-S. Kweon. Pushing the envelope of modern methods for bundle adjustment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8) :1605–1617, 2012. (page 83).
- D. Johnson and A. Ali. Modeling and simulation of landmine and improvised explosive device detection with multiple loops. *The Journal of Defense Modeling and Simulation : Applications, Methodology, Technology*, 12(3) :257–271, 2015. (page 3).
- A. Jokinen, S. Feng, W. Schuster, W. Ochieng, C. Hide, T. Moore, and C. Hill. Glonass aided gps ambiguity fixed precise point positioning. *Journal of Navigation*, 66, 2013. (page 76).
- E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization : A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4) :407–430, 2011. (page 79).
- S. J. Julier. A sparse weight kalman filter approach to simultaneous localisation and map building. In *Proceedings. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1251–1256. IEEE, 2001. (page 82).
- M. Kaess and F. Dellaert. Covariance recovery from a square root information matrix for data association. *Robotics and autonomous systems*, 57(12) :1198–1210, 2009. No citations.
- M. Kaess, A. Ranganathan, and F. Dellaert. isam : Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6) : 1365–1378, 2008. No citations.

-
- K. Kamejima, Y. Hosoda, Y. Nakano, M. Fujie, T. Iwamoto, K. Honma, and T. Yoshida. Navigation apparatus for mobile system, 1986. URL <https://www.google.com/patents/US4628453>. US Patent 4,628,453. (page 81).
- T. Kanade, C. Thorpe, and W. Whittaker. Autonomous land vehicle project at cmu. In *Proceedings of the 1986 ACM fourteenth annual conference on Computer science*, pages 71–80. ACM, 1986. (page 7).
- Y. Kanayama and B. I. Hartman. Smooth local path planning for autonomous vehicles. In *Autonomous robot vehicles*, pages 62–67. Springer, 1990. (page 171).
- H. I. Kang, B. Lee, and K. Kim. Path planning algorithm using the particle swarm optimization and the improved dijkstra algorithm. In *PACIIA'08. Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, volume 2, pages 1002–1004. IEEE, 2008. (page 170).
- O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1) : 90–98, 1986. (page 169).
- K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12 : 1437–1454, 2012. (page 35, 35).
- T. Kim, T. Lee, K. Yi, and J. Lee. Safety assessment procedure for advanced emergency braking system. In *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2015. (page 43).
- R. Kingslake and R. B. Johnson. *Lens design fundamentals*. academic press, 2009. (page 63).
- B. Kirk. Business opportunities in automated vehicles. *Journal of Unmanned Vehicle Systems*, 4, 2016. (page 7).
- A. Kirsch and N. Grinberg. *The factorization method for inverse problems*, volume 36. Oxford University Press Oxford, 2008. (page 28).
- S. G. Klauer, T. A. Dingus, V. L. Neale, J. D. Sudweeks, D. J. Ramsey, et al. The impact of driver inattention on near-crash/crash risk : An analysis using the 100-car naturalistic driving study data. 2006. (page 5).
- G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR 2007. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234. IEEE, 2007. (page 132).
- G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *ISMAR 2009. 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 83–86. IEEE, 2009. (page 132).
- K. Kolev, T. Brox, and D. Cremers. Fast joint estimation of silhouettes and dense 3D geometry from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 :493–505, 2012. (page 32, 32).
- K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. In *Robotics research*, pages 201–212. Springer, 2010. (pages 78, 79).
- V. Kostov, E. Degtiariova-Kostova, et al. The planar motion with bounded derivative of the curvature and its suboptimal paths. *Acta Math. Univ. Comenianae*, 64 :185–226, 1995. (page 171).
- R. Kümmerle, R. Triebel, P. Pfaff, and W. Burgard. Monte carlo localization in outdoor terrains using multilevel surface maps. *Journal of Field Robotics*, 25(6-7) :346–359, 2008. (page 130).
- A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *Computer Vision—ECCV 2014*, pages 703–718. Springer, 2014. (pages 45, 46).
- Z. Kurt-Yavuz and S. Yavuz. A comparison of ekf, ukf, fastslam2.0, and ukf-based fastslam algorithms. In *IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, pages 37–43. IEEE, 2012. (page 82).
- K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1330–1337. IEEE, 2012. (page 46).
- K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3050–3057. IEEE, 2014. (page 46).
- A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers. Robotic surgery : a current perspective. *Annals of surgery*, 239(1) : 14–21, 2004. (page 2).
- J. Langheim. Carsense-new environment sensing for advanced driver assistance systems. In *8th World Congress on Intelligent Transport Systems*, 2001. (page 8).
- D. Lanman, D. Crispell, and G. Taubin. Surround structured lighting for full object scanning. In *2007. 3DIM'07. Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 107–116. IEEE, 2007. (page 34, 34).
- D. Lanman, D. Crispell, and G. Taubin. Surround structured lighting : 3-d scanning with orthographic illumination. *Computer Vision and Image Understanding*, 113 :1107–1117, 2009. (page 34, 34).
- H. d. Largentaye. Velib' : A case of successful co-operation between the city of paris and the private sector in the framework of a sustainable development strategy. In *Presentation for OECD 3rd Annual Meeting on Public Private Partnerships*, 2010. (page 14).
- J.-C. Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012. (page 172).
- S. M. LaValle. Rapidly-exploring random trees : A new tool for path planning. 1998. (page 169).
- S. M. LaValle and J. J. Kuffner Jr. Rapidly-exploring random trees : Progress and prospects. 2000. (page 169).
- Q. V. Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8595–8598. IEEE, 2013. (page 46).
- I. Le Berre and C. Tissot. Systèmes d'information géographique. 1996. (page 77).
- P. Lébraly. *Etalonnage de caméras à champs disjoints et reconstruction 3D : Application à un robot mobile*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2012. (pages 63, 63, 65, 65, 66, 66, 66).
- R. Lenain, B. Thuilot, C. Cariou, and P. Martinet. Adaptive and predictive non linear control for sliding vehicle guidance : Application to trajectory tracking of farm vehicles relying on a single rtk gps. In *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004)*, volume 1, pages 455–460. IEEE, 2004. (page 192).

-
- R. Lenain, B. Thuilot, C. Cariou, and P. Martinet. Mobile robot control in presence of sliding : Application to agricultural vehicle path tracking. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6004–6009. IEEE, 2006. (page 192).
- J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg. *Real-time robot motion planning using rasterizing computer graphics hardware*, volume 24. ACM, 1990. No citations.
- J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3) :376–382, 1991a. (page 81).
- J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1442–1447. Ieee, 1991b. (page 81).
- J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, et al. Towards fully autonomous driving : Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium*, pages 163–168. IEEE, 2011. (pages 10, 38).
- M. Lhuillier and L. Quan. Image interpolation by joint view triangulation. In *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE, 1999. (page 155).
- D. Lill, M. Schappacher, A. Gutjahr, and A. Sikora. Development of a wireless communication and localization system for vru esafety. In *2010 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)*, pages 459–463. IEEE, 2010. (page 42, 42).
- V. Litvinov, M. Lhuillier, and F. Aubière. Incremental solid modeling from sparse and omnidirectional structure-from-motion data. In *BMVC*, 2013. (page 34).
- LMI Technologies. High-definition structured light scanning systems for 3d visualization, 3d reverse engineering and 3d rapid prototyping, 2016. URL <http://lmi3d.com/>. (pages 34, 35).
- H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293 :133–135, 1981. (page 144).
- H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision : Issues, Problems, Principles, and Paradigms*, MA Fischler and O. Firschein, eds, pages 61–62, 1987. (pages 142, 144, 153).
- M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical report, Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece, 2004. (page 83).
- M. A. Lourakis and A. Argyros. SBA : A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36 (1), 2009. (page 132).
- M. I. Lourakis and A. A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1526–1531. IEEE, 2005. (page 83).
- D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 2004. (page 98).
- B. D. Lucas. Generalized image matching by the method of differences. 1985. (page 32).
- B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981. (page 32).
- E. Lucet and A. Micaelli. State and parameters observation for accurate off-road navigation of wheeled mobile robots. In *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 1–6. IEEE, 2015. (page 78).
- E. Lucet, R. Lenain, and C. Grand. Dynamic path tracking control of a vehicle on slippery terrain. *Control Engineering Practice*, 42 : 60–73, 2015. (page 78).
- H. Maaref, M. Oussalah, and C. Barreti. Fusion de données capteurs en vue de la localisation absolue d'un robot mobile par une méthode basée sur la théorie des possibilités. comparaison avec le filtre de kalman. *Traitement du Signal*, 16(5), 1999. (page 78).
- J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. (page 115).
- W. Maddern, G. Pascoe, and P. Newman. Leveraging experience for large-scale lidar localisation in changing cities. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1684–1691. IEEE, 2015. (page 12, 12).
- V. N. Mahajan. Optical imaging and aberrations : Part i : Ray geometrical optics. SPIE Bellingham, 1998. (page 63).
- M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24 (3) :169–186, 2007. (page 78).
- F. Malartre. *Perception intelligente pour la navigation rapide de robots mobiles en environnement naturel*. Theses, Université Blaise Pascal - Clermont-Ferrand II, June 2011. (page 77).
- D. A. Mann, Z. Lu, M. C. Hastings, and A. N. Popper. Detection of ultrasonic tones and simulated dolphin echolocation clicks by a teleost fish, the american shad (*alosa sapidissima*). *The Journal of the Acoustical Society of America*, 104 :562–568, 1998. (page 49).
- D. Maroteaux, D. Le Guen, and E. Chauvelier. Development of a fuel economy and co 2 simulation platform for hybrid electric vehicles-application to renault eolab prototype. Technical report, SAE Technical Paper, 2015. (page 19).
- D. A. Márquez-Gámez. *Towards visual navigation in dynamic and unknown environment : trajectory learning and following, with detection and tracking of moving objects*. PhD thesis, Toulouse, INSA, 2012. (page 8).
- J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22 :761–767, 2004. (page 98).
- J. C. Maxwell. A dynamical theory of the electromagnetic field. *Proceedings of the Royal Society of London*, 13 :531–536, 1863. (page 28).
- J. C. Maxwell. *A treatise on electricity and magnetism*, volume 1. Clarendon press, 1881. (page 28).
- P. S. Maybeck. *Stochastic models, estimation, and control*, volume 3. Academic press, 1982. No citations.

-
- A. S. McLeod. Sea-me-we 4 fibre optic submarine cable project. *Proto-Type*, 1, 2013. (page 3).
- L. Meirer, R. Koch, and C. Bellaflor. Device and method for automatically laterally storing a motor vehicle in a storing device, Oct. 24 2013. URL <https://www.google.com/patents/US20130280018>. US Patent App. 13/989,911. (page 16, 16).
- K. Melbouci, S. N. Collette, V. Gay-Bellile, O. Ait, M. C. Aider, and M. Dhome. Ajustement de faisceaux du slam revisité en utilisant un capteur rgb-d. In *Journées francophones des jeunes chercheurs en vision par ordinateur*, 2015. (page 83).
- K. Mierle. *Open source 3D reconstruction from video*. PhD thesis, University of Toronto, 2008. (page 132).
- K. Mierle and W. J. Maclean. Evaluating multiview reconstruction. In *Computer Vision Systems*, pages 228–237. Springer, 2008. (page 132).
- K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1) : 63–86, 2004. (page 97).
- K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, 2005. (page 98).
- A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning vocabularies over a fine quantization. *International journal of computer vision*, 103(1) :163–175, 2013. (page 114).
- M. Milford and G. Wyeth. Persistent navigation and mapping using a biologically inspired slam system. *The International Journal of Robotics Research*, 29(9) :1131–1153, 2010. No citations.
- M. J. Milford and G. F. Wyeth. Seqslam : Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation*, pages 1643–1649. IEEE, 2012. (pages 84, 84, 116).
- N. Minois-Enache and M. Chirca. Dispositif d’aide à la conduite pour estimation du danger d’une situation, 2015a. FR Patent App. 15-59,022. (page 23).
- N. Minois-Enache and M. Chirca. Système de commande de la direction d’un véhicule automobile en cas de collision imminente avec un obstacle, 2015b. FR Patent App. 15-59,256. (page 23).
- K. Mitsu and J. Miura. Specific person detection and tracking by a mobile robot using 3d lidar and espar antenna. In *Intelligent Autonomous Systems 13*, pages 705–719. Springer, 2016. (page 31).
- MobileEye. Artificial Vision Technology. <http://www.mobileye.com/technology/>, 2016. (page 26).
- A. Mogelmoose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems : Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4) :1484–1497, 2012. (page 26).
- M.-O. Monod, R. Chapuis, P. Gosset, R. Rouveure, D. Vivet, F. Gérossier, P. Faure, P. Checchin, L. Moiroux, P. Guérin, et al. Radar panoramique hyperfréquence pour la localisation et la cartographie simultanées en environnement extérieur. *TS. Traitement du signal*, 29(6) :463–492, 2012. No citations.
- M. Montemerlo and S. Thrun. Fastslam 2.0. *FastSLAM : A scalable method for the simultaneous localization and mapping problem in robotics*, pages 63–90, 2007. (page 83).
- M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, et al. Fastslam : A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*, pages 593–598, 2002. (page 83).
- M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, et al. Junior : The stanford entry in the urban challenge. *Journal of field Robotics*, 25(9) :569–597, 2008. (page 45).
- F. Moosmann and C. Stiller. Velodyne slam. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 393–398. IEEE, 2011. (page 31).
- J. Moras, V. Cherfaoui, and P. Bonnifait. A lidar perception scheme for intelligent vehicle navigation. In *2010 11th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1809–1814. IEEE, 2010. (pages 30, 31).
- J. Moras, V. Cherfaoui, and P. Bonnifait. Credibilist occupancy grids for vehicle perception in dynamic environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 84–89. IEEE, 2011. (page 31).
- J. J. Moré. The levenberg-marquardt algorithm : implementation and theory. In *Numerical analysis*. Springer, 1978. (page 148).
- P. Morin and C. Samson. Time-varying exponential stabilization of chained form systems based on a backstepping technique. In *Proceedings of the 35th IEEE Conference on Decision and Control, 1996*, volume 2, pages 1449–1454. IEEE, 1996. No citations.
- E. M. Mouaddib, O. Labbani-Igbida, and D. Nuzillard. Odométrie visuelle pour un robot mobile en environnement industriel. 2007. (page 79).
- H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard. Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation. In *19th IEEE International Conference on Intelligent Transportation Systems Conference (ITSC 2016)*, 2016. (page 172).
- P. Moulon, P. Monasse, and R. Marlet. Openmvg (open multiple view geometry), 2012. (pages 32, 132).
- A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572. IEEE, 2007. (page 79).
- B. Multon. L’énergie sur la terre : analyse des ressources et de la consommation. la place de l’énergie électrique. *revue 3EI*, 1998. (page 3).
- R. Mur-Artal, J. Montiel, and J. D. Tardos. Orb-slam : a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5) :1147–1163, 2015. (page 84).
- R. Nair, E. Miller-Hooks, R. C. Hampshire, and A. Bušić. Large-scale vehicle sharing systems : analysis of vélib’. *International Journal of Sustainable Transportation*, 7(1) :85–106, 2013. (page 14).
- P. K. Nakrani. *Smart Car Technologies : A Comprehensive Study of the State of the Art with Analysis and Trends*. Arizona State University, 2015. (page 7).
- R. Negenborn. *Robot localization and Kalman filters*. PhD thesis, Utrecht University, 2003. (page 74).

-
- E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis. C-klam : Constrained keyframe-based localization and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3638–3643. IEEE, 2014. (page 83).
- M. C. Neves and E. Oliveira. Arcos-an autonomous mobile robot control system. In *Proc. ICSC Int. Symp. on Engineering of Intelligent Systems*, pages 220–227, 1998. No citations.
- R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam : Dense tracking and mapping in real-time. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2320–2327. IEEE, 2011. (page 132).
- NHTSA. Google - compiled response to 12 nov 15 interp request – 4 feb 16 final, 2016. URL <http://isearch.nhtsa.gov/>. (page 10).
- J. I. Nieto, J. E. Guivant, and E. M. Nebot. The hybrid metric maps (hymms) : A novel map representation for denseslam. In *Proceedings. ICRA'04. 2004 IEEE International Conference on Robotics and Automation*, volume 1, pages 391–396. IEEE, 2004. (page 131, 131).
- D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. IEEE, 2006. (page 115).
- D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1–652. IEEE, 2004. (page 78).
- D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1) :3–20, 2006. (page 78).
- A. Nüchter, K. Lingemann, and J. Hertzberg. 6d slam with kurt3d. *Robotic 3D Environment Cognition*, page 59, 2006. (page 30, 30).
- R. Odolinski, P. J. Teunissen, and D. Odijk. Combined bds, galileo, qzss and gps single-frequency rtk. *GPS Solutions*, pages 151–163, 2015. (page 76).
- J. S. Ortega. *Towards visual localization, mapping and moving objects tracking by a mobile robot : a geometric and probabilistic approach*. PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2007. (page 99).
- V. Osa, J. Matamalas, J. F. Monserrat, and J. López. Localization in wireless networks : the potential of triangulation techniques. *Wireless personal communications*, 68(4) :1525–1538, 2013. (page 77).
- R. Pajarola. Large scale terrain visualization using the restricted quadtree triangulation. In *Visualization'98. Proceedings*, pages 19–26. IEEE, 1998. (page 140).
- M. Parent. Advanced urban transport : Automation is on the way. *IEEE Intelligent Systems*, 22(2) :9–11, 2007. (page 14).
- M. Parent. Cybears : the first fully automated urban vehicles on the market, 2014. URL <http://www.v-charge.eu/wp-content/uploads/2014/07/ETH-2014-Parent.pdf>. No citations.
- R. Paul and P. Newman. Fab-map 3d : Topological mapping with spatial and visual appearance. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2649–2656. IEEE, 2010. (pages 84, 116).
- L. Paull, S. Saedi, M. Seto, and H. Li. Auv navigation and localization : A review. *IEEE Journal of Oceanic Engineering*, 39 : 131–149, 2014. (page 3).
- L. M. Paz, J. D. Tardós, and J. Neira. Divide and conquer : EKF slam in $O(n)$. *IEEE Transactions on Robotics*, 24(5) :1107–1120, 2008. (page 82).
- J. Paziewski and P. Wielgosz. Accounting for galileo–gps inter-system biases in precise satellite positioning. *Journal of Geodesy*, 89 : 81–93, 2015. (page 76).
- PEAK-System. PCAN-USB pro : PEAK-system, 2014. URL <http://www.peak-system.com/PCAN-USB-Pro.200.0.html?&L=2.00000>. (page 52).
- S. V. Pemmaraju and C. A. Shaffer. Analysis of the worst case space complexity of a pr quadtree. *Information processing letters*, 49 : 263–267, 1994. (page 140).
- X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition : Comprehensive study and good practice. *Computer Vision and Image Understanding*, 2016. (page 114).
- L. M. P. Pérez. *Divide and Conquer : EKF SLAM in $O(n)$* . PhD thesis, Ph. D. dissertation, Univ. of Zaragoza, Zaragoza, 2008. (page 82).
- A. Petrovai, R. Danescu, and S. Nedevschi. A stereovision based approach for detecting and tracking lane and forward obstacles on mobile devices. In *Intelligent Vehicles Symposium (IV)*, 2015 IEEE, pages 634–641. IEEE, 2015. (page 27).
- P. Pfaff, R. Triebel, and W. Burgard. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *The International Journal of Robotics Research*, 26(2) :217–230, 2007. (page 130, 130).
- D. Pfeiffer and U. Franke. Efficient representation of traffic scenes by means of dynamic stixels. In *2010 IEEE Intelligent Vehicles Symposium (IV)*, pages 217–224. IEEE, 2010. (page 33, 33).
- D. Pfeiffer and U. Franke. Towards a global optimal multi-layer stixel representation of dense 3d data. In *BMVC*, pages 1–12, 2011. (page 33, 33).
- M. Pierrot-Deseilligny and N. Paparoditis. A multiresolution and optimization-based image matching approach : An application to surface reconstruction from spot5-hrs stereo imagery. *Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(1/W41), 2006. (page 132).
- A. M. Pinto, A. P. Moreira, and P. G. Costa. A localization method based on map-matching and particle swarm optimization. *Journal of Intelligent & Robotic Systems*, 77(2) :313–326, 2015. (page 81).
- C. Pinto. How autonomous vehicle policy in california and nevada addresses technological and non-technological liabilities. *Intersect : The Stanford Journal of Science, Technology and Society*, 5, 2012. (page 10).
- M. Poorsartep and T. Stephens. Truck automation opportunities. In *Road Vehicle Automation 2*, pages 173–185. Springer, 2015. (page 42).
- H. Pornon. *SIG : la dimension géographique du système d'information*. Dunod, 2015. (page 77).

-
- P. Praveen Kumar Peter. Truck driver usage of main driving controls. 2013. (page 13, 13).
- PSA Paris-Bordeaux. Un prototype psa parcourt le trajet paris-bordeaux en mode autonome, 2016. URL <http://media.groupe-psa.com/fr/psa-peugeot-citroen/communiqués-de-presse/innovation-technologie/un-prototype-psa-parcourt-le-trajet>. (page 12, 12).
- X. Qi, C.-G. Li, G. Zhao, X. Hong, and M. Pietikäinen. Dynamic texture and scene classification by transferring deep image features. *Neurocomputing*, 171 :1230–1241, 2016. (page 27).
- Quanergy. Future of 3d sensing and perception. 3d lidar for adas, autonomous vehicles & 3d mapping, 2016. URL <http://www.quanergy.com/>. (page 31).
- P. Quelhas and J.-M. Odobez. Multi-level local descriptor quantization for bag-of-features image representation. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 242–249. ACM, 2007. (page 114).
- Y. Quemener. 2rt3d - rapport des expérimentations sur données simulées. Technical report, 2010. (pages 129, 130).
- R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. *Computer Vision—ECCV 2008*, pages 500–513, 2008. (pages 142, 155).
- S. A. Rahok, Y. Shikanai, and K. Ozaki. Navigation using an environmental magnetic field for outdoor autonomous mobile robots. *Advanced Robotics*, 25(13-14) :1751–1771, 2011. (page 75).
- M. Raibert. Dynamic legged robots for rough terrain. In *2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2010. (page 4).
- M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and T. Team. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, volume 17, pages 10822–10825, 2008. (pages 3, 4).
- D. Ramadasan. *SLAM temporel à contraintes multiples*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2015. (page 83).
- D. Ramadasan, M. Chevaldonné, and T. Chateau. Slam constraint en environnement de grande taille. In *Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014*, 2014. No citations.
- D. Ramadasan, T. Chateau, and M. Chevaldonne. Dcslam : A dynamically constrained real-time slam. In *IEEE International Conference on Image Processing (ICIP)*, pages 1130–1134. IEEE, 2015. (page 83).
- Rapport Accident Google Car. Rapport accident google car, 2016. URL <http://dmv.ca.gov/portal/wcm/connect/3946fbb8-e04e-4d52-8f80-b33948df34b2/Google+Auto+LLC+02.14.16.pdf?MOD=AJPERES>. (page 11).
- U. Raschke and J. Borenstein. A comparison of grid-type map-building techniques by index of performance. In *1990 IEEE International Conference on Robotics and Automation*, pages 1828–1832. IEEE, 1990. (page 129).
- J. Reeds and L. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific journal of mathematics*, 145(2) : 367–393, 1990. (page 170).
- M. Reichel and M. E. Bouzouraa. Driver assistance system and method for authorizing an autonomous or piloted garage parking, Oct. 13 2015. US Patent 9,156,497. (page 18).
- P. Renaud. Accès transports viables, 2016. URL <http://http://transportsviables.org/question-despace/>. (page 14).
- Renault Next-Two. Renault next two : Our take on the autonomous and connected car, 2016. URL <https://group.renault.com/en/passion-2/innovation/renault-a-born-innovator/autonomous-and-connected-car/>. (page 13, 13).
- Rholf. Parking steering assistance system and method for operating a steering assistance system patent, 2007. (page 44).
- B. Richards, M. Gan, J. Dayton, J. Quintana, J. Liu, and M. Enriquez. Obstacle avoidance system for uavs using computer vision. 2014. (page 27).
- T. D. Ringler, R. P. Heikes, and D. A. Randall. Modeling the atmospheric general circulation using a spherical geodesic grid : A new class of dynamical cores. *Monthly Weather Review*, 128(7) :2471–2490, 2000. (page 41).
- B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter : Particle filters for tracking applications*, volume 685. Artech house Boston, 2004. (page 102).
- G. Roberts, L. Gomes, J. Brenguier, A. Apituley, K. Wilson, R. Boers, D. Donovan, J. Pelon, D. Josset, M. Boquet, et al. Assessing aerosol-cloud interactions linking multi-platform observations and remote sensing. In *AGU Fall Meeting Abstracts*, volume 1, page 0320, 2009. (page 30).
- S. D. Roth. Ray casting for modeling solids. *Computer graphics and image processing*, 18(2) :109–144, 1982. (page 129).
- E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb : an efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011. (page 146).
- S. Saedi, L. Paull, M. Trentini, and H. Li. Occupancy grid map merging for multiple robot simultaneous localization and mapping. *International Journal of Robotics and Automation*, 30(2) :149–157, 2015. No citations.
- K. Sahr, D. White, and A. J. Kimerling. Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, 30(2) :121–134, 2003. (page 41).
- T. Salgado-Jimenez, J. Gonzalez-Lopez, L. Martinez-Soto, E. Olguin-Lopez, P. Resendiz-Gonzalez, and M. Bandala-Sanchez. Deep water rov design for the mexican oil industry. In *OCEANS 2010 IEEE-Sydney*, pages 1–6. IEEE, 2010. (page 3).
- Salvador. Vehicle park assist with steering centering, 2011. (page 44).
- H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16 :187–260, 1984. (page 130).
- C. Samson, P. Morin, and R. Lenain. Modeling and control of wheeled mobile robots. In *Springer Handbook of Robotics*, pages 1235–1266. Springer, 2016. (page 192).
- E. G. Sarabia, J. R. Llata, S. Robla, C. Torre-Ferrero, and J. P. Oria. Accurate estimation of airborne ultrasonic time-of-flight for overlapping echoes. *Sensors*, 13, 2013. (page 49).
- SARTRE Project. Safe road trains for the environment, 2016. URL <http://www.sartre-project.eu/>. No citations.
- T. Sato, Y. Kaneda, and M. Hikita. Ultrasonic precise distance measurement and movement detection installed in sensor network. In *2014 IEEE International Ultrasonics Symposium*, 2014. (page 49).

-
- T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2102–2110, 2015. (page 32, 32).
- D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *Robotics & Automation Magazine, IEEE*, 18(4) :80–92, 2011. (page 79).
- A. Scheuer. *Planification de chemins à courbure continue pour robot mobile non-holonyme*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1998. (pages 171, 172, 174, 174, 176, 186).
- C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2) : 151–172, 2000. (page 98).
- J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. *CVPR*, 2016. (page 132).
- D. Schrank, B. Eisele, and T. Lomax. Tti’s 2012 urban mobility report. *Texas A&M Transportation Institute. The Texas A&M University System*, 2012. (page 4).
- J. T. Schwartz and M. Sharir. On the “piano movers” problem i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on pure and applied mathematics*, 36(3) :345–398, 1983a. (page 169).
- J. T. Schwartz and M. Sharir. On the “piano movers” problem. ii. general techniques for computing topological properties of real algebraic manifolds. *Advances in applied Mathematics*, 4(3) :298–351, 1983b. (page 169).
- B. Schwarz. Mapping the world in 3d. *Nat. Photonics*, 4 :429–430, 2010. (page 31).
- J. H. Seinfeld and S. N. Pandis. *Atmospheric chemistry and physics : from air pollution to climate change*. John Wiley & Sons, 2012. (page 3).
- P. Seiniger, A. Hellmann, O. Bartels, M. Wisch, and J. Gail. Test procedures and results for pedestrian aeb systems. In *24th Conference on the Enhancement of the Safety of Vehicles (ESV)*, 2015. (page 5).
- Y.-W. Seo. Generating omni-directional view of neighboring objects for ensuring safe urban driving, 2014. (page 45).
- J. R. Shewchuk. Triangle : Engineering a 2d quality mesh generator and delaunay triangulator. In *Applied computational geometry towards geometric engineering*, pages 203–222. Springer, 1996. (page 154).
- J. Shi and C. Tomasi. Good features to track. In *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994. (page 97).
- T. Shi and S. Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 2012. (page 46).
- S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis. Cooperative adaptive cruise control (cacc) definitions and operating concepts. In *Proceedings of the 94th Annual TRB Meeting*, 2015. (page 42).
- SICK. Sensor intelligence, 2016. URL <http://www.sick.com>. (pages 30, 31).
- A. Singhal. Issues in autonomous mobile robot navigation. *Computer Science Dept, U. of Rochester*, 1997. (pages 74, 80).
- J. Sivic and A. Zisserman. Video google : A text retrieval approach to object matching in videos. In *Proceedings. Ninth IEEE International Conference on Computer Vision*, pages 1470–1477. IEEE, 2003. (page 115).
- J.-J. E. Slotine, W. Li, et al. *Applied nonlinear control*, volume 199. prentice-Hall Englewood Cliffs, NJ, 1991. No citations.
- D. Smith. Beyond the cave : Lascaux and the prehistoric in post-war french culture. *French studies*, 58(2) :219–232, 2004. (page 37).
- R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 850–850. IEEE, 1987. (page 81).
- R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4) :56–68, 1986. (page 81).
- S. M. Smith and J. M. Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1) : 45–78, 1997. (page 98).
- N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism : exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. (page 132, 132).
- N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80 :189–210, 2008. (page 132).
- P. Soille. *Morphological image analysis : principles and applications*. Springer Science & Business Media, 2013. (page 66).
- S. Song and M. Chandraker. Robust scale estimation in real-time monocular sfm for autonomous driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. (page 32).
- P. Soueres and J.-P. Laumond. Shortest paths synthesis for a car-like robot. *IEEE Transactions on Automatic Control*, 41(5) :672–688, 1996. (page 170).
- M. Specht. The evaluation of the positioning accuracy of the egnos and dgps systems based on the long-term measurements in the years 2006-2014. *Polish Cartographical Review*, 47 :99–108, 2015. (page 76).
- SQL Server 2008 R2. Vue d’ensemble de l’indexation spatiale. [https://technet.microsoft.com/fr-fr/library/bb964712\(v=sql.105\).aspx](https://technet.microsoft.com/fr-fr/library/bb964712(v=sql.105).aspx), 2008. (page 40, 40, 40).
- D. M. Stewart. New technology and the law of armed conflict. *Int’l L. Stud. Ser. US Naval War Col.*, 87 :271, 2011. (page 9, 9).
- J. Stolaroff. The need for a life cycle assessment of drone-based commercial package delivery. *Livermore, CA : Lawrence Livermore National Laboratory*, 2014. (page 3, 3).
- U. Störkel, K. Vodopyanov, and W. Grill. Ghz ultrasound wave packets in water generated by an er laser. *Journal of Physics D : Applied Physics*, 31(18) :2258, 1998. (page 49).
- P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(1–2) :1–183, 2011. (page 63).

-
- N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel. Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In *Autonome Mobile Systeme 2005*, pages 157–163. Springer, 2006. (page 79).
- G. L. H. J. Sussmann. Approach to motion planning. *Nonholonomic Motion Planning*, 192 :235, 2012. (page 170).
- C. Sweeney, T. Hollerer, and M. Turk. Theia : A fast and scalable structure-from-motion library. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, pages 693–696. ACM, 2015. (page 132).
- R. Szeliski. Rapid octree construction from image sequences. *CVGIP : Image understanding*, 58 :23–32, 1993. (page 130).
- A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto. Local path planning and motion control for agv in positioning. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS'89. Proceedings*, pages 392–397. IEEE, 1989. (page 171).
- R. Talluri and J. K. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. *IEEE Transactions on Robotics and Automation*, 8(5) :573–584, 1992. (page 80).
- M. Tamazousti. *L'ajustement de faisceaux contraint comme cadre d'unification des méthodes de localisation : application à la réalité augmentée sur des objets 3D*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2013. (page 83).
- J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz. A new approach to vision-aided inertial navigation. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4161–4168. IEEE, 2010a. (page 79).
- J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz. Vision-aided inertial navigation for power line inspection. In *Proc. International Conference on Applied Robotics for the Power Industry, Montreal, Canada*, pages 1–6, 2010b. (page 79).
- J.-P. Tarel. Calibration de caméra fondée sur les ellipses. 1994. (page 66).
- O. Tekdas and V. Isler. Sensor placement for triangulation-based localization. *IEEE Transactions on Automation Science and Engineering*, 7(3) :681–685, 2010. (page 77).
- C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer. Vision and navigation for the carnegie-mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3) :362–373, 1988. (page 7).
- C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. *Vision and Navigation for the Carnegie Mellon Navlab*. Springer, 1989. (page 7).
- S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99 :21–71, 1998. (page 39).
- S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2) :111–127, 2003. (page 129, 129).
- S. Thrun and Y. Liu. Multi-robot slam with sparse extended information filters. In *Robotics Research. The Eleventh International Symposium*, pages 254–266. Springer, 2005. (page 83).
- S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekirk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the darpa grand challenge. *Journal of Field Robotics*, 2006a. accepted for publication. (page 38).
- S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley : The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9) :661–692, 2006b. (page 8).
- F. Tian and P. Shi. Image mosaic using orb descriptor and improved blending algorithm. In *7th International Congress on Image and Signal Processing (CISP)*, pages 693–698. IEEE, 2014. (page 146).
- R. Timofte, K. Zimmermann, and L. Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. *Machine Vision and Applications*, 25(3) :633–647, 2014. (page 26).
- D. J. Todd. *Fundamentals of robot technology : An introduction to industrial robots, teleoperators and robot vehicles*. Springer Science & Business Media, 2012. (page 2).
- E. C. Tolman. Cognitive maps in rats and men. *Psychological review*, 55 :189, 1948. (page 38).
- C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991. (page 97).
- D.-C. Tseng, C.-C. Huang, and T.-W. Chao. Vision-based parking guidance with adaptive isometric transformation. *Sensors & Transducers*, 25 :44, 2013. (page 45).
- S. Tsugawa, T. Yatabe, T. Hirose, and S. Matsumoto. An automobile with artificial intelligence. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 2*, pages 893–895. Morgan Kaufmann Publishers Inc., 1979. (page 7).
- S. Tsugawa, S. Kato, and K. Aoki. An automated truck platoon for energy saving. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4109–4114. IEEE, 2011. (page 4).
- G. Tuna, K. Gulez, V. C. Gungor, and T. V. Mumcu. Evaluations of different simultaneous localization and mapping (slam) algorithms. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pages 2693–2698. IEEE, 2012. (page 82).
- M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra. Vits-a vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3) :342–361, 1988. (page 7).
- L. Ulrich. Top 10 tech cars : slenderized. *Spectrum, IEEE*, 50(4) :34–41, 2013. (page 5).
- C. Urmson, J. A. Bagnell, C. R. Baker, M. Hebert, A. Kelly, R. Rajkumar, P. E. Rybski, S. Scherer, R. Simmons, S. Singh, et al. Tartan racing : A multi-modal approach to the darpa urban challenge. 2007. (page 45).
- V-Charge, ETH Zurich - Autonomous Systems Lab, ETH Zurich - Computer Vision and Geometry Lab, Volkswagen AG, University of Oxford, Università degli Studi di Parma, Technische Universität Braunschweig, and Robert Bosch GmbH. Driverless parking and charging for emobility, 2011-2016. URL <http://www.v-charge.eu>. (page 18, 18).
- Velodyne LiDAR. The most advanced lidar sensors in the market, 2016. URL <http://www.velodynelidar.com/>. (page 31, 31).
- J. Vetelino and A. Reghu. *Introduction to sensors*. CRC Press, 2010. (page 27).
- D. G. Viswanathan. Features from accelerated segment test (fast), 2009. (page 97).

-
- D. Vivet. *Perception de l'environnement par radar hyperfréquence. Application à la localisation et la cartographie simultanées, à la détection et au suivi d'objets mobiles en milieu extérieur*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2011. No citations.
- F. Von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche. Driving with tentacles : Integral structures for sensing and motion. *Journal of Field Robotics*, 25(9) :640–673, 2008. (page 172).
- H. Vorobieva. *La conception et méthode de validation de lois de contrôle pour des systèmes de conduite automatisée du véhicule*. Theses, Université d'Evry Val d'Essone, 2014. (pages 92, 171, 172, 173, 173, 174, 175, 175, 176, 176, 186, 186, 187, 192).
- M. Walter, R. Eustice, and J. Leonard. A provably consistent method for imposing sparsity in feature-based slam information filters. In *Robotics Research*, pages 214–234. Springer, 2007. (page 83).
- F. M. Wanlass. Low stand-by power complementary field effect circuitry, Dec. 5 1967. US Patent 3,356,858. (page 55).
- J. G. Webster and H. Eren. *Measurement, Instrumentation, and Sensors Handbook : Spatial, Mechanical, Thermal, and Radiation Measurement*, volume 1. CRC press, 2014. (page 28).
- S. E. Webster, J. M. Walls, L. L. Whitcomb, and R. M. Eustice. Decentralized extended information filter for single-beacon cooperative acoustic navigation : Theory and experiments. *IEEE Transactions on Robotics*, 29(4) :957–974, 2013. (page 83).
- W. T. Welford. *Aberrations of optical systems*. CRC Press, 1986. (page 63).
- J. Weng, T. S. Huang, and N. Ahuja. *Motion and structure from image sequences*, volume 29. Springer Science & Business Media, 2013. (page 45).
- C. Wheatstone. On binocular vision : and on the stereoscope, an instrument for illustrating its phenomena. *Report to the British Association for the Advancement of Science Pt*, pages 16–17, 1838. (page 32).
- C. Wheatstone. The bakerian lecture : contributions to the physiology of vision. part the second. on some remarkable, and hitherto unobserved, phenomena of binocular vision (continued). *Philosophical Transactions of the Royal Society of London*, 142 :1–17, 1852. (page 32).
- W. Wierwille, R. Hanowski, J. Hankey, C. Kieliszewski, S. Lee, A. Medina, A. Keisler, and T. Dingus. Identification and evaluation of driver errors : Overview and recommendations. Technical report, 2002. (page 5).
- Wigle. All the networks. found by everyone, 2016. URL <https://wigle.net/>. (page 77).
- B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. An image-to-map loop closing method for monocular slam. In *IROS 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2053–2059. IEEE, 2008. (page 84, 84).
- C. Witzgall, G. S. Cheok, and D. E. Gilsinn. Terrain characterization from ground-based lidar. Technical report, DTIC Document, 2003. (page 131, 131).
- O. J. Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14 :15, 2007. (page 78).
- S. Woop, J. Schmittler, and P. Slusallek. Rpu : a programmable ray processing unit for realtime ray tracing. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 434–444. ACM, 2005. (page 129).
- C. Wu. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision-3DV*, pages 127–134. IEEE, 2013. (page 83).
- C. Wu, I. Lenz, and A. Saxena. Hierarchical semantic labeling for task-relevant rgb-d perception. In *Robotics : Science and systems (RSS)*, 2014. (page 35, 35).
- S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath. Collision sensing by stereo vision and radar sensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 10(4) :606–614, 2009. (page 36).
- K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap : A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2, 2010. (pages 130, 131).
- X. Xinjilefu, S. Feng, W. Huang, and C. G. Atkeson. Decoupled state estimation for humanoids using full-body dynamics. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 195–201. IEEE, 2014. (page 4).
- J. Xu, G. Chen, and M. Xie. Vision-guided automatic parking for smart car. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 725–730, 2000. (page 45).
- B. Xue, M. Zhang, and W. N. Browne. Particle swarm optimization for feature selection in classification : a multi-objective approach. *IEEE transactions on cybernetics*, 43(6) :1656–1671, 2013. (page 222).
- H. Yamasaki. *Intelligent sensors*. Elsevier, 1996. (page 28).
- J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206. ACM, 2007. (page 114).
- Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3) :492–504, 2009. (page 115).
- D. Yazdani, B. Nasiri, A. Sepas-Moghaddam, and M. R. Meybodi. A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. *Applied Soft Computing*, 13(4) :2144–2158, 2013. (page 222).
- C. Ye and J. Borenstein. A method for mobile robot navigation on rough terrain. In *Proceedings. ICRA'04. 2004 IEEE International Conference on Robotics and Automation*, volume 4, pages 3863–3869. IEEE, 2004. (page 129).
- X. Ying and Z. Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *Computer Vision-ECCV 2004*, pages 442–455. Springer, 2004. (pages 64, 65).
- K. L. Young and P. M. Salmon. Examining the relationship between driver distraction and driving errors : A discussion of theory, studies and methods. *Safety science*, 50(2) :165–174, 2012. (page 5).

-
- T. Yuan and B. Wang. Accelerometer-based chinese traffic police gesture recognition system. *Chinese Journal of Electronics*, 19(2) : 270–274, 2010. (page 26).
- L. Zalud, P. Kocmanova, F. Burian, T. Jilek, P. Kalvoda, and L. Kopečný. Calibration and evaluation of parameters in a 3d proximity rotating scanner. *Elektronika ir Elektrotechnika*, 21 :3–12, 2015. (page 31).
- J. Zhang and S. Singh. Visual-lidar odometry and mapping : Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015. (page 79).
- J. Zhang, M. Kaess, and S. Singh. Real-time depth enhanced monocular odometry. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 4973–4980. IEEE, 2014. (page 79).
- M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8) :1819–1837, 2014. (page 46).
- S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. How far are we from solving pedestrian detection? *arXiv preprint arXiv :1602.01237*, 2016. (page 44).
- Y. Zhang, L. Ma, and R. Zhang. A quick image registration algorithm based on delaunay triangulation. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(2) :761–773, 2013. (page 155).
- Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 78(1) :87–119, 1995. No citations.
- L. Zhao, S. Huang, L. Yan, and G. Dissanayake. Parallax angle parametrization for monocular slam. In *IEEE International Conference on Robotics and Automation*, pages 3117–3124. IEEE, 2011. No citations.
- L. Zhao, S. Huang, and G. Dissanayake. Linear slam : A linear solution to the feature-based and pose graph slam based on submap joining. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 24–30. IEEE, 2013. (page 81, 81).
- L. Zhao, S. Huang, Y. Sun, L. Yan, and G. Dissanayake. Parallaxba : bundle adjustment using parallax angle feature parametrization. *The International Journal of Robotics Research*, 34(4-5) :493–516, 2015. No citations.
- W.-j. Zhao, S.-r. Gong, C.-p. Liu, and X.-j. SHEN. Adaptive harris corner detection algorithm. *Computer Engineering*, 10(5) :212–215, 2008. (page 97).
- K. Zhou, Q. Hou, R. Wang, and B. Guo. Real-time kd-tree construction on graphics hardware. *ACM Transactions on Graphics (TOG)*, 27 :126, 2008. (page 115).
- S. Zhou. *Dynamic incentive scheme for rental vehicle fleet management*. PhD thesis, Massachusetts Institute of Technology, 2012. (page 14).
- J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, et al. Making bertha drive — an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6, 2014. (pages 10, 10, 33).