



HAL
open science

Perception intelligente pour la navigation rapide de robots mobiles en environnement naturel

Florent Malartre

► **To cite this version:**

Florent Malartre. Perception intelligente pour la navigation rapide de robots mobiles en environnement naturel. Autre. Université Blaise Pascal - Clermont-Ferrand II, 2011. Français. NNT : 2011CLF22132 . tel-00673435

HAL Id: tel-00673435

<https://theses.hal.science/tel-00673435>

Submitted on 23 Feb 2012

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 2132

EDSPIC : 524

UNIVERSITÉ BLAISE PASCAL - CLERMONT II

ÉCOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR DE
CLERMONT-FERRAND

T H È S E

pour obtenir le titre de

Docteur d'université

Spécialité : VISION POUR LA ROBOTIQUE

Présentée par

Florent MALARTRE

**Perception intelligente pour la
navigation rapide de robots mobiles
en environnement naturel**

Thèse dirigée par Roland CHAPUIS

Soutenue publiquement le 16 Juin 2011 devant le jury :

M. Jean-Pierre Derutin	Professeur à l'univ. Blaise Pascal	Président du jury
M. Simon Lacroix	Directeur de Recherche au LAAS	Rapporteur
M. Karsten Berns	Professeur à l'Université de Kaiserslautern (Allemagne)	Rapporteur
M. Christophe Debain	Chargé de Recherche Cemagref Clermont	Examineur
M. Aurélien Godin	Architecte à la DGA	Examineur
M. Roland Chapuis	Professeur à l'univ. Blaise Pascal	Directeur de Thèse

Remerciements

Les travaux présentés dans cette thèse ont été réalisés dans le cadre d'une collaboration entre le Lasmea et le Cemagref de Clermont-Ferrand. Je tiens donc à remercier ces deux organismes qui m'ont permis de réaliser cette thèse dans les meilleures conditions. Je remercie tout particulièrement Michel Dhome, directeur du Lasmea et Anne Rizand, directrice du Cemagref de Clermont-Ferrand pour avoir mis à ma disposition locaux et matériels. Je remercie également la DGA (Direction générale de l'armement) pour le financement et pour le suivi de cette thèse.

Je tiens à exprimer tout mes remerciements à Simon Lacroix et à Karsten Berns pour l'intérêt qu'ils ont porté à ma thèse en acceptant d'en être rapporteurs et à Jean-Pierre Dérutin pour avoir présidé mon jury de thèse.

Cette thèse ne se serait pas aussi bien déroulée si elle n'avait pas été encadrée par des personnes aussi compétentes que sympathiques. Aussi je tiens à remercier mon directeur de thèse Roland Chapuis pour ses conseils avisés et son expertise dans le domaine. Je remercie aussi mon co-encadrant Christophe Debain pour son implication dans mes travaux et son suivi au jour le jour.

Cette thèse s'est également bien déroulée grâce au cadre dans lequel elle a été effectuée. Pour cela je remercie chaleureusement l'équipe technique Gravir du laboratoire Lasmea ainsi que l'équipe TEAM du Cemagref pour toute l'aide et la bonne humeur que ces personnes m'auront procurées.

Je remercie aussi tous mes amis et collègues qui m'auront permis de décompresser et de me détendre pendant toute cette période pour pouvoir mieux repartir par la suite. Je tiens notamment à remercier Pierre Delmas, mon partenaire de tranchée, pour les longs moments de débat philosophique sur les répliques de Montaigne.

Comment écrire des remerciements sans parler de sa famille. J'adresse donc un profond remerciement à mes parents et mes frères pour tout ces moments parfois anodins qui m'auront fait voyager hors du quotidien. C'est donc tout naturellement que je leur dédie cette thèse.

Résumé

Cette thèse concerne la perception de l'environnement pour le guidage automatique d'un robot mobile. Lorsque l'on souhaite réaliser un système de navigation autonome, plusieurs éléments doivent être abordés. Parmi ceux-ci nous traiterons de la franchissabilité de l'environnement sur la trajectoire du véhicule. Cette franchissabilité dépend notamment de la géométrie et du type de sol mais également de la position du robot par rapport à son environnement (dans un repère local) ainsi que l'objectif qu'il doit atteindre (dans un repère global).

Les travaux de cette thèse traitent donc de la perception de l'environnement d'un robot au sens large du terme en adressant la cartographie de l'environnement et la localisation du véhicule. Pour cela un système de fusion de données est proposé afin d'estimer ces informations. Ce système de fusion est alimenté par plusieurs capteurs dont une caméra, un télémètre laser et un GPS. L'originalité de ces travaux porte sur la façon de combiner ces informations capteurs.

A la base du processus de fusion, nous utilisons un algorithme d'odométrie visuelle basé sur les images de la caméra. Pour accroître la précision et la robustesse l'initialisation de la position des points sélectionnés se fait grâce à un télémètre laser qui fournit les informations de profondeur. De plus, le positionnement dans un repère global est effectué en combinant cette odométrie visuelle avec les informations GPS. Pour cela un procédé a été mis en place pour assurer l'intégrité de localisation du véhicule avant de fusionner sa position avec les données GPS.

La cartographie de l'environnement est toute aussi importante puisqu'elle va permettre de calculer le chemin qui assurera au véhicule une évolution sans risque de collision ou de renversement. Dans cette optique, le télémètre laser déjà présent dans le processus de localisation est utilisé pour compléter la liste courante de points 3D qui matérialisent le terrain à l'avant du véhicule. En combinant la localisation précise du véhicule avec les informations denses du télémètre il est possible d'obtenir une cartographie précise, dense et géo-localisée de l'environnement.

Tout ces travaux ont été expérimentés sur un simulateur robotique développé pour l'occasion puis sur un véhicule tout-terrain réel évoluant dans un monde naturel. Les résultats de cette approche ont montré la pertinence de ces travaux pour le guidage autonome de robots mobiles.

Mots-clefs : fusion de données, vision, perception, localisation, cartographie, robot mobile.

Abstract

This thesis addresses the perception of the environment for the automatic guidance of a mobile robot. When one wishes to achieve autonomous navigation, several elements must be addressed. Among them we will discuss the traversability of the environment on the vehicle path. This traversability depends on the ground geometry and type and also the position of the robot in its environment (in a local coordinate system) taking into account the objective that must be achieved (in a global coordinate system).

The works of this thesis deal with the environment perception of a robot in the broad sense by addressing the mapping of the environment and the location of the vehicle. To do this, a data fusion system is proposed to estimate these informations. The fusion system is supplied by several low cost sensors including a camera, a rangefinder and a GPS receiver. The originality of this work focuses on how to combine these sensors informations.

The base of the fusion process is a visual odometry algorithm based on camera images. To increase the accuracy and the robustness, the initialization of the selected points position is done with a rangefinder that provides the depth information. In addition, the localization in a global reference is made by combining the visual odometry with GPS information. For this, a process has been established to ensure the integrity of localization of the vehicle before merging its position with the GPS data.

The mapping of the environment is also important as it will allow to compute the path that will ensure an evolution of the vehicle without risk of collision or overturn. From this perspective, the rangefinder already present in the localization process is used to complete the current list of 3D points that represent the field in front of the vehicle. By combining an accurate localization of the vehicle with informations of the rangefinder it is possible to obtain an accurate, dense and geo-located map environment.

All these works have been tested on a robotic simulator developed for this purpose and on a real all-terrain vehicle moving in a natural world. The results of this approach have shown the relevance of this work for autonomous guidance of mobile robots.

Keywords : data fusion, vision, perception, localization, mapping, mobile robot.

Table des matières

Introduction Générale	xv
1 Problématique : la navigation autonome	1
1.1 Cahier des charges	2
1.2 La perception pour l'action	5
1.2.1 L'influence de la perception sur le guidage	6
1.2.2 L'influence du guidage sur la perception	6
1.3 Conclusion	6
2 Localisation d'un véhicule par fusion Caméra-Télémètre	7
2.1 Introduction Générale	7
2.2 Techniques de localisation pour véhicules autonomes	7
2.2.1 Localisation absolue	8
2.2.1.1 Localisation absolue par GPS	8
2.2.1.2 localisation absolue par SIG	8
2.2.2 Localisation relative	9
2.2.3 Fusion de localisation absolue-relative	10
2.2.4 Bilan des méthodes de localisation	10
2.3 Odométrie visuelle pour la navigation de robots	11
2.3.1 Localisation par triangulation	12
2.3.2 La géométrie épipolaire	13
2.3.3 L'ajustement de faisceaux	15
2.3.4 Le filtrage de Kalman étendu	15
2.3.5 Le filtre d'information	17
2.3.6 Le filtre à particules	18
2.3.7 Bilan des méthodes d'odométrie visuelle	19
2.4 Odométrie visuelle par filtrage de Kalman étendu	19
2.4.1 Observation de points en coordonnées cartésiennes	21
2.4.2 Observation de points en coordonnées sphériques	25
2.4.3 Conclusion	27
2.5 Utilisation de données Lidar pour l'aide à la localisation	28

2.5.1	Introduction	28
2.5.2	Odométrie visuelle par couplage Lidar/Caméra	30
2.5.2.1	Principe	30
2.5.2.2	Ajout des données Lidar dans le vecteur d'état	31
2.5.2.3	Asynchronisme entre les mesures Lidar et les mesures de Localisation	32
2.5.2.4	Sélection des points Lidar dans le vecteur d'état	33
2.5.3	Mise en correspondance de points issus du télémètre	34
2.5.4	Bilan de la fusion Caméra-Télémètre	35
2.6	Perception Intelligente de l'environnement dans l'espace de la caméra	35
2.6.1	Introduction	35
2.6.2	Méthode classique	36
2.6.2.1	Principe	36
2.6.2.2	Recherche des appariements dans l'espace caméra	37
2.6.2.3	Coût opératoire de la méthode	37
2.6.3	Méthode point par point	40
2.6.3.1	Principe	40
2.6.3.2	Recherche des appariements dans l'espace caméra	40
2.6.3.3	Coût opératoire de la méthode	41
2.6.4	Bilan de la perception focalisée	42
2.7	Correction de la dérive de la localisation absolue	43
2.7.1	État du système	47
2.7.2	Estimation du biais $\vec{B} + \vec{D}_t$	48
2.7.2.1	Évolution du biais	48
2.7.2.2	Évolution par addition	52
2.7.2.3	Mise à jour du biais	55
2.7.2.4	Prise en compte du biais et correction de l'état estimé	56
2.8	Conclusion	57
3	Modèle Numérique de Terrain	59
3.1	Introduction	59
3.2	Approches de reconstruction de terrain	61
3.2.1	Introduction	61
3.2.2	Reconstruction de l'environnement par Stéréo-Vision	61
3.2.3	Reconstruction de l'environnement par télémètre 3D dense	64
3.2.4	Reconstruction de l'environnement par SLAM	65
3.2.5	Bilan des méthodes de reconstruction de l'environnement	65
3.3	Approche de modèle numérique de terrain par couplage caméra télémètre	66

3.3.1	Positionnement Espace-Temps d'une donnée Lidar dans le Modèle Numérique de Terrain	67
3.3.2	Mise à jour du MNT lors d'une correction de la dérive de localisation	67
3.4	Stratégie de gestion de la carte de l'environnement	67
3.4.1	Interaction du processus de guidage pour les besoins de perception	68
3.5	Conclusion	71
4	Résultats expérimentaux	73
4.1	Résultats théoriques sur MATLAB	73
4.1.1	Procédure de test	73
4.1.2	Test des différentes méthodes de positionnement	76
4.1.3	Test des méthodes globales et point par point	79
4.1.4	Test des différentes méthodes d'observation	82
4.2	Résultats en environnement simulé	85
4.2.1	Résultat de localisation	85
4.2.1.1	Introduction du simulateur ASROCAM	88
4.2.1.2	Résultats sans correction de la dérive	90
4.2.1.3	Résultats avec correction de la dérive	95
4.2.1.4	Bilan intermédiaire	96
4.2.2	Résultat de construction du MNT global	105
4.2.3	Résultat de construction du MNT local	107
4.2.3.1	Reconstruction par Lidar seul	107
4.2.3.2	Reconstruction densifiée par la caméra	108
4.2.4	Bilan	108
4.3	Résultats en environnement réel	114
4.3.1	Résultats de localisation	114
4.3.1.1	Dispositif expérimental	114
4.3.1.2	Résultats sans correction de la dérive	115
4.3.1.3	Résultats avec correction de la dérive	120
4.3.2	Résultat de construction du MNT local	120
4.3.3	Bilan	125
4.4	Résultat de temps de Calcul	131
4.4.1	Temps de calcul de la mise à jour de l'EKF	131
4.4.2	Temps de calcul du processus entier en expérimentation	131
	Conclusion - Perspective	135

A Annexe	139
A.1 Calcul des jacobienes des observations	139
A.2 Calibration du système Caméra-Télemètre	144
Bibliographie	147

Table des figures

1	Un véhicule de transport urbain autonome : le VIPA.	xv
1.1	<i>Stanley</i> , le vainqueur du DARPA Grand Challenge 2005	2
1.2	La démarche de traitement des informations.	3
1.3	La comparaison entre une démarche linéaire (à gauche) et une démarche cognitive (à droite).	3
1.4	La voiture autonome du laboratoire Italien VisLab.	4
1.5	La voiture autonome du projet de Google : la Google Car.	5
2.1	Les effets de l'ignorance du facteur d'échelle.	12
2.2	La géométrie épipolaire. Source [Forsyth et Ponce, 2002].	13
2.3	L'ajustement de faisceaux local proposé par Mouragnon lorsque la prise de vue de la caméra C_i est ajoutée.	16
2.4	Première façon d'observer un point : coordonnée cartésienne	21
2.5	Observation d'un amer dans la caméra.	23
2.6	Deuxième façon d'observer un point : coordonnée sphérique	25
2.7	Observation d'angle dans une caméra	28
2.8	Points de Harris extrait de l'image vis-à-vis de la zone d'intérêt.	29
2.9	Le système caméra-télémètre	30
2.10	Le couplage Lidar-Caméra : utiliser les informations du télémètre dans l'odométrie visuelle.	31
2.11	Le positionnement dans le repère monde des données télémètre.	32
2.12	Prise en compte de la désynchronisation entre données caméra et données télémètre.	33
2.13	La récolte des patchs dans une image valide : on projette les données Lidar d'une date j dans une image caméra de date valide k puis on récupère les patchs de cette image.	35
2.14	La focalisation d'attention : utiliser les informations précédentes pour les nouvelles recherches.	36
2.15	La focalisation d'attention : utiliser la covariance de l'observation pour la recherche d'appariement.	37

2.16	La projection des ellipses d'incertitude dans l'espace caméra en méthode globale.	38
2.17	La projection des ellipses d'incertitude dans l'espace caméra en méthode point par point (en vert).	41
2.18	Utilisation des matrices creuses en méthode point par point.	41
2.19	La courbe d'évolution du coût opératoire de chaque méthode.	43
2.20	Vue de dessus de la trajectoire effectuée par le robot virtuel (en noir) ainsi que la trajectoire estimée par l'odométrie visuelle (en bleu).	44
2.21	Erreur de localisation en X, Y et Z de l'EKF par rapport à une trajectoire de référence et incertitudes associées.	45
2.22	Le modèle d'évolution du véhicule.	48
2.23	Evolution de l'incertitude de position pour une incertitude en cap de π	50
2.24	Ellipse de l'incertitude en position en utilisant une jacobienne de l'évolution	51
2.25	Ellipses d'incertitudes possibles pour englober la position vraie.	52
2.26	Définition des incertitudes en fonction du biais en cap.	53
2.27	Correction de l'odométrie visuelle : transformation rigide du repère estimé au repère absolu	56
3.1	Être ou ne pas être un obstacle.	60
3.2	La partie de l'environnement utile pour le guidage du véhicule.	61
3.3	Une paire stéréo et le champ de vision commun aux 2 caméras.	62
3.4	La rectification de deux images stéréo.	63
3.5	Une carte de disparité produite avec une paire stéréo.	63
3.6	Sélection de points : les points qui passent derrière le véhicule sont supprimés de la carte.	69
3.7	Configurations où des espaces non-observables de l'environnement apparaissent.	70
4.1	L'initialisation d'un amer dans l'EKF "classique" à partir d'un point télémètre.	74
4.2	Vue de la trajectoire effectuée par le robot virtuel.	75
4.3	Vue de dessus de la trajectoire effectuée par le robot virtuel ainsi que les trajectoires estimées par les différents types de localisation (GM : Méthode globale).	77
4.4	Erreur totale moyenne des 100 réalisations des processus pour les différents types de localisation.	78
4.5	Erreur moyenne de localisation des 100 réalisations en X, Y et Z de chaque méthode par rapport à la trajectoire de référence.	79

4.6	Erreur moyenne d'orientation des 100 réalisations en tangage, lacet, roulis (α , β et γ) de chaque méthode par rapport à l'orientation de référence.	80
4.7	Erreur moyenne commise sur la métrique des 100 réalisations de chaque méthode.	81
4.8	Erreurs moyenne de localisation des 100 réalisations en X, Y et Z en méthode globale et point par point.	82
4.9	Erreurs moyenne d'orientation des 100 réalisations en tangage, lacet et roulis (α , β et γ) en méthode globale et point par point. . . .	83
4.10	Erreur totale moyenne des 100 réalisations en méthode globale (GM) et point par point (PPM).	84
4.11	Erreur totale moyenne des 100 réalisations suivant chaque méthode d'observation.	85
4.12	Erreurs moyenne de localisation des 100 réalisations en X, Y et Z suivant chaque méthode d'observation.	86
4.13	Erreurs moyenne d'orientation des 100 réalisations en tangage, lacet et roulis (α , β et γ) suivant chaque méthode d'observation. . .	87
4.14	Vue externe du scénario présenté sur le simulateur.	91
4.15	Erreur totale de localisation en utilisant la méthode globale.	92
4.16	Vue du dessus de la trajectoire calculée en méthode globale.	92
4.17	Erreur de localisation suivant chaque variable en méthode globale.	93
4.18	Erreur d'orientation suivant chaque variable en méthode globale.	94
4.19	Erreur totale de localisation en utilisant la méthode point par point.	95
4.20	Vue du dessus de la trajectoire calculée en méthode point par point.	96
4.21	Erreur de localisation suivant chaque variable en méthode point par point.	97
4.22	Erreur d'orientation suivant chaque variable en méthode point par point.	98
4.23	Vue du dessus de la trajectoire parcourue en environnement simulé après correction de la dérive.	99
4.24	Erreur de localisation suivant chaque variable après correction de la dérive.	100
4.25	Erreur d'orientation suivant chaque variable après correction de la dérive.	101
4.26	Le phénomène de glissement pour l'appariement des points d'une image à l'autre	102
4.27	Glissements en u et v obtenus par rapport aux vraies projections des amers	103
4.28	Erreur totale de localisation en utilisant la méthode point par point en absence de glissement.	104

4.29	Modèle numérique de terrain obtenu à partir des points télémétriques en utilisant la vraie pose du robot.	106
4.30	Modèle numérique de terrain obtenu à partir des points télémétriques en utilisant l'observation cartésienne en méthode point par point.	106
4.31	Modèle numérique de terrain local obtenu à partir des points télémétriques seuls pour l'itération 300.	109
4.32	Modèle numérique de terrain local obtenu à partir des points télémétriques seuls pour l'itération 800.	110
4.33	Erreur de reconstruction locale obtenu sur l'ensemble de la trajectoire simulé	111
4.34	Histogramme des erreurs de reconstruction obtenu sur une itération du processus.	111
4.35	Modèle numérique de terrain local obtenu pour l'itération 300 avec peu de données télémétriques (une acquisition tout les mètres). . .	112
4.36	Modèle numérique de terrain local obtenu pour l'itération 300, augmenté avec les données caméra.	113
4.37	Notre véhicule d'expérimentation AROCO lors du Symposium ROBOTICS 2010.	114
4.38	Erreur totale de localisation en expérimentation réelle sans correction de la dérive.	116
4.39	Vue du dessus des trajectoires calculées en expérimentation réelle sans correction de la dérive.	117
4.40	Paramètres de localisation suivant chaque variable en expérimentation réelle sans correction de la dérive.	118
4.41	Paramètres d'orientation suivant chaque variable en expérimentation réelle sans correction de la dérive.	119
4.42	La correspondance entre la variation de l'état et l'environnement pour des points clés du parcours.	121
4.43	Vue du dessus de la trajectoire parcourue en environnement réelle après correction de la dérive.	122
4.44	Erreur de localisation suivant chaque variable après correction de la dérive.	123
4.45	Erreur d'orientation suivant chaque variable après correction de la dérive.	124
4.46	Schéma des moments clés de l'expérimentation où seront montrés le MNT en local.	125
4.47	Modèle numérique de terrain local obtenu en environnement réel pour l'itération 180.	126
4.48	Modèle numérique de terrain local obtenu en environnement réel pour l'itération 325	127

4.49	Modèle numérique de terrain local obtenu en environnement réel pour l'itération 650.	128
4.50	Modèle numérique de terrain local obtenu en environnement réel pour l'itération 920.	129
4.51	Modèle numérique de terrain local obtenu en environnement réel pour l'itération 1920.	130
4.52	Temps de calcul de la mise à jour de l'EKF pour la méthode globale en fonction du nombre d'amers dans le vecteur d'état.	132
4.53	Temps de calcul de la mise à jour de l'EKF pour la méthode point par point en fonction du nombre d'amers dans le vecteur d'état.	132
4.54	Temps de calcul en méthode globale en fonction du nombre d'amers dans le vecteur d'état.	133
4.55	Temps de calcul en méthode point par point en fonction du nombre d'amers dans le vecteur d'état.	134
4.56	L'utilisation du posé de robot pour prendre en compte le principe que le robot roule sur le sol.	137
A.1	La calibration : passer par un repère tiers	146

Introduction Générale

Ces dernières années, les véhicules autonomes sont devenus un enjeu économique et écologique de la société. En effet, que ce soit dans le domaine civil, militaire ou agricole, de nombreuses applications peuvent bénéficier des avancées scientifiques et technologiques de la communauté robotique. Par exemple en agriculture, l'inflation de la puissance et du poids des machines cause des dommages aux sols les rendant plus sensibles aux phénomènes de lessivage. De plus leur grande taille rend difficile leur contrôle précis par un opérateur, ce qui paraît peu compatible avec les exigences d'une agriculture soucieuse d'un développement durable. Des solutions basées sur des développements robotiques sont proposées. Elles impliquent des véhicules de taille moyenne dont un est dirigé par un opérateur, les autres étant guidés par un système automatique.

Cela se produit aussi dans d'autres domaines comme les transports en commun. Par exemple les Véhicules Individuels Publics Autonomes (VIPA) permettent le transport de passagers en milieu urbain sans conducteur (voir figure 1) : c'est le concept de l'ascenseur horizontal dont la fonction est de transporter des personnes sur la fin de leurs parcours en complément des transports en commun actuels.

Pour être économiquement viable, ce concept peut avantageusement bénéficier de l'utilisation de machines autonomes c'est à dire fonctionnant sans intervention



FIGURE 1 – Un véhicule de transport urbain autonome : le VIPA.

humaine. Si l'intérêt économique est clairement justifié dans cette approche elle doit également proposer une sécurité d'utilisation conforme aux standards attendus. On peut d'ores et déjà arguer le fait que l'utilisation ou le fonctionnement autonome permettront de s'affranchir des problèmes de fatigue accumulée par les conducteurs de véhicules ; cette fatigue étant à la source de pertes de vigilance de ces derniers. Dans le domaine militaire, on peut évoquer le cas du déminage ou du convoi de véhicules humanitaires pour lesquels des robots autonomes pourraient explorer un espace avec peu d'intervention de la part des opérateurs (exemple : projet RINAVEC ¹).

Toutes ces applications exploitent des développements robotiques importants visant à augmenter l'autonomie décisionnelle des véhicules. Mais pour pouvoir considérer un véhicule comme autonome il doit disposer de certaines fonctionnalités :

- Un module permettant d'analyser sa situation actuelle et les actions à mettre en place pour satisfaire la demande exprimée (mission-objectif).
- Un module permettant de récolter des informations sur son environnement et/ou sa localisation.
- Un module permettant de commander ses actions.

Clairement, ces trois modules sont en interactions les uns avec les autres. Par exemple les actions du robot dépendront de ses capacités à percevoir son environnement. De même sa stratégie pour connaître sa situation dépendra des capacités perceptives mises à sa disposition.

Dans des applications spécifiques, un robot peut très bien avoir seulement certaines fonctionnalités pour être considéré comme autonome. Citons notamment la ligne 14 du métro Parisien qui transporte chaque jour des milliers de passagers. Son processus décisionnel est très fortement réduit du fait de l'environnement fermé dans lequel elle évolue. Cependant, toutes ces fonctionnalités doivent être présentes dans un milieu naturel tout comme en milieu urbain et leurs interactions possibles sont très nombreuses.

Si l'on considère ces problématiques indépendamment les unes des autres, plusieurs solutions existent. Des solutions technologiques ont été développées pour percevoir l'environnement. En effet la reconstruction de terrain est simple en soi dès lors que l'on dispose d'un capteur télémétrique dense. Cependant ces technologies ne sont pas suffisantes pour répondre à l'ensemble des contraintes de perception d'un véhicule autonome qu'est la perception, notamment en terme de localisation. De plus, les coûts engendrés par ces technologies sont prohibitifs.

Les travaux présentés dans ce mémoire proposent une solution de perception de l'environnement au sens large du terme (i.e. localisation et géométrie

1. RINAVEC : Reconnaissance d'Itinéraire et Navigation en convoi de Véhicules Communicants

du monde) d'un véhicule évoluant en milieu ouvert non instrumenté. Aucune connaissance *a priori* de cet environnement n'est à disposition du système. Les problèmes du processus décisionnel et de l'action ne seront pas traités dans ce mémoire. Cependant pour répondre aux besoins d'un véhicule autonome, les interactions avec ces fonctionnalités seront prises en compte.

Objectifs de la thèse

Comme nous l'avons vu, percevoir c'est à la fois connaître l'environnement par rapport à soi-même, mais aussi se connaître par rapport à l'environnement. Autrement dit c'est pouvoir se localiser, à la fois relativement à l'environnement proche mais aussi globalement dans le monde. C'est également savoir modéliser l'environnement pour pouvoir y déceler les informations qui sont pertinentes. Cependant le monde est rempli d'informations riches et complexes, et ces informations sont souvent de différentes natures. Des systèmes ont été créés pour capter ces informations, et ceci de plusieurs façons. Ces capteurs sont dits "extéroceptifs" pour percevoir les informations venant de l'extérieur ou "proprioceptifs" pour celles qui relatent du comportement interne du robot.

Si on prend l'être humain comme référence, on voit tout de suite qu'il a lui aussi plusieurs "capteurs" pour l'aider à percevoir le monde. Il se sert de ses yeux pour voir la géométrie devant lui et pour se localiser en relatif par rapport à celui-ci. Il utilise ses mains pour déterminer la nature des objets ainsi que la distance qui les sépare. Ses oreilles lui permettent d'anticiper les mouvements tout autour de lui. Grâce à son oreille interne, il peut trouver son orientation référencée par rapport à la gravité. Au final, cet ensemble de "capteurs" lui donne la possibilité d'évoluer librement dans son environnement car le processus de décision/action de son cerveau dispose des informations nécessaires. Aussi, cette perception peut être sollicitée pour mieux répondre aux besoins car la finalité de cet ensemble est la navigation. C'est ce système cognitif dans sa globalité qui fait la force des êtres vivants.

Maintenant si on regarde ce qui se fait "classiquement" dans les domaines scientifiques, on remarque que cet aspect cognitif n'est pas pris en compte. En effet, les méthodes classiques de guidage autonome suivent une logique séquentielle. La plupart du temps, on commence par percevoir : on utilise alors les algorithmes les plus performants pour arriver aux meilleurs résultats de localisation et de reconstruction de l'environnement. Puis, ces résultats servent à alimenter les systèmes décisionnels qui doivent absorber toutes ces informations pour en extraire le comportement souhaité du véhicule en espérant qu'il corresponde aux objectifs de l'application. Finalement ce comportement est obtenu au moyen des lois de commandes et des actionneurs embarqués sur le véhicule. Tout ceci est réitéré aussi souvent que faire

se peut. Ce cheminement a permis de mettre en place des robots mobiles autonomes en situation réelle. Cependant la mise en pratique de ce formalisme requiert beaucoup d'hypothèses de travail. Ces hypothèses vont de la limite 'dure' de vitesse de navigation à la puissance de calcul nécessaire quels que soient les besoins de l'application. Par exemple si le système de perception requiert beaucoup de ressources pour ses calculs, alors les fonctionnalités suivantes s'en trouvent pénalisées. Une réaction en chaîne se produit faisant échouer le système.

L'objectif de cette thèse est de concevoir un système de perception intelligent pour un robot mobile évoluant dans un milieu naturel. On s'intéressera particulièrement à deux fonctions :

- **la localisation absolue intègre permettant de garantir que le véhicule reste dans un couloir de navigation et suffisamment précise pour permettre son guidage automatique. On s'attachera dans ces travaux à montrer les améliorations nécessaires aux techniques actuelles pour assurer cette intégrité de localisation et obtenir une estimation précise de la pose du véhicule.**
- **la reconstruction d'une partie de l'environnement du robot correspondant à la zone dans laquelle il est susceptible d'évoluer dans un futur proche. Dans cette partie, l'accent sera porté sur l'adéquation entre les besoins de l'application et la richesse de la reconstruction de la zone perçue localement par le robot puis à sa projection dans le repère global de navigation.**

Contributions

Les recherches scientifiques de ces dernières années ont apporté beaucoup d'innovations dans le domaine de la perception, que ce soit sur les aspects localisation ou reconstruction de la géométrie d'une surface. On peut notamment parler des fermetures de boucles pour les systèmes de localisation ou des méthodes hors-ligne d'ajustement de données. D'un autre côté, dans les domaines décisionnels et de guidage les innovations sont elles aussi importantes. Cependant, les innovations de ces domaines vont chacune de leur côté et un fossé commence à se creuser. Par conséquent même un système dit "classique" a du mal à être mis en œuvre à cause des écarts scientifiques trop importants entre perception et guidage. De plus en plus souvent des problèmes de sécurité apparaissent ce qui freine la diffusion de ces technologies. Pourtant, chacun sait qu'il vaut mieux valider chaque étape élémentaire avant de s'attaquer au problème de façon macroscopique. Dans notre cas, nous pensons qu'il vaut mieux valider notre approche de système cognitif. Aussi, les travaux menés dans cette thèse ne cherchent pas à faire de *la perception pour la perception* mais visent à implanter *une perception pour l'application*. Dans ce

cadre, deux contributions seront suggérées :

A Fusion de caméra et télémètre pour une localisation améliorée

Nous proposons une méthode de fusion de données caméra plus télémètre. Ce procédé permet d'utiliser une connaissance *a priori* d'informations de profondeur du laser dans une image caméra. Ces travaux sont décrits dans [Malartre *et al.*, 2009b].

B Stratégie de Calcul pour une localisation et une reconstruction de géométrie temps réelle

Une fois la fusion de données effectuée, la localisation du véhicule est calculée en utilisant uniquement le temps de calcul disponible. Ensuite cette donnée de positionnement est utilisée pour effectuer une reconstruction 3D de l'environnement grâce encore aux informations télémètre. Tous ces procédés ont été exposés dans [Malartre *et al.*, 2010].

Organisation du mémoire

Pour répondre à la problématique de perception intelligente le mémoire va être organisé autour de 4 parties :

Le contexte de travail ainsi que la philosophie de notre approche seront expliqués dans le chapitre 1. Nous partirons de l'objectif à atteindre pour exprimer nos besoins en terme de perception.

Notre méthode de localisation et perception de l'environnement par fusion de données sera présentée dans le chapitre 2. Nous décrirons comment extraire une localisation qui réponde aux besoins de l'objectif grâce à l'utilisation de nos capteurs. Ensuite nous expliquerons comment nous obtenons une géométrie du monde dans le chapitre 3. L'interaction du système de perception avec le processus décisionnel sera expliquée.

Dans le chapitre 4 nous exposerons notre procédure de mise en œuvre ainsi que les résultats obtenus.

Finalement, nous conclurons et donnerons quelques perspectives de ces travaux.

Chapitre 1

Problématique : la navigation autonome

Faire naviguer un robot de manière autonome est une tâche difficile à mettre en œuvre qui nécessite la mise en œuvre d'un certain nombre de domaines de compétence. Ces compétences vont du plus bas niveau de conception comme la mécanique des actionneurs au concept de plus haut niveau comme la planification de trajectoires sécurisées. Elles ont pour objectifs la réalisation de la mission de navigation mais également la nécessité de préserver l'intégrité du robot et celle de son environnement (personne, objet, autre robots, etc...). Pendant de nombreuses années ce double challenge a souvent été jugé comme une utopie par la société. Pourtant, en 2004, le DARPA ¹ lance son premier grand challenge dans ce domaine.

Ce challenge consiste à parcourir un circuit dans le désert des Mojaves de plus de 200 km en moins de 10 heures et de façon entièrement autonome. Si la première édition ne s'est pas vu attribuée de vainqueur, la deuxième en 2005 a été remportée par l'équipe de l'université de Stanford avec son véhicule *Stanley* (figure 1.1).

Ce challenge a permis de montrer des éléments de faisabilité d'un système de navigation. L'ensemble des techniques utilisées par l'université de Stanford pour réaliser le DARPA Grand Challenge a été détaillé dans [Thrun *et al.*, 2006]. Les notions essentielles de localisation, de perception de l'environnement, de calcul de trajectoire ont été mises bout à bout afin de réaliser ce défi.

Si les travaux réalisés à l'occasion de ces challenges ont été un formidable catalyseur pour le développement et la promotion de la robotique, ils ne prétendent pas déboucher sur des systèmes répondant aux préoccupations de la société. Pour cela, ils doivent être généralisés en étant capable d'adresser un grand nombre de scénarios dont les paramètres sont souvent dynamiques. A ce titre, de nombreux développements sont en cours comme par exemple ceux du projet ANR SafePlatoon dont l'objet est de mettre en situation des véhicules autonomes dans des en-

1. Defense Advanced Research Projects Agency



FIGURE 1.1 – *Stanley*, le vainqueur du DARPA Grand Challenge 2005

vironnements intégrant toutes les contraintes d’une utilisation réelle même les plus improbables.

Pour cela, de nouvelles approches doivent être considérées comme celles proposées par Kelly ([Urmson *et al.*, 2008]) où les auteurs focalisent leur perception sur les informations pertinentes de l’environnement afin de réaliser au mieux la navigation. Nous considérerons pour notre part que le processus de navigation doit piloter les ressources et non l’inverse. En effet, il est plus pertinent de solliciter les ressources au bon moment au regard de l’objectif visé. Ces approches peuvent souvent être qualifiées de cognitives car le fonctionnement du système s’adapte à la dynamique de l’environnement jusque dans la gestion de ses ressources. La différence avec les approches classiques est montrée en figure 1.3.

1.1 Cahier des charges

L’ambition de notre équipe de recherche est de concevoir un système autonome de navigation qui garantira l’intégrité physique du véhicule ainsi que celle de ses occupants ou des éléments extérieurs. Le DARPA grand challenge a montré le caractère réaliste de cette ambition. Maintenant, l’enjeu des recherches actuelles est de pouvoir généraliser ce concept à différents types de véhicules et de pouvoir les appliquer en répondant aux problèmes de notre société aussi bien sur le plan technique qu’économique.

Pour cela il nous apparait évident que la priorité pour notre système de navigation est la garantie de l’intégrité physique du véhicule, des éléments environnants et de ses occupants. Aussi nous décomposerons la notion de guidage autonome selon les étapes suivantes :

- **A partir d’une mission qui lui a été assignée, par exemple aller de sa position actuelle A à une position future B, le robot doit définir les actions**

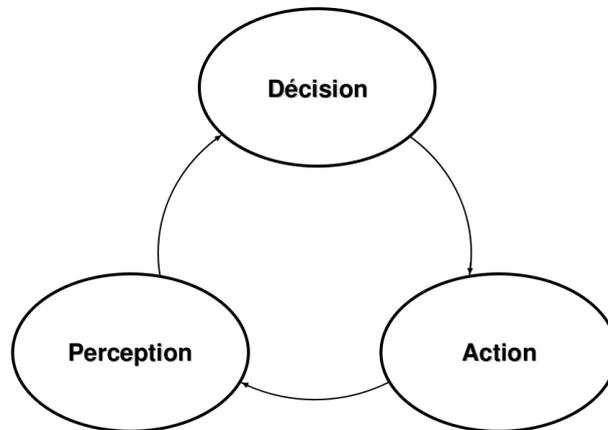


FIGURE 1.2 – La démarche de traitement des informations.

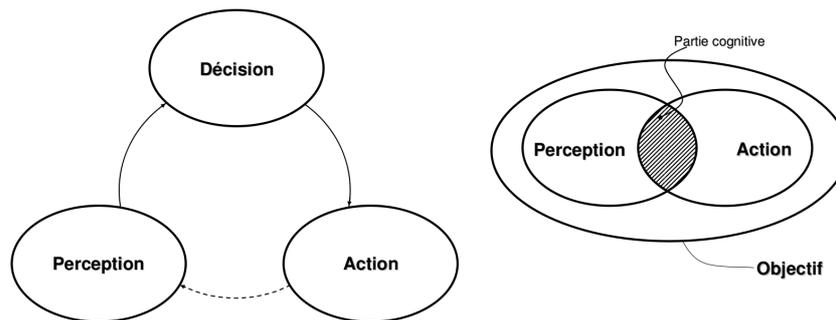


FIGURE 1.3 – La comparaison entre une démarche linéaire (à gauche) et une démarche cognitive (à droite).

à exécuter pour réaliser cette mission. Une approche classique consiste à lancer une tâche de localisation avant de commander le robot. Un processus parallèle serait alors chargé de surveiller l'environnement proche du véhicule afin d'éviter toute collision. Nous proposons une autre vision des choses : pour nous le véhicule doit identifier à chaque instant de la mission les meilleures actions à mettre en place pour assurer que toutes les contraintes de sa mission soient respectées et en particulier un fonctionnement précis et intègre.

- Parmi tous les paramètres d'une mission de navigation autonome, la localisation du robot en est certainement un des plus importants. Cependant, la tâche de localisation n'est utile que si le système de guidage ne satisfait pas



FIGURE 1.4 – La voiture autonome du laboratoire Italien VisLab.

- aux contraintes de localisation de la mission (voir [Tessier, 2007]). Ainsi les contraintes de localisation seront probablement très différentes pour une mission qui se déroule dans un désert ou pour une mission effectuée dans un bâtiment. C'est la stratégie évoquée précédemment qui organisera la gestion des ressources en fonction des paramètres de l'objectif et de l'état du robot.
- Cette localisation est généralement associée à une carte qui peut être donnée à l'avance ou construite au fur et à mesure du déroulement de la mission. Là aussi, c'est la stratégie qui met à contribution les éléments de la carte ou qui la renseigne de façon pertinente.
 - La préservation de l'intégrité du véhicule est fonction des capacités du véhicule et des paramètres de l'environnement qu'il doit traverser. Ceux-ci sont généralement estimés par des techniques de perception de l'environnement souvent coûteuses en temps de calcul mais indispensables à la réalisation d'un système opérationnel en environnement ouvert. La gestion de cette étape est donc un élément critique pour la stratégie de navigation puisqu'impliquant directement la sûreté de fonctionnement du véhicule et la mobilisation des ressources.

Aujourd'hui plusieurs équipes sont impliquées dans l'application de ces concepts à la conduite automatique de véhicules dans un environnement rural, urbain ou péri-urbain. On citera plus particulièrement l'équipe Italienne du laboratoire VisLab de Parme avec leur projet de traverser deux continents *sans a priori sur l'environnement et sans hypothèse de présence ou d'absence d'autres véhicules* (voir figure 1.4). Leur projet consistait à rejoindre l'exposition mondiale de Chine en parcourant les 13000 km qui séparent leur laboratoire de Shanghai (<http://viac.vislab.it/>).

Un autre projet encore plus médiatisé est celui de l'entreprise Google avec ses



FIGURE 1.5 – La voiture autonome du projet de Google : la Google Car.

véhicules (voir figure 1.5) qui ont déjà parcouru plus de 2500 km sans l'aide de conducteurs.

Tous ces projets permettent aujourd'hui de concrétiser une ambition scientifique qui date du début des années 90 ([Dickmanns *et al.*, 1993]) en proposant les premiers systèmes intégrés capables de remplir simultanément les principales fonctions du cahier des charges d'un système de conduite automatique.

De ces projets on peut dégager deux grands processus qui sont en interaction permanente : un premier typé **sensoriel** dont l'objet est la compréhension de l'environnement du robot, un second typé **action** dont l'objet est de réaliser la mission en toute sécurité.

Le fonctionnement des processus sera optimal s'ils sont alimentés par une stratégie intégrant en plus de l'objectif de la mission leurs interactions. Dans notre cas on parlera de **perception pour l'action**.

1.2 La perception pour l'action

L'interaction entre les processus de perception et d'action se traduit par une relation bi-directionnelle :

- L'influence de la tâche de perception sur la tâche action en terme de précision et d'intégrité des résultats fournis ainsi qu'en terme de ressources consommées.
- L'influence de la tâche action sur la tâche perception en terme de besoins exprimés, de prédiction et d'influence sur les conditions d'exécution du processus de perception.

1.2.1 L'influence de la perception sur le guidage

Le système de guidage est contraint dans ses déplacements par les limites de la tâche de perception. Ses actions sont exécutées conditionnellement à sa connaissance de l'environnement. Par exemple la vitesse maximale que le véhicule pourra atteindre sera limitée par la distance de perception. En effet le système doit se laisser le temps (et donc la distance) nécessaire pour stopper le véhicule lorsqu'un élément totalement infranchissable apparaît dans le système perceptif.

Aussi, dans un virage rapide ou lors d'un franchissement d'un "obstacle", un changement rapide de trajectoire peut créer temporairement des pertes de localisation et des absences de perception (tout comme chez l'être humain). Ceci amène des limites de perception en terme de vitesse pour le processus de guidage.

1.2.2 L'influence du guidage sur la perception

D'un autre côté toute les informations d'un point de vue guidage peuvent aussi intervenir sur la tâche de perception. C'est à dire lorsque le processus de guidage sait qu'il va tourner il peut très bien "prévenir" la perception pour ne pas qu'elle se perde. Il se passe implicitement la même chose chez l'être humain quand il prend un virage en voiture il regarde du côté de celui-ci. Ensuite, un système de perception ne sait pas en lui-même les endroits les plus pertinents à regarder. Cependant la stratégie de navigation sait où elle doit aller. Ces informations pertinentes doivent donc être "demandées" à la perception par le processus de guidage.

Aussi, le système de navigation sait déterminer l'influence de l'environnement sur son état. Par exemple il peut évaluer sa position et son orientation lorsque l'environnement est connu (par un posé de robot par exemple). Il est donc capable d'estimer la variation de son état en fonction de la trajectoire prévue. Cette information peut être utilisée par le système de perception pour focaliser ses calculs sur un espace possible de résultat plus restreint.

1.3 Conclusion

Dans ce chapitre nous avons exposé l'approche que nous souhaitons mettre en œuvre pour réaliser notre système de navigation. Ce système doit inévitablement comporter un système de perception et un système de navigation. Cependant notre philosophie diffère dans le sens où perception et action ne s'exécutent pas de façon séquentielle mais interagissent l'une avec l'autre et partagent des informations de façon bi-latérale afin de réaliser au mieux la tâche demandée dans un soucis constant de sécurité.

Chapitre 2

Localisation d'un véhicule par fusion Caméra-Télémetre

2.1 Introduction Générale

Lorsque l'on parle de navigation autonome, les systèmes GPS sont très souvent cités. Cette addiction au système GPS vient du fait que l'on associe implicitement l'autonomie d'un véhicule avec un système de localisation. En effet, on imagine mal rejoindre notre destination si l'on ne sait pas où l'on est. Cette donnée de localisation constitue le retour d'information de notre système de guidage. Sans cela, aucun système ne peut converger vers une solution proche de la valeur finale. Cette localisation est de ce fait très importante si on veut se rapprocher d'un système complètement autonome. La priorité de notre système de perception est de reconstruire le monde à l'avant du robot. Cette reconstruction dépend de la localisation du véhicule qui est évaluée au fur et à mesure de l'avancée du robot. La question qu'il faut se poser maintenant est "Comment va-t-on se localiser ?".

Ce chapitre est dédié à la localisation d'un véhicule dans son environnement mais aussi dans le monde puisque comme nous l'avons vu ces deux aspects ne peuvent être dissociés. L'objectif de cette partie est de mettre en place un système de localisation à la fois précis localement et globalement, qui permettra la mise en place de la reconstruction de l'environnement.

2.2 Techniques de localisation pour véhicules autonomes

Pour se localiser, nous avons besoin d'informations relatives à l'état de notre véhicule. La localisation d'un véhicule ou d'un objet dépend du repère initial dans lequel elle va être effectuée. On peut mettre en vis-à-vis deux grands principes : la

localisation absolue et la localisation relative.

2.2.1 Localisation absolue

Le principe de la localisation absolue est d'avoir accès aux paramètres qui définissent la position du robot dans l'univers. D'un point de vue pratique, on peut limiter cette connaissance à la position sur terre. Le GPS (Global Positioning System) est un système de localisation par satellite créé dans les années 1970 par le département de la défense des États-Unis pour calculer la localisation d'un récepteur sur terre dans un contexte militaire. Le principe du GPS est d'utiliser la mesure du temps de vol des signaux émis par au moins 4 satellites. La position du récepteur est ensuite calculée par triangulation de ces mesures avec la position de chaque satellite (dont la position est précisément connue). D'autres systèmes de localisation absolue sont réalisés en utilisant une base de connaissances géoréférencées appelé SIG qui permet au véhicule de se situer par rapport aux éléments de la base qu'il perçoit.

2.2.1.1 Localisation absolue par GPS

De nombreux travaux concernant le suivi de trajectoire et autres processus de guidage utilisent le système GPS-RTK comme donnée de localisation (voir [Lenain *et al.*, 2006] où les auteurs mettent en œuvre un suivi de trajectoire en présence de glissement). Cependant, même si les données fournies par un tel système paraissent très précises (de l'ordre du centimètre), elles sont insuffisantes pour plusieurs raisons :

- Aussi précise soit cette localisation d'un point de vue global, elle n'est pas forcément suffisante d'un point de vue local. Ces quelques centimètres de précision peuvent être très importants relativement au véhicule et à la tâche qu'il doit accomplir.
- Ces informations sont dépendantes de l'environnement. Si l'on se trouve dans une ruelle étroite ou proche d'un arbre, des phénomènes de multi-trajets ou même une perte totale du signal peuvent apparaître.
- Enfin, les informations GPS sont incomplètes. La longitude, la latitude sont les données principales du GPS, le cap étant quant à lui calculé avec une succession de mesures. En revanche, les informations de roulis et de tangage du véhicule ne sont tout simplement pas présentes alors qu'un système de guidage pourrait en avoir besoin pour assurer son intégrité physique.

2.2.1.2 localisation absolue par SIG

On peut aussi réaliser une localisation absolue en utilisant un SIG (Système d'Informations Géographique). Un SIG est une base de données d'amers géo-

référencés. En observant ces amers à travers un capteur extéroceptif (comme une caméra par exemple dans [Jensfelt et Kristensen, 2001]) et en utilisant des méthodes de triangulation nous pouvons remonter à une localisation absolue de notre robot. Dans notre application, les interactions avec l'environnement se font localement. Ces méthodes sont en général très précises mais dépendent fortement de la qualité de la carte sur laquelle elles s'appuient. Ainsi, des éléments non cartographiés mais néanmoins détectés par les capteurs du véhicule peuvent être appariés avec des éléments présents de la base de données ce qui va rendre le système de localisation non-intègre. Ils peuvent cependant être avantageusement utilisés par le système comme cela est proposé par les méthodes localisation et cartographie simultanée (SLAM).

Si le référencement absolu de la localisation d'un véhicule a un intérêt indéniable dans la gestion des objectifs de sa mission (points de passage, couloir de navigation, ...) il comporte des limites dues à l'absence d'information locale susceptible de changements incompatibles avec une gestion globale de la navigation.

2.2.2 Localisation relative

L'autre façon de voir la localisation est plus centrée sur un aspect 'local' de positionnement. En localisation relative nous cherchons à savoir, en partant d'une position initiale connue, de combien s'est déplacé le véhicule relativement à son ancienne position. L'intérêt de cette méthode est que l'interaction entre localisation et guidage est permanente puisque l'application va chercher à évoluer dans un environnement dont les obstacles sont définis de façon locale par rapport à lui. La méthode la plus simple pour effectuer une localisation relative est d'utiliser les informations des capteurs proprioceptifs du véhicule comme les odomètres, les gyromètres ou encore l'angle de braquage des roues. La mesure fournie par un odomètre est la distance curviligne parcourue par la roue pendant un laps de temps. Le gyromètre est un capteur de vitesse angulaire qui peut, par exemple, déterminer le changement de cap du véhicule. Si on associe ces informations à un modèle cinématique de véhicule (voir [Campion *et al.*, 1996]) on peut estimer le déplacement relatif de notre véhicule. Une fusion plus poussée de ces capteurs permet d'obtenir des informations plus précises encore comme dans [Borenstein et Feng, 1996] par exemple où les auteurs proposent de fusionner l'odomètre et le gyromètre pour obtenir une information plus pertinente de l'orientation du véhicule.

Une autre façon est de recourir à une centrale inertielle qui fournit les paramètres d'accélération linéaire et de vitesse angulaire d'un objet (voir [Barbour et Schmidt, 2001]). Ensuite, une double intégration de ces informations permet de remonter à une localisation 6D du robot.

Nous pouvons aussi nous localiser en relatif en utilisant des capteurs extéroceptifs comme un télémètre laser ou une caméra par exemple. Pour cela, une observa-

tion d'un amer fixe dans l'espace capteur plusieurs fois de suite permet de retrouver le déplacement relatif du robot. Le SLAM (Simultaneous Localization And Mapping) est une technique de localisation et cartographie qui s'inspire de ce principe (voir [Durrant-Whyte et Bailey, 2006] et [Bailey et Durrant-Whyte, 2006]).

Le problème principal des méthodes de localisation relative est qu'elle subissent des dérives plus ou moins grandes au cours du temps. De plus, même si on connaît le déplacement relatif de notre robot à partir d'un repère initial, on ne connaît pas le positionnement initial du véhicule dans un repère monde. Le problème est que les lois de commande utilisées pour actionner le véhicule sont, la plupart du temps, écrites pour fonctionner dans un référentiel absolu. Une solution de localisation relative n'est donc pas suffisante.

2.2.3 Fusion de localisation absolue-relative

Nous avons vu que la localisation est une étape très importante pour prétendre à une autonomie totale des robots. Cette localisation doit être plus ou moins complète en terme d'information suivant le contexte de travail. Par exemple, dans ses travaux Laneurit cherche à calculer la position de son véhicule sur une route [Laneurit, 2006]. Pour cela il estime la position et le cap de son véhicule en considérant un *a priori* de sa position sur un monde plan.

La combinaison de localisation relative et absolue est possible comme dans [Bétaille *et al.*, 2007] où les auteurs fusionnent plusieurs types de capteurs extéroceptifs et proprioceptifs pour arriver à un résultat de localisation très précis et présentant peu de dérives. Cependant la mise en œuvre de cette fusion de capteurs est lourde et résout seulement le problème de localisation et pas celui de la perception de l'environnement. De plus, pour arriver à une telle précision, les dispositifs utilisés sont très coûteux.

Les travaux de Tessier [Tessier, 2007] comportent les mêmes problématiques que précédemment avec pour objectif supplémentaire de guider le véhicule et non plus de le localiser uniquement. Le but de ses travaux est de guider un véhicule sur une trajectoire connue (référéncée dans un repère monde) et réalisable (la trajectoire ne comporte pas d'obstacles à contourner). Cette approche est nommée le SLAG (Simultaneous Localization And Guidance). Elle permet d'assurer une précision sur le positionnement du robot à laquelle est associée une confiance en utilisant les mêmes hypothèses de monde plat.

2.2.4 Bilan des méthodes de localisation

Dans notre contexte de travail, pour lequel le véhicule évolue dans un environnement naturel, cette hypothèse de monde plan ne peut pas être utilisée. En effet

on ne peut pas assurer une intégrité physique si notre robot est proche d'un renversement latéral par exemple. Cette localisation doit donc comprendre toutes les coordonnées de positionnement d'un véhicule, à savoir les 3 paramètres de position notés $(X_v \ Y_v \ Z_v)$ et les 3 paramètres d'orientation (Tangage, Cap, Roulis) notés $(\alpha_v \ \beta_v \ \gamma_v)$. Pour cela, les approches de localisation absolue ne sont pas suffisantes car :

- dans le cas du GPS, elles ne fournissent pas assez de données de positionnement et nécessitent des conditions extérieures d'utilisation qui ne sont pas applicables en environnement naturel (possibilité de se trouver dans une forêt etc...)
- il n'existe pas encore de SIG utilisable assez détaillé pour réaliser cette tâche de localisation.

Les méthodes de localisation relatives ne sont pas utilisables non plus en l'état puisque le système de guidage du véhicule a besoin d'un positionnement absolu. De plus les phénomènes de dérive présents dans les techniques de localisation relative réduisent les chances de succès d'une navigation autonome.

Finalement c'est la fusion de ces deux techniques de localisation qui paraît la mieux adaptée à notre objectif. Cependant, dans notre cas, l'hypothèse de monde plat ne peut pas être appliquée. Il nous faut définir un nouveau système de localisation pour une navigation qui mette en œuvre ces deux types de localisation avec toutes les composantes nécessaires. Nous proposons donc d'estimer le déplacement relatif du véhicule par un moyen d'odométrie visuelle tout en la fusionnant avec une localisation absolue par GPS.

2.3 Odométrie visuelle pour la navigation de robots

Si on se réfère à l'être humain, notre capacité à nous localiser au sein de notre environnement est principalement basée sur nos yeux. Cet organe a de plus un double effet puisqu'il permet aussi de percevoir la géométrie du monde dans notre champ de vision. Cependant le principal inconvénient de l'œil humain est sa difficulté à déterminer le facteur d'échelle de l'environnement qu'il perçoit (inconvénient qui est détourné par l'utilisation de nos deux yeux). Le facteur d'échelle (aussi appelé la "métrique") est le facteur qui va nous permettre de transformer une distance connue sans unité en une distance en mètre. Cette métrique est très importante puisqu'elle nous permet de comparer les dimensions de notre environnement avec celles de notre propre corps (figure 2.1). Malgré ce manque de métrique, l'œil est tout à fait capable de fournir un déplacement relatif précis. Le pendant électronique de l'œil humain est la caméra projective. De nombreux travaux traitent de localisation grâce à des caméras. Cependant la détermination de la métrique reste un problème dans ce genre d'application.

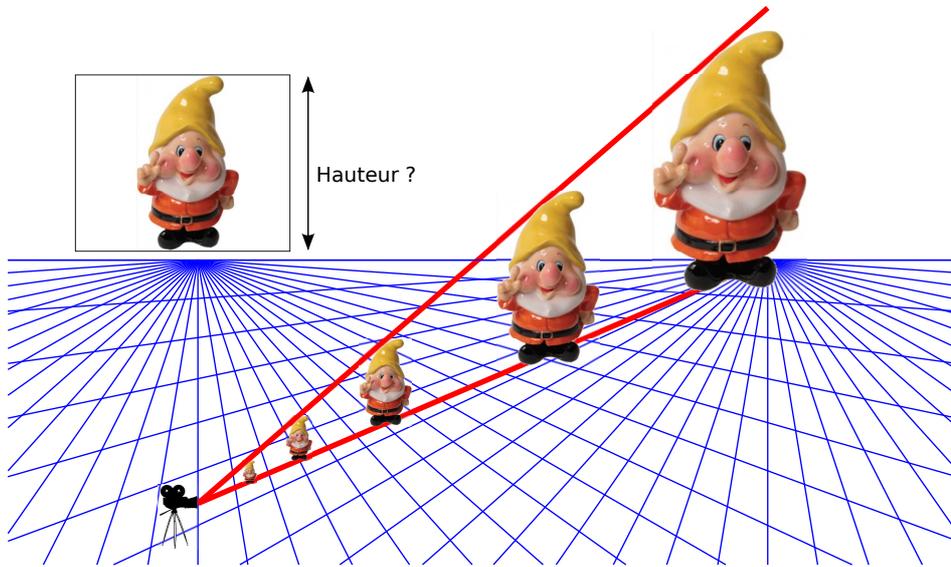


FIGURE 2.1 – Les effets de l'ignorance du facteur d'échelle.

En écartant les problèmes de métrique, les approches basées sur des techniques de vision sont intéressantes car elles permettent d'obtenir des résultats complets en termes de localisation et d'estimation de la géométrie de l'environnement avec une bonne précision : ce sont les techniques d'odométrie visuelle.

Les images successives d'une caméra sont les principales informations en odométrie visuelle. Ces images une fois traitées, servent à calculer le déplacement de la caméra entre chaque prise de vue. Pour cela, on cherche des correspondances de points dans plusieurs images consécutives. Ces correspondances de points constituent l'entrée du système qui va permettre d'obtenir le déplacement relatif de la caméra de la première image vers la deuxième. Ce déplacement peut alors être calculé de différentes façons comme nous allons le voir dans les paragraphes suivants. Il faut garder également à l'esprit que l'on doit avoir la possibilité d'estimer les points 3D utilisés comme amers et de les préciser au fur et à mesure de l'avancée du véhicule.

2.3.1 Localisation par triangulation

Dans un cadre idéal, faisons l'hypothèse que les coordonnées 3D des points de la carte sont parfaitement connues. Quelle est la méthode pour obtenir la pose qui donnerait la projection telle qu'elle est vue dans la caméra ? Le problème est une triangulation de position/orientation. Un groupe de N correspondances 2D/3D est alors nécessaire pour estimer la pose de la caméra. Le nombre N varie en fonction de la méthode utilisée et des paramètres à estimer (il faut au minimum 3 appariements 2D/3D). Haralick [Haralick *et al.*, 1994] a répertorié un ensemble

de méthodes pour estimer la pose avec $N = 3$ points, Quan [Quan et Lan, 1999] propose deux algorithmes pour respectivement $N = 4$ et $N = 5$ points. Bujnak [Bujnak et al., 2008] quant à lui propose une méthode pour déterminer la pose et la focale avec 4 points. A ce calcul analytique est souvent associé un processus de tirage aléatoire RANSAC (voir [Fischler et Bolles, 1981] où les auteurs se servent notamment de ce processus pour estimer la pose d'une caméra avec 3 points) qui va permettre de choisir le groupe de points qui minimise les erreurs de projections de tous les points appariés. Ces méthodes offrent toutes une précision très bonne de localisation, cependant leur utilisation nécessite la connaissance des points référencés dans un repère 3D. Cette connaissance préalable des points 3D est souvent obtenue par un calcul hors ligne de reconstruction comme proposé par Royer [Royer et al., 2006]. La précision de la localisation va donc dépendre de la précision de ces points 3D. De plus, dans notre cas, cette hypothèse de connaissance des points 3D n'est pas vérifiée puisqu'elle fait aussi partie des objectifs.

2.3.2 La géométrie épipolaire

La géométrie épipolaire est la géométrie qui définit les contraintes qui lient plusieurs vues d'une même scène. Soit \vec{P} un point 3D quelconque visible dans deux images de prises de vue différentes. Soit O_1 et O_2 les centres optiques de ces deux vues. Soit $\vec{p}_1 = \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$ et $\vec{p}_2 = \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}$ les projections du point \vec{P} en coordonnées homogènes respectivement dans la première et deuxième vue. La figure 2.2 montre cette configuration.

La projection d'un point quelconque \vec{P} dans l'image d'une caméra appartient au plan formé par les deux centres optiques O_1 et O_2 et le point \vec{P} .

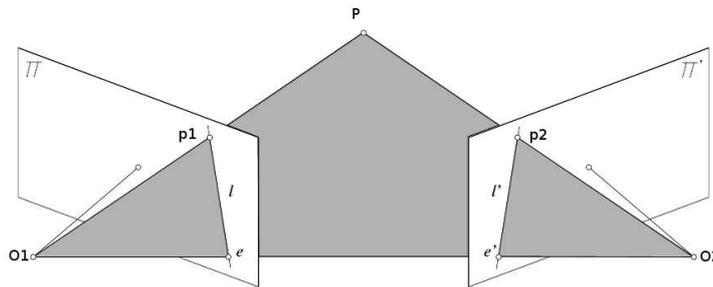


FIGURE 2.2 – La géométrie épipolaire. Source [Forsyth et Ponce, 2002].

Il en découle que la projection \vec{p}_1 de la première vue appartient au plan formé par la projection \vec{p}_2 de la deuxième vue avec les deux centres optiques O_1 et O_2 . Cette contrainte s'appelle la *contrainte épipolaire*. Elle est représentée par une ma-

trice (3×3) appelée *matrice essentielle* (notée ε) et a été introduite pour la première fois par [Longuet-Higgins, 1981]. Cette matrice contient les informations de translations et de rotations de la première caméra vers la deuxième à un facteur d'échelle près. Elle est construite à partir de la matrice de rotation \mathbf{R} et du vecteur de translation $\vec{T} = (T_1 \ T_2 \ T_3)^T$ de la première caméra vers la deuxième. Plus particulièrement, l'équation 2.1 définit cette matrice essentielle.

$$\varepsilon = [T]_{\times} \cdot \mathbf{R} \text{ où } [T]_{\times} = \begin{pmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{pmatrix} \quad (2.1)$$

On a donc la relation $\vec{p}_2^T \cdot \varepsilon \cdot \vec{p}_1 = 0$ si la contrainte épipolaire est respectée (on note T la transposée d'un vecteur ou d'une matrice). Dans le cas où les paramètres intrinsèques des caméras ne sont pas connus, une matrice analogue à la matrice essentielle existe : la matrice fondamentale (notée \mathbf{F}). La relation qui lie la matrice essentielle à la matrice fondamentale est la suivante :

$$\mathbf{F} = \mathbf{K}_1^{-T} \cdot \varepsilon \cdot \mathbf{K}_2^{-1}$$

où \mathbf{K}_j est la matrice des paramètres intrinsèques de la caméra j . Plusieurs méthodes existent pour calculer soit la matrice essentielle soit la matrice fondamentale. On peut citer notamment la méthode des 8 points introduite par [Longuet-Higgins, 1981] puis revisitée par [Hartley, 1995] pour calculer la matrice fondamentale. Pour la matrice essentielle, 5 points suffisent pour déterminer tous les paramètres (un point par paramètre de position-orientation moins un du fait de la métrique). Des méthodes pour calculer la matrice essentielle et pour en extraire les paramètres de rotations et translations ont été proposées par [Nistér, 2004, Stewénius *et al.*, 2006, Li et Hartley, 2006].

Cette façon de calculer le déplacement d'une caméra offre de bons résultats mais ne résout pas le problème de la métrique. En effet, la contrainte épipolaire est respectée même si un facteur quelconque est appliqué au vecteur de translation \vec{T} . Ce vecteur est le plus souvent obtenu avec une norme unitaire. Cette métrique peut cependant être chiffrée en utilisant des informations provenant de l'odométrie des roues par exemple. Le problème est que ce raisonnement donne des résultats approximatifs en environnement naturel puisque l'odométrie traduit la distance curviligne parcourue par la roue alors que nous avons besoin de la norme de la distance entre les deux positions de la caméra. Le relief du sol ou bien encore les phénomènes de glissement peuvent alors très facilement induire des erreurs. Aussi, l'estimation des coordonnées des points 3D ayant servi comme amers doit se faire par une étape supplémentaire.

2.3.3 L'ajustement de faisceaux

La connaissance des points en 3D est nécessaire si on veut utiliser des méthodes par triangulation. Cependant dans notre cas nous recherchons aussi cette information puisque nous considérons l'environnement inconnu. Le problème devient le suivant : comment estimer en même temps la pose de la caméra ainsi que les positions 3D de points dont les projections dans plusieurs images sont connues ? Le problème revient alors à trouver les poses \vec{C}_i et les points 3D \vec{P}_j qui minimisent les erreurs de reprojections $\min(f(C_1 \dots C_i, \vec{P}_1 \dots \vec{P}_i))$. Ce principe est communément appelé l'ajustement de faisceaux. L'ajustement de faisceaux est un problème de minimisation globale qui nécessite la connaissance de toutes les informations disponibles au moment de sa résolution. Ce problème de minimisation est souvent résolu par des algorithmes de Levenberg-Marquardt [Press *et al.*, 1992]. [Royer *et al.*, 2006] utilise cette méthode pour connaître l'ensemble des poses d'une caméra et des points 3D d'un ensemble de données pour ensuite les utiliser comme référence dans un guidage de véhicule autonome. L'inconvénient majeur de cette approche est qu'elle nécessite de connaître toutes les données au moment du calcul. Cela implique d'avoir à effectuer la minimisation sur une très grande quantité de données. [Mouragnon *et al.*, 2006] a cependant amélioré cette procédure en effectuant *en ligne* un ajustement de faisceaux local qui utilise les données des dernières images pour le calcul des poses et des points (voir figure 2.3) à une fréquence de calcul d'environ 7 images par seconde. Les auteurs de [Konolige et Agrawal, 2008] ont eux aussi proposé une méthode pour effectuer un ajustement de faisceaux en temps réel en limitant grandement le nombre de variables (états du véhicule et amers) à estimer.

Cependant, même si ce procédé est compatible avec notre objectif, il a besoin des données disponibles d'un certain nombre d'itérations précédentes. Ces informations disponibles doivent être triées en terme d'images clés ou non. L'ajustement de faisceaux est alors réalisé seulement lorsqu'une nouvelle image clé est produite. Les coordonnées des points 3D sont alors affinées seulement à chaque image clé. Un calcul de pose par triangulation est effectué pour les autres images. Malgré le fait d'effectuer l'ajustement de faisceaux seulement pour les images clés, les performances d'un tel algorithme sont à la limite du fonctionnement temps réel lors d'un rejeu et ne présentent pas d'un fonctionnement correct lors d'une expérimentation en direct sur véhicule d'expérimentation (Par exemple l'algorithme de Mouragnon fonctionne à environ 7 images par seconde).

2.3.4 Le filtrage de Kalman étendu

Comme nous l'avons vu, utiliser le système perceptif pour calculer la pose du véhicule ainsi que la géométrie de l'environnement est une tâche importante, ce-

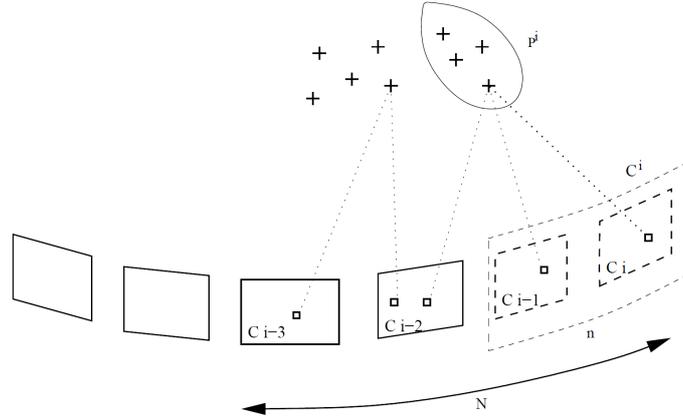


FIGURE 2.3 – L'ajustement de faisceaux local proposé par Mouragnon lorsque la prise de vue de la caméra C_i est ajoutée. Seuls les points et les caméras encadrés sont optimisés. Source : [Mouragnon *et al.*, 2006]

pendant elle est souvent coûteuse en temps de calcul. Elle doit être réalisée tout en permettant au système de guidage d'avoir suffisamment de ressource pour contrôler le véhicule. Actuellement, l'approche par ajustement de faisceaux ne permet pas cette flexibilité dans le calcul puisque tout est résolu en une passe de la méthode sans possibilité de contrôle sur les ressources ou résultats à obtenir. Il nous faut donc un moyen d'estimer à la fois la position du véhicule, son orientation mais aussi les positions 3D des points de l'image qui nous ont servi d'amers. De plus, le choix et la quantité de données à fournir à cet estimateur doit pouvoir être contrôlable. Le filtre de Kalman est un estimateur qui possède toutes ces particularités. Notre choix se porte plus particulièrement vers le filtre de Kalman étendu en raison du caractère non-linéaire de notre application. Le filtre de Kalman étendu (EKF) est un estimateur bayésien itératif qui concentre l'ensemble des informations des états précédents dans son estimation courante. L'état du filtre est représenté par deux informations : l'état estimé \vec{X} du vrai état \vec{X} et la matrice de covariance de cet état estimé P . L'état vrai du système évolue grâce à l'équation :

$$\vec{X}_{k|k-1} = f(\vec{X}_{k-1|k-1}, \vec{U}_k, \vec{Q}_k) \quad (2.2)$$

$$(2.3)$$

où \vec{U}_k est l'entrée du système, \vec{Q}_k est le bruit du processus d'évolution, $f(\vec{X}_{k-1|k-1}, \vec{U}_k, \vec{Q}_k)$ est la fonction d'évolution du système.

Le filtre de Kalman ([Mourllion *et al.*, 2006]) est découpé en deux étapes majeures :

- Prédiction : l'état *a priori* (à l'instant k) est estimé en utilisant l'état courant

(instant $k - 1$).

$$\vec{X}_{k|k-1} = f(\vec{X}_{k-1|k-1}, u_k, 0) \quad (2.4)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \cdot \mathbf{P}_{k-1|k-1} \cdot \mathbf{F}_k^T + \mathbf{Q}_k \quad (2.5)$$

où \mathbf{F}_k est la jacobienne de $f(\vec{X}_{k-1|k-1}, \vec{U}_k)$ et \mathbf{Q}_k la matrice de covariance du bruit d'évolution du processus.

- Mise à jour : l'état *a posteriori* (à l'instant k) est calculé en utilisant l'innovation de l'observation par rapport à l'état *a priori*.

$$y_k = h(\vec{X}_{k|k}) \quad (2.6)$$

$$\mathbf{S}_k = \mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T + \mathbf{R}_k \quad (2.7)$$

$$\mathbf{G}_k = \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T \cdot \mathbf{S}_k^{-1} \quad (2.8)$$

$$\vec{X}_{k|k} = \vec{X}_{k|k-1} + \mathbf{G}_k \cdot (y_k - h(\vec{X}_{k|k-1})) \quad (2.9)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{G}_k \cdot \mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \quad (2.10)$$

où y_k est la mesure de l'état, h est la fonction d'observation de l'état, \mathbf{H}_k sa jacobienne à l'instant k , \mathbf{R}_k la covariance du bruit d'observation et \mathbf{G}_k le gain de Kalman.

Ces dernières années le filtre de Kalman a été utilisé dans beaucoup d'applications de diverses natures. Dans notre cas de perception/navigation le filtre de Kalman a commencé à être utilisé dans un contexte de localisation et cartographie simultanée (SLAM : Simultaneous Localization And Mapping en anglais). Par exemple dans [Dissanayake *et al.*, 2001] les auteurs utilisent ce filtre pour estimer la position et le cap (X, Y, θ) d'un véhicule dans une carte 2D constituée d'amers (A_{x_i}, A_{y_i}) . En ce qui concerne le SLAM par vision, Davison ([Davison, 2003]) a été un des tout premiers à mettre en œuvre une application de ce type fonctionnant en temps réel. Ces approches de type SLAM ont aussi été utilisées avec des données provenant de caméras stéréo ([Lemaire *et al.*, 2007]).

Le filtre de Kalman étendu est bien adapté à notre problème puisqu'il permet d'estimer la pose courante du véhicule ainsi que les coordonnées des amers avec l'ancien état seulement. De plus, il se prête convenablement à l'exploitation de données proprioceptives et à la fusion de données extéroceptives.

2.3.5 Le filtre d'information

Il existe d'autres filtres se rapprochant du filtre de Kalman. Dans le filtre d'information par exemple ([Gibbens *et al.*, 2000]), les informations du système sont contenues dans le vecteur d'information \vec{y} et la matrice d'information \mathbf{Y} . Les relations qui lient le vecteur d'état avec le vecteur d'information et la matrice de covariance avec la matrice d'information sont les suivantes :

$$\begin{aligned} \mathbf{Y} &= \mathbf{P}^{-1} \\ \vec{y} &= \mathbf{P}^{-1} \cdot \vec{X} \end{aligned} \quad (2.11)$$

L'avantage avec le filtre d'information est que plusieurs mesures peuvent être prises en compte indépendamment les unes des autres simplement en ajoutant leurs effets au vecteur d'informations et à la matrice d'information. De ce fait, si N mesures ont été effectuées il suffit d'ajouter ces N informations au vecteur d'état et à la matrice d'information. Notre système dans son ensemble (perception/action) va dialoguer avec le processus de guidage afin de lui fournir les informations dont il a besoin pour évoluer. Parmi ces informations à transmettre se trouvent les informations de localisation et des amers 3D contenues dans le vecteur d'état et leurs incertitudes. Avec ce filtre il nous serait alors nécessaire d'effectuer la conversion du vecteur d'information et de la matrice d'information à chaque demande ce qui nécessite d'inverser une matrice et donc une consommation des ressources supplémentaire.

2.3.6 Le filtre à particules

Une autre façon de résoudre le problème est de tester une multitude d'hypothèses à la manière d'un tirage aléatoire de type Monte-Carlo, c'est le filtre à particules. Pour cela, on effectue un échantillonnage aléatoire de l'état suivant sa densité de probabilité supposée connue. Chacun de ces tirages aléatoires constitue une *particule*. Un tutoriel sur le filtre à particule a été publié par Arulampalam [Arulampalam *et al.*, 2001]. Puis, les observations sont utilisées pour pondérer chacune de ces particules en terme de probabilité afin de leur donner plus d'importance. Si suffisamment de particules sont créées, celle qui possède le plus gros score est une bonne estimation du vecteur d'état. Cependant cette technique nécessite de faire un tirage aléatoire sur l'ensemble de l'état. Or la position d'un amer dans la carte est dépendante de la pose du véhicule. C'est grâce à cette information que les algorithmes FastSLAM 1.0 ([Montemerlo *et al.*, 2002]) et FastSLAM 2.0 ([Montemerlo *et al.*, 2003]) ont vu le jour. L'idée du FastSLAM est de générer un échantillonnage sur la trajectoire du véhicule uniquement, les amers étant générés conditionnellement à cette trajectoire. Pour chacune de ces trajectoires un filtre de Kalman étendu par amer est appliqué. L'innovation apportée par FastSLAM 2 réside dans l'amélioration de la qualité des échantillons générés. Cette amélioration a donc pour effet de diminuer le nombre de particules nécessaires. Si on met de côté les problèmes d'inconsistance du FastSLAM ([Bailey *et al.*, 2006b]), il reste le problème de savoir quelles informations nous allons délivrer au système de guidage pour son fonctionnement. Le fait de renvoyer la particule présentant la plus grosse probabilité peut provoquer des sauts dans l'estimation de la trajectoire

puisque l'on peut changer de particule à tout instant. Il faut aussi générer un nombre de particules suffisamment important, chacune de ces particules doit être vérifiée. Aussi, [Bailey *et al.*, 2006b] conclut dans son article que le FastSLAM serait plus approprié à l'initialisation de la carte sur une courte période.

2.3.7 Bilan des méthodes d'odométrie visuelle

Nous avons vu que la meilleure approche pour effectuer une localisation par vision serait de faire une triangulation avec des points 3D connus. Cependant nous n'avons pas ces informations à notre disposition puisque nous avons aussi comme objectif d'estimer ces amers 3D. Les techniques de géométrie épipolaire n'ont pas besoin de cette connaissance pour pouvoir être mises en œuvre mais les mêmes problèmes d'estimation des points 3D apparaissent ici aussi. De plus, associer une métrique correcte au calcul du déplacement est ambiguë. Les méthodes d'ajustement de faisceaux répondent au problème donné mais ont été créées à mon sens pour des calculs hors ligne de pose de robots ou de construction de carte. Même si des méthodes locales d'ajustement de faisceaux existent, leurs utilisations ne sont pas faites pour des applications temps réel. Pour les filtres à particules le problème est de savoir quelle est l'information de positionnement que nous allons transmettre au processus de guidage. En effet les lois de commande associées au système de guidage sont faites pour fonctionner avec une localisation uni-modale, le changement de modalité peut avoir un effet indésirable sur le système. Le filtre de Kalman étendu ainsi que le filtre d'informations sont assez proches l'un de l'autre. Ils proposent tout les deux une estimation uni-modale avec estimation de la métrique. Les filtres d'informations ne sont pas adaptés à un système complet de perception et guidage puisque l'information de localisation n'est pas accessible directement, il faut passer du vecteur d'information au vecteur d'état pour connaître la position du véhicule et cette étape peut être très consommatrice de ressources si beaucoup d'informations sont présentes dans le vecteur d'état. Finalement, le filtre de Kalman permet de satisfaire tous les aspects de notre objectif en une seule étape tout en permettant une évolution en temps réel (cadence de traitement et traitement sur flux de données direct). Nous utiliserons donc le filtre de Kalman étendu dans la suite de ce mémoire.

2.4 Odométrie visuelle par filtrage de Kalman étendu

Afin d'effectuer notre odométrie visuelle, plaçons nous dans un cadre d'estimation de pose d'une caméra monoculaire. Pour mettre en place le filtre de Kalman, nous avons besoin de connaître les équations d'état et les équations d'observations

de notre système. Dans notre cas, les observations de l'état courant du véhicule vont se faire grâce à la caméra. Le vecteur d'état qui constitue la pose du véhicule dans le monde ainsi que les points 3D à estimer est donc composé des 6 paramètres de positionnement (un vecteur noté \vec{X}_{v-m}) et des coordonnées de chaque point 3D nécessaire (appelé *amer*) notées \vec{P}_n :

$$\begin{aligned}\vec{X}_{v-m} &= (X_v \ Y_v \ Z_v \ \alpha_v \ \beta_v \ \gamma_v)^T = \begin{pmatrix} \vec{T}_{(v-m)} \\ \vec{O}_{(v-m)} \end{pmatrix} \\ \vec{P}_n &= (P_{x_n} \ P_{y_n} \ P_{z_n})^T\end{aligned}\tag{2.12}$$

où :

- $\vec{T}_{(v-m)} = (X_v \ Y_v \ Z_v)^T$ sont les positions respectivement sur X, Y, Z du véhicule dans le repère monde.
- $\vec{O}_{(v-m)} = (\alpha_v \ \beta_v \ \gamma_v)^T$ sont les paramètres respectivement de tangage, lacet, roulis du véhicule dans le repère monde par rapport aux axes x, y, z.
- $\vec{P}_n = (P_{x_n} \ P_{y_n} \ P_{z_n})^T$ sont les coordonnées de l'amer n ($n \in [1, N]$) pour respectivement sa position en X, Y, Z dans le repère monde.

Le vecteur d'état entier de notre système est alors de la forme :

$$\vec{X} = \left(\vec{T}_{(v-m)}^T \ \vec{O}_{(v-m)}^T \ \vec{P}_1^T \ \dots \ \vec{P}_n^T \ \dots \ \vec{P}_N^T \right)^T$$

où N est le nombre de points présents dans notre carte.

La position estimée du véhicule se situe sur le sol au milieu de l'essieu arrière. La caméra quant à elle se situe à l'avant du véhicule et en hauteur. Elle est considérée fixe par rapport au châssis du véhicule ; de ce fait une transformation rigide existe entre le repère véhicule et le repère caméra. Ce changement de repère est défini par une rotation $\mathbf{R}_{(c-v)}$ et une translation $\vec{T}_{(c-v)}$.

Afin de mettre à jour le vecteur d'état de notre système, une observation de celui-ci doit se faire, ici, au moyen de la caméra. Pour cela nous devons donc définir sous quelle forme seront observés les points dans la caméra. La nature de l'observation a une grande importance dans un EKF. En effet, la jacobienne de ces équations d'observations doit être fournie au filtre pour mettre à jour l'état *a posteriori* et sa covariance. Cependant le fait d'approximer la fonction d'observation par une fonction linéaire (en calculant la jacobienne) amène une non-intégrité des estimations dans le filtre de Kalman. Une synthèse des causes et conséquences de l'inconsistance est proposée par [Joly, 2010]. Cette non-intégrité est très pénalisante puisque **la vraie position du véhicule n'appartient pas à l'ellipse de confiance de l'état**. Cette incertitude de localisation peut alors être fatale au véhicule et à ses occupants suivant le contexte.

Ces équations doivent être formalisées d'une façon la plus linéaire possible pour rester le plus intègre possible. Aussi nous proposons de définir deux façons possibles pour repérer un point 3D dans l'espace visible de la caméra :

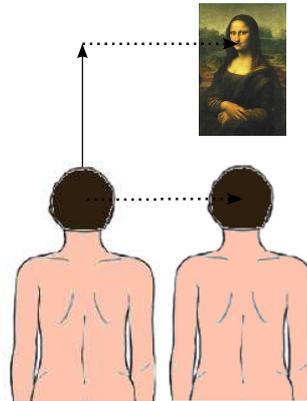


FIGURE 2.4 – Première façon d’observer un point : coordonnée cartésienne

- en coordonnées cartésiennes : vue de la projection comme une coordonnée (u, v) .
- en coordonnées sphériques : vue du point comme 2 angles (ϕ, ψ) partant du centre optique vers le point observé.

Ces 2 façons de décrire un point ont été définies parce qu’il nous semble pertinent d’utiliser des équations qui semblent plus linéaires suivant tels ou tels paramètres. Par exemple, on peut aisément penser qu’une variation de coordonnée de pixel dans une image amène une information plus pertinente en variation de coordonnée de localisation x, y, z . De même il semble plus judicieux de mettre à jour notre état en lacet-tangage par une mesure de variation d’angle plutôt que de pixel. Nous allons donc présenter dans ce mémoire différentes façons de représenter un point dans un repère caméra. Le choix de la représentation d’un point fait encore l’objet de travaux. Dans ce mémoire nous exposerons brièvement les représentations.

2.4.1 Observation de points en coordonnées cartésiennes

La façon la plus évidente pour repérer un point dans l’espace caméra est d’utiliser ses coordonnées pixeliques. Ces travaux ont été montrés dans [Malartre *et al.*, 2009a].

Équation d’observation

Soient $\mathbf{R}_x(\alpha_v)$, $\mathbf{R}_y(\beta_v)$ et $\mathbf{R}_z(\gamma_v)$ les trois matrices de rotation élémentaires respectivement suivant les axes x, y et z .

$$\mathbf{R}_x(\alpha_v) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_v) & -\sin(\alpha_v) \\ 0 & \sin(\alpha_v) & \cos(\alpha) \end{pmatrix}$$

$$\begin{aligned}\mathbf{R}_y(\beta_v) &= \begin{pmatrix} \cos(\beta_v) & 0 & -\sin(\beta_v) \\ 0 & 1 & 0 \\ \sin(\beta_v) & 0 & \cos(\beta_v) \end{pmatrix} \\ \mathbf{R}_z(\gamma_v) &= \begin{pmatrix} \cos(\gamma_v) & -\sin(\gamma_v) & 0 \\ \sin(\gamma_v) & \cos(\gamma_v) & 0 \\ 0 & 0 & 1 \end{pmatrix}\end{aligned}\quad (2.13)$$

Soit $\mathbf{R}_{(v-m)} = \mathbf{R}_y(\beta_v) \cdot \mathbf{R}_x(\alpha_v) \cdot \mathbf{R}_z(\gamma_v)$ la matrice de rotation du repère véhicule \mathcal{R}_v vers le repère monde \mathcal{R}_m . De même, nous avons besoin de $\mathbf{R}_{(c-v)} = \mathbf{R}_y(\beta_c) \cdot \mathbf{R}_x(\alpha_c) \cdot \mathbf{R}_z(\gamma_c)$ qui est la matrice de rotation du repère caméra \mathcal{R}_c vers le repère véhicule \mathcal{R}_v et de $\vec{T}_{(c-v)} = (T_x \ T_y \ T_z)^T$ qui est le vecteur de translation de la caméra vers le véhicule. Un point du monde $\vec{P} = (P_x \ P_y \ P_z)^T$ sera observé dans le repère caméra par la relation :

$$\vec{P}_c = \mathbf{K} \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)}) = (x_c \ y_c \ z_c)^T$$

où \mathbf{K} est la matrice des paramètres intrinsèques de la caméra :

$$\mathbf{K} = \begin{pmatrix} f \cdot k_x & 0 & C_x \\ 0 & f \cdot k_y & C_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \vec{K}_1 \\ \vec{K}_2 \\ \vec{K}_3 \end{pmatrix}$$

avec f la distance focale (en mètres), k_u et k_v les facteurs d'agrandissement de l'image (en pixels/mètres) et C_x et C_y les coordonnées du centre optique de la caméra (en pixels). Dans le cas d'une caméra perspective, nous avons :

$$u = \frac{x_c}{z_c} \quad \text{and} \quad v = \frac{y_c}{z_c} \quad (2.14)$$

z_c est dans ce cas l'axe de projection (voir figure 2.5). En gardant un formalisme des équations d'observations sous forme matricielle comme proposé par Féraud [Féraud *et al.*, 2010] et utilisé dans [Malartre *et al.*, 2009a] nous obtenons les équations 2.15 et 2.16 :

$$u = \frac{\vec{K}_1 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{\vec{K}_3 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} = \frac{N_u}{D_{uv}} \quad (2.15)$$

$$v = \frac{\vec{K}_2 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{\vec{K}_3 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} = \frac{N_v}{D_{uv}} \quad (2.16)$$

Où \vec{K}_i représente la $i^{\text{ème}}$ ligne de \mathbf{K} .

Nous avons donc la relation qui relie l'observation dans le repère caméra $\vec{y}_i = (\hat{u}_i, \hat{v}_i)^T$ du $n^{\text{ème}}$ point $\vec{P}_n = (x_i, y_i, z_i)$ avec le vecteur d'état \vec{X} .

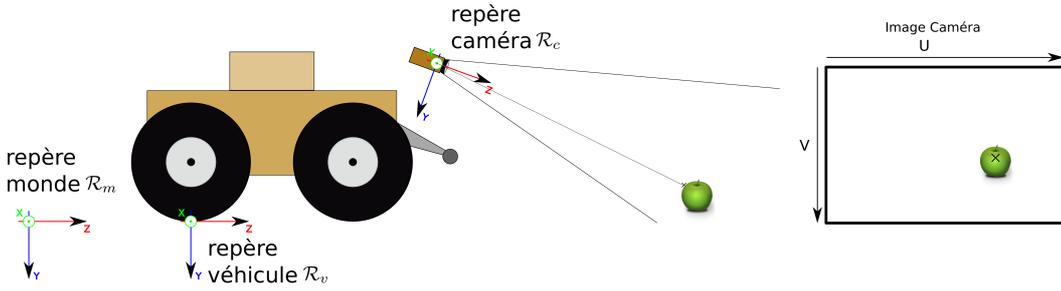


FIGURE 2.5 – Observation d'un amer dans la caméra.

Ce qui implique :

$$\vec{y}_i = \begin{pmatrix} \hat{u}_i \\ \hat{v}_i \end{pmatrix} = \mathbf{h}(\vec{X}) + \vec{\varepsilon}_i = \begin{pmatrix} u_i + \varepsilon_{ui} \\ v_i + \varepsilon_{vi} \end{pmatrix}$$

avec $\vec{\varepsilon}_i = \begin{pmatrix} \varepsilon_{ui} \\ \varepsilon_{vi} \end{pmatrix}$ qui est le bruit de mesure d'un pixel dans la caméra et qui sera considéré comme un bruit blanc gaussien d'écart type 1 pixel. Nous disposons à présent de la fonction d'observation de l'état dans la caméra. Le filtre de Kalman étendu a besoin de la jacobienne de cette fonction pour projeter les incertitudes de l'état dans l'espace d'observation. La jacobienne d'une fonction quelconque $f(x_1, \dots, x_n)$ est la matrice des dérivées partielles du premier ordre $J_f = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \dots & \frac{\partial f}{\partial x_n} \end{pmatrix}$.

Jacobienne de la position

Le filtre de Kalman étendu a besoin de la jacobienne de cette observation pour mettre à jour le vecteur d'état ainsi que la covariance. Prenons le cas du calcul de la jacobienne d'observation d'un point quelconque $P = (x \ y \ z)$. Cette jacobienne est représentée dans l'équation 2.17.

$$\mathbf{J}_{uv} = \begin{pmatrix} \frac{\partial u}{\partial X_v} & \frac{\partial u}{\partial Y_v} & \frac{\partial u}{\partial Z_v} & \frac{\partial u}{\partial \alpha_v} & \frac{\partial u}{\partial \beta_v} & \frac{\partial u}{\partial \gamma_v} & \dots & \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial X_v} & \frac{\partial v}{\partial Y_v} & \frac{\partial v}{\partial Z_v} & \frac{\partial v}{\partial \alpha_v} & \frac{\partial v}{\partial \beta_v} & \frac{\partial v}{\partial \gamma_v} & \dots & \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{pmatrix} \quad (2.17)$$

Nous avons donc besoin des équations d'observations (2.15) et (2.16) pour calculer ces jacobienes. Prenons le cas de $\frac{\partial u}{\partial X_v}$:

$$\frac{\partial u}{\partial X_v} = \frac{D_{uv} \frac{\partial N_u}{\partial X_v} - N_u \frac{\partial D_{uv}}{\partial X_v}}{D_{uv}^2} = K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

avec $K_u = \frac{D_{uv}\vec{K}_1 - N_u\vec{K}_3}{D_{uv}^2}$. Nous en déduisons alors $\frac{\partial u}{\partial Y}$ et $\frac{\partial u}{\partial Z}$.

$$\frac{\partial u}{\partial Y_v} = K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

$$\frac{\partial u}{\partial Z_v} = K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Les dérivées partielles $\frac{\partial v}{\partial X_v}$, $\frac{\partial v}{\partial Y_v}$ et $\frac{\partial v}{\partial Z_v}$ sont calculées de la même façon avec $K_v = \frac{D_{uv}\vec{K}_2 - N_v\vec{K}_3}{D_{uv}^2}$.

Jacobienne de l'orientation

Le même principe peut être appliqué aux calculs des jacobiennes de l'orientation. Reprenons l'équation (2.15) pour l'angle de tangage α_v :

$$\begin{aligned} \frac{\partial u}{\partial \alpha_v} &= \frac{D_{uv} \frac{\partial N_u}{\partial \alpha_v} - N_u \frac{\partial D_{uv}}{\partial \alpha_v}}{D_{uv}^2} \\ \frac{\partial u}{\partial \alpha_v} &= \frac{D_{uv}\vec{K}_1 \mathbf{R}_{(c-v)}^T \frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} (\vec{P} - \vec{T}_{(v-m)}) - N_u \vec{K}_3 \mathbf{R}_{(c-v)}^T \frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} (\vec{P} - \vec{T}_{(v-m)})}{De^2} \\ &= K_u \mathbf{R}_{(c-v)}^T \frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} (\vec{P} - \vec{T}_{(v-m)}) \end{aligned} \quad (2.18)$$

Il peut être montré que :

$$\frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} = \left(\frac{\partial \mathbf{R}_{(v-m)}}{\partial \alpha_v} \right)^T = [\mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v)]^T \quad (2.19)$$

ce qui nous donne :

$$\frac{\partial u}{\partial \alpha_v} = K_u \mathbf{R}_{(c-v)}^T [\mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v)]^T (\vec{P}_n - \vec{T}_{(v-m)})$$

Nous pouvons calculer de la même façon $\frac{\partial u}{\partial \beta_v}$, $\frac{\partial u}{\partial \gamma_v}$, $\frac{\partial v}{\partial \alpha_v}$, $\frac{\partial v}{\partial \beta_v}$, $\frac{\partial v}{\partial \gamma_v}$.

Jacobienne du point 3D

Maintenant nous avons besoin des dérivées partielles de l'observation par rapport à la position des amers 3D. En observant les équations 2.15 et 2.16, on peut remarquer que $\frac{\partial u}{\partial \vec{P}} = -\frac{\partial u}{\partial \vec{T}_{(v-m)}}$ et $\frac{\partial v}{\partial \vec{P}} = -\frac{\partial v}{\partial \vec{T}_{(v-m)}}$. Les dérivées partielles sont alors évidentes :

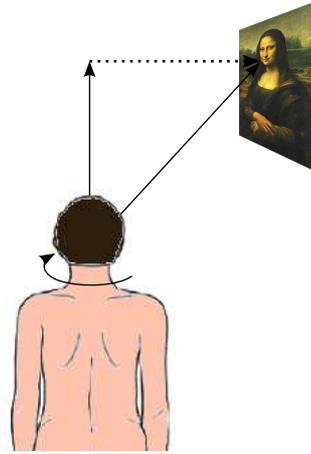


FIGURE 2.6 – Deuxième façon d’observer un point : coordonnée sphérique

$$\begin{aligned} \frac{\partial u}{\partial P_{xi}} &= -\frac{\partial u}{\partial X_v} \cdot \frac{\partial u}{\partial P_{yi}} = -\frac{\partial u}{\partial Y_v} \cdot \frac{\partial u}{\partial P_{zi}} = -\frac{\partial u}{\partial Z_v} \\ \frac{\partial v}{\partial P_{xi}} &= -\frac{\partial v}{\partial X_v} \cdot \frac{\partial v}{\partial P_{yi}} = -\frac{\partial v}{\partial Y_v} \cdot \frac{\partial v}{\partial P_{zi}} = -\frac{\partial v}{\partial Z_v} \end{aligned} \quad (2.20)$$

Le détails des calculs est montré en annexe A.1.

Conclusion

Nous venons de voir comment écrire facilement les équations d’observations du système sous forme matricielle. Cette observation de projection de points 3D dans l’espace caméra nous permet d’associer un déplacement du véhicule et/ou des coordonnées 3D de l’amer avec une variation de pixel dans l’image. Le calcul de la jacobienne de cette observation nous permettra alors d’estimer ces composantes.

2.4.2 Observation de points en coordonnées sphériques

Dans cette façon de représenter un point, l’information est contenue dans les angles formés par le point 3D, le centre optique de la caméra et la focale à la manière d’un Bearing-Only SLAM. Cette méthode est semblable à celle utilisée dans [Lemaire *et al.*, 2005]. Grâce à cette méthode les angles de tangage et de lacet devraient être mieux estimés. Pour illustrer ces propos on peut voir un exemple en figure 2.6 où on peut se rendre compte qu’il faut tourner la tête pour mieux voir le tableau et non pas se déplacer latéralement. Ce constat nous a motivé pour mettre en place cette façon d’observer un amer dans la caméra. Le début des équations est très similaire à la première façon d’observer un point à un détail près. Comme nous allons mesurer des angles, les calculs à effectuer sont à faire avec un repère orthonormé. En d’autres termes, la matrice de paramètres intrinsèques n’est pas utilisée dans ce cas là.

$$\vec{P}_c = \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)}) = (x_c \ y_c \ z_c)^T$$

Ici nous cherchons à observer les angles formés par le point 3D et le centre optique de la caméra. Ces angles sont définis par :

$$\begin{aligned} \begin{pmatrix} \phi \\ \psi \end{pmatrix} &= \begin{pmatrix} \arctan \left(\frac{x_c}{z_c} \right) \\ \arctan \left(\frac{y_c}{z_c} \right) \end{pmatrix} = \begin{pmatrix} \arctan \left(\frac{(1 \ 0 \ 0) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{(0 \ 0 \ 1) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} \right) \\ \arctan \left(\frac{(0 \ 1 \ 0) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{(0 \ 0 \ 1) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} \right) \end{pmatrix} \\ &= \begin{pmatrix} \arctan \left(\frac{N_\phi}{D_{\phi\psi}} \right) \\ \arctan \left(\frac{N_\psi}{D_{\phi\psi}} \right) \end{pmatrix} \end{aligned} \quad (2.21)$$

Une fois les équations d'observations écrites il faut calculer leurs jacobiennes.

Jacobienne de la position

La méthodologie pour calculer ces jacobiennes est la même que celle du paragraphe 2.4.1. Prenons le cas de $\frac{\partial \phi}{\partial X_v}$:

$$\frac{\partial \phi}{\partial X_v} = \frac{D_{\phi\psi} \frac{\partial N_\phi}{\partial X_v} - N_\phi \frac{\partial D_{\phi\psi}}{\partial X_v}}{D_{\phi\psi}^2 + N_\phi^2} = K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

avec $K_\phi = \frac{D_{\phi\psi} (1 \ 0 \ 0) - N_\phi (0 \ 0 \ 1)}{D_{\phi\psi}^2 + N_\phi^2}$. Nous en déduisons alors $\frac{\partial \phi}{\partial Y_v}$ et $\frac{\partial \phi}{\partial Z_v}$.

$$\frac{\partial \phi}{\partial Y_v} = K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

$$\frac{\partial \phi}{\partial Z_v} = K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Le résultat pour les dérivées partielles $\frac{\partial \psi}{\partial X_v}$, $\frac{\partial \psi}{\partial Y_v}$ et $\frac{\partial \psi}{\partial Z_v}$ sont trouvés de la même façon avec $K_\psi = \frac{D_{\phi\psi} (0 \ 1 \ 0) - N_\psi (0 \ 0 \ 1)}{D_{\phi\psi}^2 + N_\psi^2}$.

Jacobienne de l'orientation

Toujours en reprenant le même principe que pour l'observation en coordonnées cartésiennes on peut obtenir les dérivées partielles des angles ϕ et ψ par rapport

aux angles α_v , β_v et γ_v . Reprenons l'équation 2.21 en ϕ pour l'angle de tangage α_v :

$$\begin{aligned}
\frac{\partial \phi}{\partial \alpha_v} &= \frac{D_{\phi\psi} \frac{\partial N_\phi}{\partial \alpha_v} - N_\phi \frac{\partial D_e}{\partial \alpha_v}}{D_{\phi\psi}^2 + N_\phi^2} \\
&= \frac{D_{\phi\psi} (1 \ 0 \ 0) \mathbf{R}_{(c-v)}^T \frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} (\vec{P}_n - \vec{T}_{(v-m)})}{D_{\phi\psi}^2 + N_\phi^2} \\
&\quad - \frac{N_\phi (0 \ 0 \ 1) \mathbf{R}_{(c-v)}^T \frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} (\vec{P}_n - \vec{T}_{(v-m)})}{D_{\phi\psi}^2 + N_\phi^2} \\
&= K_\phi \mathbf{R}_{(c-v)}^T \frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} (\vec{P}_n - \vec{T}_{(v-m)}) \tag{2.22}
\end{aligned}$$

En utilisant l'équation 2.19 nous obtenons :

$$\frac{\partial \phi}{\partial \alpha_v} = K_\phi \mathbf{R}_{(c-v)}^T [\mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v)]^T (\vec{P}_n - \vec{T}_{(v-m)})$$

Nous pouvons calculer de la même façon $\frac{\partial u}{\partial \beta_v}$, $\frac{\partial u}{\partial \gamma_v}$, $\frac{\partial v}{\partial \alpha_v}$, $\frac{\partial v}{\partial \beta_v}$, $\frac{\partial v}{\partial \gamma_v}$.

Jacobienne du point 3D

De la même façon que pour le premier type d'observation on a :

$$\begin{aligned}
\frac{\partial \phi}{\partial P_{xi}} &= -\frac{\partial \phi}{\partial X_v} \cdot \frac{\partial \phi}{\partial P_{yi}} = -\frac{\partial \phi}{\partial Y_v} \cdot \frac{\partial \phi}{\partial P_{yi}} = -\frac{\partial \phi}{\partial Z_v} \\
\frac{\partial \psi}{\partial P_{xi}} &= -\frac{\partial \psi}{\partial X_v} \cdot \frac{\partial \psi}{\partial P_{yi}} = -\frac{\partial \psi}{\partial Y_v} \cdot \frac{\partial \psi}{\partial P_{yi}} = -\frac{\partial \psi}{\partial Z_v}
\end{aligned} \tag{2.23}$$

Conclusion

Nous pouvons voir dans cette façon d'observer un point avec des angles que les équations sont très ressemblantes à celles de l'observation avec des coordonnées cartésiennes. Cependant ces équations mettent en correspondance un déplacement du véhicule avec cette fois-ci une variation d'angle dans l'image. Cette différence devrait amener une meilleure estimation des angles de lacet et de tangage.

2.4.3 Conclusion

Dans cette section nous avons vu comment, à partir d'une caméra monoculaire, nous pouvions estimer le déplacement du robot auquel elle est attachée. Cependant en introduction nous avons évoqué le fait que déterminer la métrique avec une seule caméra n'était pas très précis et pas très stable sur du long terme. De

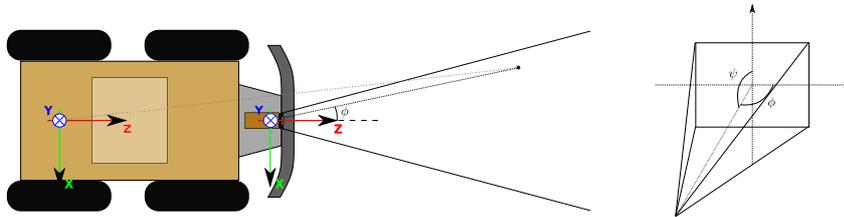


FIGURE 2.7 – Observation d'angle dans une caméra

plus, la localisation est également utilisée pour reconstruire la carte locale de l'environnement du véhicule au fur et à mesure de son avancée. L'odométrie visuelle est capable de reconstruire l'environnement dans ce même processus (c'est le principe de la localisation et cartographie simultanée SLAM) mais la précision de la reconstruction obtenue dépend aussi de la précision de la localisation et donc de la métrique. Cette information est donc très importante dans notre système. Aussi nous avons cherché un moyen de renseigner cette métrique le plus tôt possible.

2.5 Utilisation de données Lidar pour l'aide à la localisation

2.5.1 Introduction

Dans notre application de navigation autonome, nous cherchons à estimer la position de notre véhicule mais aussi à reconstruire l'environnement utile à notre système. Dans un premier temps, nous pouvons dire que l'environnement utile se situe *a priori* sur le sol et à l'avant du véhicule. La reconstruction du monde doit se faire à cet endroit.

Dans les systèmes SLAM, les auteurs utilisent souvent des amers visuels remarquables. Ce sont, la plupart du temps, des points de l'image présentant un fort gradient. Harris et Lowe ([Harris et Stephens, 1988, Lowe, 1999]) ont présenté des façons d'extraire ces points remarquables d'une image. Or, si on veut déterminer la géométrie du sol, l'environnement ne propose pas forcément des points d'intérêt se situant dans une zone pertinente aux yeux de notre application. La figure 2.8 montre des points de Harris extraits de la zone pertinente de l'environnement que l'on doit reconstruire. Dans une telle zone le système pourrait avoir besoin de milliers de points pour être suffisamment précis, cependant la zone montrée n'en compte pas autant.

D'autres problèmes surviennent lors de l'utilisation de ces amers dans l'odométrie visuelle. L'initialisation de la position de ces amers dans le filtre de Kalman doit être la plus proche possible de la vérité afin de faire converger le

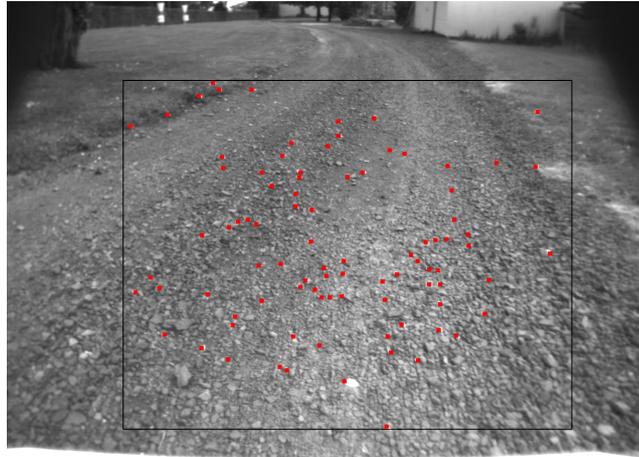


FIGURE 2.8 – Points de Harris extrait de l'image vis-à-vis de la zone d'intérêt.

filtre. Pour des amers pouvant se trouver à une distance inconnue de la caméra [Lemaire *et al.*, 2005] utilise une somme de gaussiennes comme expliqué par [Kwok *et al.*, 2005]. Ensuite il faut choisir la gaussienne qui approxime le mieux l'amer. Cette initialisation pourrait être évitée si une estimation de la distance entre l'amer et la caméra était connue.

Une solution pour initialiser la géométrie de cet endroit serait d'utiliser un système stéréo. Un tel système fournirait suffisamment d'information à l'odométrie visuelle pour achever la localisation. Cependant la qualité d'une reconstruction stéréoscopique dépend de beaucoup de critères :

- les deux caméras doivent être bien synchronisées afin de bien faire correspondre l'image de droite avec l'image de gauche.
- l'environnement doit être suffisamment texturé pour que les correspondances droite-gauche soient faites.
- la calibration des paramètres intrinsèques et extrinsèques doit être très précise.

De plus la précision de la carte obtenue décroît avec le carré de la distance. Aussi nous avons choisi de ne pas utiliser un système stéréoscopique.

Cependant nous devons toujours trouver un moyen d'accéder à la géométrie du sol aux endroits pertinents et de façon précise pour aider le système de localisation à remplir son rôle. L'objectif est donc d'utiliser, en plus de notre caméra, un capteur qui fournisse une mesure de profondeur précise, dense si possible, et qui puisse être orientée vers un endroit utile. Un capteur de profondeur existe et est connu pour sa grande précision en terme de mesure : le télémètre laser. Fremont ([Fremont, 2009]) parle dans ses travaux des bienfaits que pourrait apporter une collaboration entre un Lidar et une caméra pour des approches d'odométrie

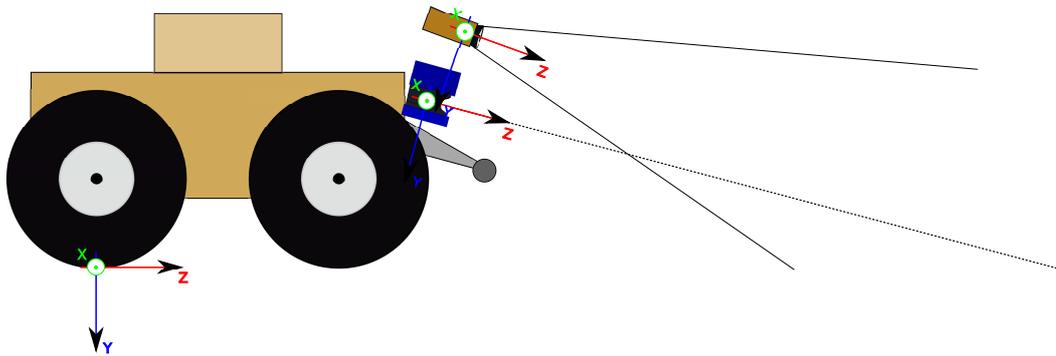


FIGURE 2.9 – Le système caméra-télemètre

visuelle. La complémentarité de ces capteurs est un atout important au vu de l'objectif de localisation et de perception de l'environnement. En effet, la caméra est capable de voir et de revoir des informations qui couvrent une zone importante mais la précision de ces informations dépend de leurs positions dans la scène. Le télémètre quant à lui permet de percevoir une information avec une précision "indépendante" de sa distance par rapport au capteur mais il est incapable de revoir cette même information si le capteur est en mouvement.

En faisant l'hypothèse de la visibilité du point télémètre dans la caméra on aurait alors une très bonne estimation de la profondeur d'un d'amer projeté dans l'image. Cette bonne connaissance de l'amer est une aide lors du calcul de la pose du véhicule en rendant la métrique estimée beaucoup plus précise (l'information de métrique est estimée seulement grâce aux capteurs proprioceptifs dans le cas d'un SLAM monoculaire classique).

Il est possible d'utiliser les points issus du télémètre dans la caméra si on dispose de la transformation rigide qui lie le repère télémètre au repère caméra (voir figure 2.9). Ces points, une fois positionnés dans le repère caméra, peuvent être utilisés comme n'importe quel amer pour former une observation utile au filtre de Kalman. Cette approche devrait permettre de résoudre simultanément le problème de la localisation de notre robot et de la richesse de la reconstruction du terrain.

2.5.2 Odométrie visuelle par couplage Lidar/Caméra

2.5.2.1 Principe

Le but du couplage Lidar/Caméra est d'utiliser les informations précises du télémètre dans le processus d'odométrie visuelle. Pour cela nous allons chercher à positionner les données télémétriques dans un autre repère : celui de la caméra. Nous avons alors, grâce à la fonction d'observation de la caméra, à la fois la po-

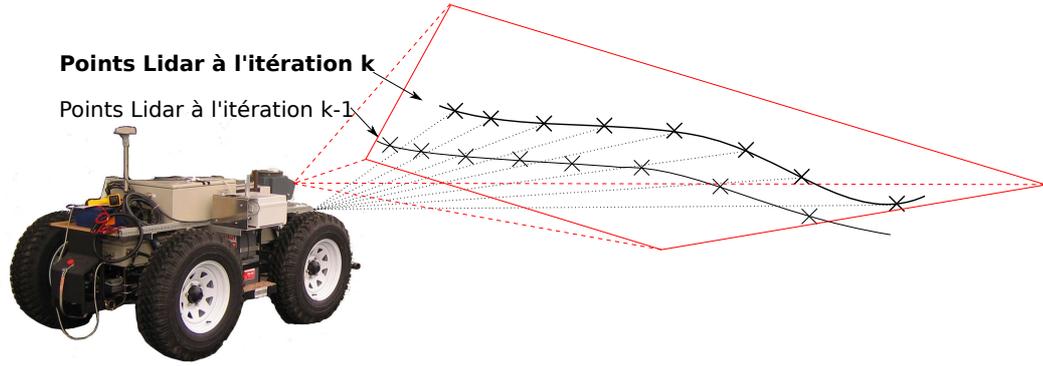


FIGURE 2.10 – Le couplage Lidar-Caméra : utiliser les informations du télémètre dans l'odométrie visuelle.

sition 3D d'un point et sa projection dans la caméra de façon précise. Ensuite, en suivant la projection de ce point 3D dans le flux d'images nous pouvons alimenter le processus d'odométrie visuelle. La figure 2.10 montre la position des impacts lasers sur l'environnement et leurs visibilité dans la caméra.

2.5.2.2 Ajout des données Lidar dans le vecteur d'état

Afin d'être positionnées correctement dans le vecteur d'état, les données Lidar doivent être associées à des données de localisation. Cette étape nécessite la connaissance de la matrice de calibration $\mathbf{M}_{(t-c)}$ définie en annexe. Le passage des données du télémètre vers le repère monde est alors trivial :

$$\vec{P} = \mathbf{M}_{(v-m)} \cdot \mathbf{M}_{(c-v)} \cdot \mathbf{M}_{(t-c)} \cdot \vec{P}_t \quad (2.24)$$

où $\mathbf{M}_{(v-m)}$ est la matrice de passage du véhicule vers le monde, $\mathbf{M}_{(c-v)}$ est celle de la caméra vers le véhicule (voir figure 2.11).

Grâce à cette équation nous avons pu mettre en relation les données locales du télémètre avec leurs positions dans le repère monde. Les incertitudes associées à la pose du véhicule doivent aussi se répercuter sur les positions des amers Lidar dans le repère monde. Pour cela il faut utiliser la jacobienne de la matrice de passage $\mathbf{M}_{(v-m)}$ par rapport aux 6 paramètres de pose du véhicule dans le monde.

Soit $\vec{P}_v = \mathbf{M}_{(c-v)} \cdot \mathbf{M}_{(t-c)} \cdot \vec{P}_t$ les points issus du télémètre dans le repère véhicule. Le passage d'un point \vec{P}_v du repère véhicule à un point \vec{P}_m du repère monde en fonction de l'état \vec{X}_{v-m} est donc un changement de repère (voir 2.25).

$$\vec{P}_m = \mathbf{R}_{(v-m)} \cdot \vec{P}_v + \vec{T}_{(v-m)} \quad (2.25)$$

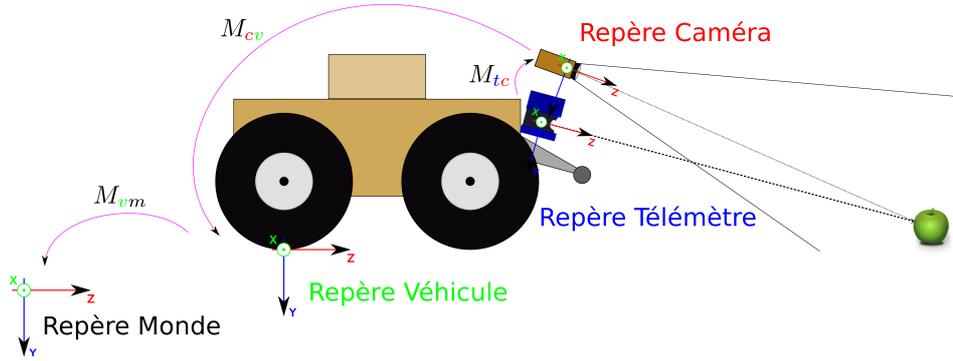


FIGURE 2.11 – Le positionnement dans le repère monde des données télémètre.

La jacobienne de la fonction permettant d'exprimer un point du repère véhicule dans le repère monde vaut donc :

$$\frac{\partial \vec{P}_m}{\partial \vec{X}} = \begin{pmatrix} 1 & 0 & 0 & \frac{\partial \mathbf{R}_{(v-m)}}{\partial \alpha_v} \cdot \vec{P}_v & \frac{\partial \mathbf{R}_{(v-m)}}{\partial \beta_v} \cdot \vec{P}_v & \frac{\partial \mathbf{R}_{(v-m)}}{\partial \gamma_v} \cdot \vec{P}_v \\ 0 & 1 & 0 & & & \\ 0 & 0 & 1 & & & \end{pmatrix} \quad (2.26)$$

Lors de l'ajout d'un point télémètre dans le vecteur d'état il suffit d'utiliser cette jacobienne pour connaître l'incertitude qui va initialiser ce point dans la matrice de covariance.

$$\mathbf{P}_{k|k_{\vec{P}_m}} = \left(\frac{\partial \vec{P}_m}{\partial \vec{X}} \right) \cdot \mathbf{P}_{k|k_{\vec{x}_{v-m}}} \cdot \left(\frac{\partial \vec{P}_m}{\partial \vec{X}} \right)^T + \mathbf{Q} \quad (2.27)$$

où \mathbf{Q} est la matrice de covariance d'un bruit blanc gaussien qui représente l'incertitude des données télémétriques.

Cependant une chose n'a pas encore été prise en compte dans cette équation. En effet, les mesures effectuées par le télémètre sont datées par rapport à celui-ci, cependant les calculs de localisation sont faits grâce à la caméra et donc datés par rapport à cette dernière. Il existe donc un temps Δ_t entre les données provenant du Lidar et les calculs de pose du véhicule. Or le véhicule bouge pendant ce laps de temps. Il existe donc un changement de repère entre la position du véhicule lors de l'acquisition télémètre et la position du véhicule lors du calcul de la pose. Cette différence doit être corrigée avant de positionner les informations du laser dans le MNT.

2.5.2.3 Asynchronisme entre les mesures Lidar et les mesures de Localisation

Les données du télémètre et de la localisation recueillies par le système sont faites à des instants différents. Cependant il est nécessaire de connaître ces in-

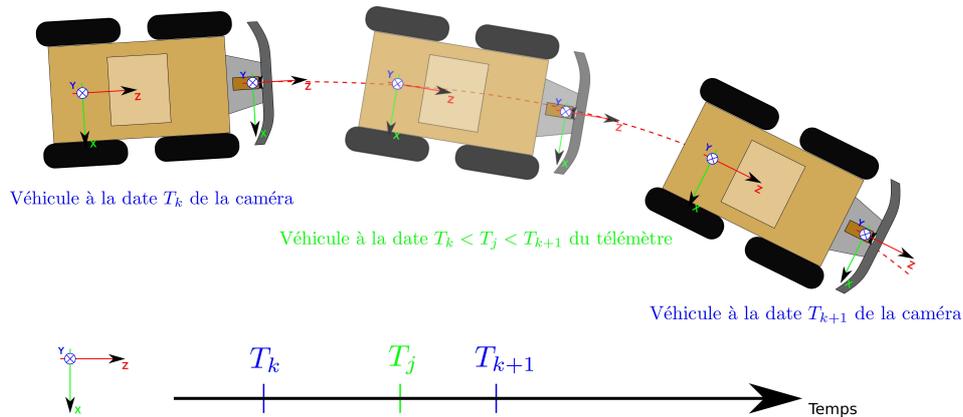


FIGURE 2.12 – Prise en compte de la désynchronisation entre données caméra et données télémètre.

formations à une même date pour pouvoir les fusionner. Les travaux de Tessier ([Tessier, 2007]) ont présenté une architecture de fusion d'informations issues de dates différentes. Dans notre cas, nous devons estimer une variable aléatoire, soit la pose du véhicule soit la mesure télémètre, à la date de validité de l'autre variable. Estimer la pose du véhicule à la date de l'acquisition du télémètre est aisée car elle peut se faire par extrapolation ou interpolation grâce à des modèles simples d'évolution. En revanche, Il est très difficile de prédire la mesure du télémètre à la date du calcul de pose puisque cette acquisition dépend de l'environnement. Nous allons donc chercher à estimer la pose du véhicule à la date de perception du Lidar. Cette estimation nous permettra d'insérer les informations du Lidar dans le vecteur d'état.

Il est plus aisé et plus précis d'interpoler une position entre deux valeurs connues plutôt que de l'extrapoler. Aussi, si cette extrapolation était faite en utilisant le modèle d'évolution par mesures proprioceptives cette estimation de position serait incomplète puisque le modèle d'évolution ne prend pas en compte l'effet de l'environnement sur la pose du véhicule. Nous allons donc interpoler les 6 paramètres de localisation à partir des dernières estimations calculées par le filtre de Kalman étendu (voir figure 2.12). Nous avons donc besoin d'avoir une estimation de l'état du véhicule à une date postérieure à celle de la mesure télémétrique. L'interpolation est fait à l'aide d'une spline cubique. Enfin, les points issus du télémètre sont insérés dans le vecteur d'état en utilisant l'équation 2.24.

2.5.2.4 Sélection des points Lidar dans le vecteur d'état

A ce stade, nous pouvons ajouter un point venant du télémètre dans notre vecteur d'état. Cependant le Lidar fournit un très grand nombre de points par mesure (par exemple le SICK LMS221 que nous utilisons sur notre véhicule expérimental

fournit 361 mesures). Aussi, un grand nombre de ces points ne sont pas visibles dans la caméra, il ne peuvent donc pas être suivis dans le flux d'images pour estimer la pose du véhicule. Nous avons adopté une stratégie simple pour sélectionner les points du télémètre qui seront ajoutés : nous gardons les points visibles dans la caméra et nous prenons un point sur N où N est choisi de façon à avoir environ 6 points sur toute la nappe télémétrique.

Nous n'avons pas choisi de sélectionner les points télémétriques suivant un critère visuel comme celui de Harris par exemple car :

- Si on filtre les points du Lidar par rapport à un score minimal sur un critère nous ne sommes pas sûrs d'avoir suffisamment de points pour effectuer notre calcul de pose.
- Si on garde les M meilleurs points par rapport à un critère (M choisi de façon empirique dans un premier temps) nous pouvons être confrontés à une mauvaise répartition des points dans l'image qui pourrait nuire à l'estimation de l'état du véhicule.

Nous avons choisi de garder cette stratégie dans un premier temps, sans toutefois nous interdire de la changer si nécessaire.

2.5.3 Mise en correspondance de points issus du télémètre

Maintenant que les points du télémètre ont été ajoutés correctement dans le vecteur d'état, il faut s'assurer qu'ils pourront être correctement mis en correspondance dans le flux de données de la caméra. En effet, les techniques de SLAM reposent sur la ré-observation des données. Nous devons donc nous assurer que nous pourrions effectivement observer ces données "télémétriques" dans les images de la caméra plusieurs fois de suite.

La mise en correspondance de points dans une image se fait par corrélation d'échantillons de texture appelé *patches*. Dans notre cas, nous utilisons le score de corrélation centrée normée (ZNCC). Cependant il est nécessaire de connaître le *patch* d'un point projeté dans la caméra pour pouvoir effectuer sa mise en correspondance. Or, il n'existe pas de *patch* valide lors de l'acquisition d'une donnée télémétrique puisque celle-ci est acquise à la date du capteur et non de la caméra.

Nous avons donc besoin de cette information télémétrique dans une image existante pour pouvoir réaliser la mise en correspondance. Pour cela, nous recherchons le vecteur d'état ayant la date la plus proche de celle des données du télémètre. Ensuite, nous projetons ces informations télémétriques dans l'espace caméra. Finalement, les *patches* sont récoltés dans cette image (la figure 2.13 illustre cette méthode).

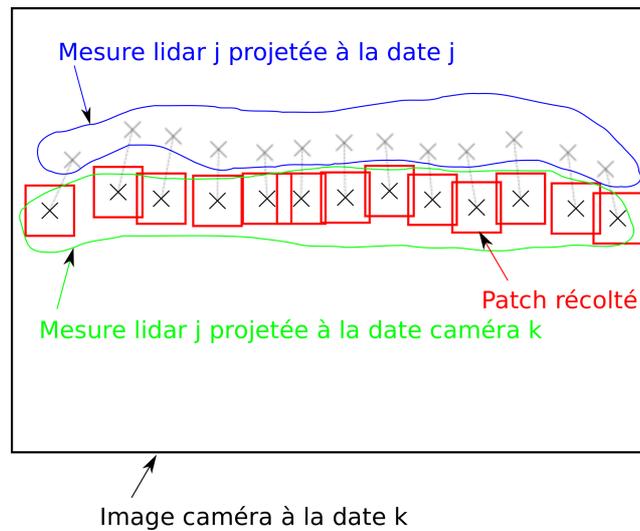


FIGURE 2.13 – La récolte des patches dans une image valide : on projette les données Lidar d'une date j dans une image caméra de date valide k puis on récupère les patches de cette image.

2.5.4 Bilan de la fusion Caméra-Télémètre

Au début de cette section nous avons vu que "localisation" et "reconstruction" étaient liées. Pour cela nous avons choisi une méthode d'estimation de pose du robot et de coordonnées de points 3D par filtrage de Kalman étendu. Afin d'estimer au mieux les composantes de notre vecteur d'état, deux méthodes d'observation d'amers dans la caméra ont été présentées. Pour nous localiser nous utilisons des amers de l'environnement, cependant nous avons besoin d'une position initiale la plus juste possible pour améliorer la précision de la méthode tant pour la densité du MNT que pour la précision de la localisation. Pour cela nous avons proposé une approche innovante d'aide à la localisation en combinant les informations d'une caméra et d'un télémètre laser. Cette fusion de données permet d'avoir une information initiale précise pour l'estimation de l'odométrie du robot et par conséquent une meilleure reconstruction.

2.6 Perception Intelligente de l'environnement dans l'espace de la caméra

2.6.1 Introduction

Il est nécessaire de pouvoir utiliser le minimum de ressource dans ce type d'application car d'autres algorithmes sont actifs en même temps (notamment celui qui

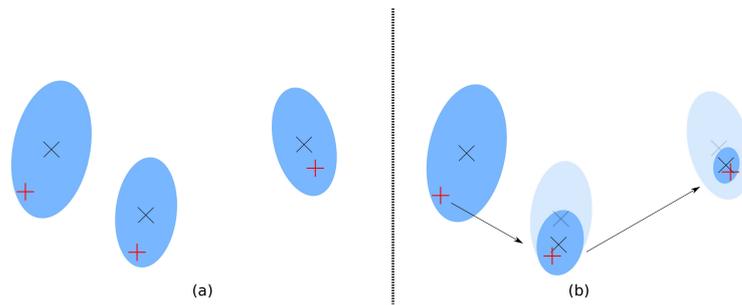


FIGURE 2.14 – La focalisation d'attention : utiliser les informations précédentes pour focaliser les nouvelles recherches. (a) système sans focalisation d'attention : les zones de recherche sont maximales. (b) système avec focalisation d'attention : les zones de recherche resserrées après mise à jour.

gère la trajectoire à suivre). Notre application se veut intelligente dans le sens où nous n'avons pas besoin de tout percevoir. Par exemple il n'est pas nécessaire de percevoir sur une grande largeur si nous n'avons pas l'intention de beaucoup nous écarter de la trajectoire. De plus, l'application doit seulement consommer les ressources nécessaires et suffisantes à la bonne réalisation de la navigation.

Dans le contexte de notre application, il est possible de mettre en place une perception intelligente en effectuant une focalisation de notre attention sur des zones pertinentes de l'environnement. Cette focalisation d'attention peut être progressive ce qui permet de minimiser les ressources nécessaires à notre tâche. Par exemple dans le cas de l'estimation de la position du véhicule, si une correspondance de point est trouvée l'espace de recherche des futurs points doit être plus petit (voir figure 2.14).

Dans cette section nous allons mettre en opposition une méthodologie classique de perception de l'environnement avec la perception focalisée que nous voulons mettre en œuvre. Deux approches vont donc être mise en concurrence :

- 1) Une méthode globale qui met à jour l'ensemble des points observés en même temps (voir algorithme 1).
- 2) Une méthode point par point qui met à jour le vecteur d'état progressivement en considérant les observations les unes après les autres (voir algorithme 2).

2.6.2 Méthode classique

2.6.2.1 Principe

A ce stade, les équations de mise à jour du filtre de Kalman étendu peuvent être mises en pratique. Dans une approche classique de filtrage de Kalman étendu, la solution consiste à utiliser toutes les observations faites à un instant k de manière

2.6. Perception Intelligente de l'environnement dans l'espace de la caméra 37

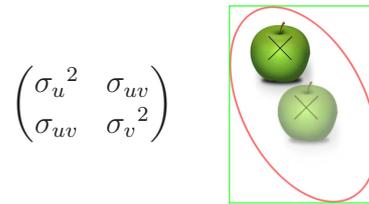


FIGURE 2.15 – La focalisation d’attention : utiliser la covariance de l’observation pour la recherche d’appariement.

globale afin d’estimer le vecteur d’état. Les équations mises en œuvre sont donc exactement celles proposées en section 2.3.4. L’algorithme 1 présente la mise en œuvre du filtre en méthode globale.

Algorithme 1 SLAM méthode Globale

- 1: Prédiction de l’évolution de l’état.
 - 2: Calculer les jacobienes de toutes les observations.
 - 3: Calculer la covariance des observations.
 - 4: Appariement des points observés dans la nouvelle image.
 - 5: Récupérer les jacobienes des points correctement mis en correspondance.
 - 6: Mise à jour de l’état et de sa covariance grâce aux innovations.
-

2.6.2.2 Recherche des appariements dans l’espace caméra

Pour pouvoir utiliser toutes les observations nous avons besoin d’apparier les points de notre vecteur d’état de l’instant $k - 1$ à l’instant k . Des méthodes d’appariements existent (par exemple Chli [Chli et Davison, 2008] détermine l’espace de recherche d’un appariement à partir d’une mixture de gaussienne) cependant nous n’allons pas les utiliser. En effet nous disposons de la covariance de l’observation calculée par l’EKF à partir d’une incertitude très petite (grâce au télémètre). Cette covariance est utilisée afin de déterminer l’espace de recherche d’un point dans l’espace d’observation de la caméra. Elle représente les ellipses d’incertitude dans lesquelles les correspondances de points devraient se trouver (voir figure 2.15).

Cette matrice de covariance peut être directement associée à une région d’intérêt où l’on va rechercher l’appariement. Pour cela il suffit de récupérer l’écart type des incertitudes sur u et v . La figure 2.16 montre ces ellipses.

2.6.2.3 Coût opératoire de la méthode

Nous allons rappeler les équations de mise à jour du filtre de Kalman ainsi que la taille des matrices en fonction du nombre de points dans le vecteur d’état N .

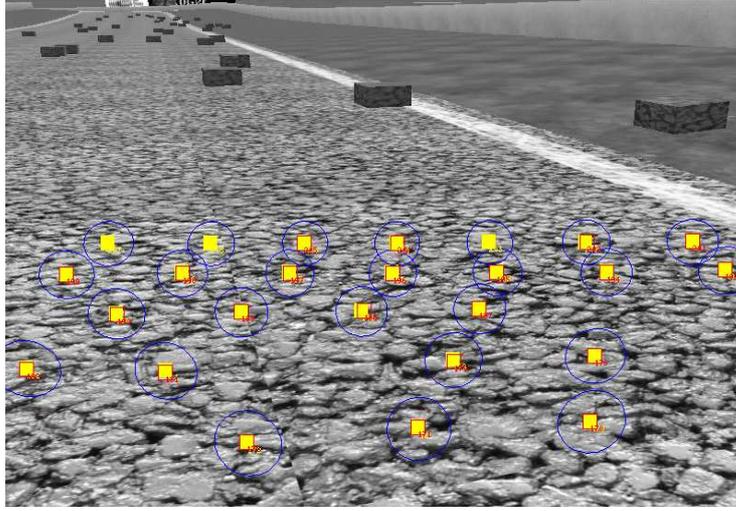


FIGURE 2.16 – La projection des ellipses d'incertitude dans l'espace caméra en méthode globale. En bleu : les ellipses d'incertitude projetées dans l'image, en jaune : les amers prédits dans l'image, en rouge : les amers correctement appariés.

$$\begin{aligned}
 y_{k(2,N)} &= h(\vec{X}_{k|k}(2,N)) - h(\vec{X}_{k|k-1}(2,N)) \\
 \mathbf{S}_{k(2,N,2,N)} &= \mathbf{H}_{k(2,N,3,N+6)} \cdot \mathbf{P}_{k|k-1(3,N+6,3,N+6)} \cdot \mathbf{H}_{k(3,N+6,2,N)}^T + \mathbf{R}_{k(2,N,2,N)} \\
 \mathbf{G}_{k(3,N+6,2,N)} &= \mathbf{P}_{k|k-1(3,N+6,3,N+6)} \cdot \mathbf{H}_{k(3,N+6,2,N)}^T \cdot \mathbf{S}_{k(2,N,2,N)}^{-1} \\
 \vec{X}_{k|k(3,N+6)} &= \vec{X}_{k|k-1(3,N+6)} + \mathbf{G}_{k(3,N+6,2,N)} \cdot y_{k(2,N)} \\
 \mathbf{P}_{k|k(3,N+6,3,N+6)} &= \mathbf{P}_{k|k-1(3,N+6,3,N+6)} - \mathbf{G}_{k(3,N+6,2,N)} \cdot \mathbf{H}_{k(2,N,3,N+6)} \cdot \mathbf{P}_{k|k-1(3,N+6,3,N+6)} \quad (2.28)
 \end{aligned}$$

En considérant les informations sur les étapes précédentes du filtre de Kalman, les étapes qui sont les plus chronophages sont manifestement la mise à jour de la covariance *a posteriori* ainsi que le calcul du gain de Kalman. Intéressons-nous au cas de la mise à jour de la covariance (voir l'équation 2.29).

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{G}_k \cdot \mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \quad (2.29)$$

Cette équation sous cette forme comprend 1 soustraction et 2 multiplications matricielles. Regardons le coût opératoire de ce calcul en fonction de N le nombre d'amers du vecteur d'état observés et correctement appariés. les couts opératoires des opérations matricielles sont estimés comme ceci :

- Pour une addition de matrices (n,m) : $2.n.m$ lectures + $n.m$ additions + $n.m$ stockages soit $4.n.m$ opérations.
- Pour une multiplication matricielle $(n,m).(m,o)$: $n.m.o$ + $m.o$ lectures + $n.(m-1).o$ additions + $n.m.o$ multiplications + $n.o$ stockages soit environ $3.n.m.o$ opération.

2.6. Perception Intelligente de l'environnement dans l'espace de la caméra 39

Pour une carte disposant de N amers, nous avons le calcul de la mise à jour de la matrice de covariance qui est de taille :

$$\mathbf{P}^{k|k}_{(3.N+6,3.N+6)} = \mathbf{P}^{k|k-1}_{(3.N+6,3.N+6)} - \mathbf{G}_{k(3.N+6,2.N)} \cdot \mathbf{H}_{k(2.N,3.N+6)} \cdot \mathbf{P}^{k|k-1}_{(3.N+6,3.N+6)}$$

soit un coût opératoire de :

$$\begin{aligned} C_{majcov} &= \mathcal{O}(4.(3.N+6)^2) + \mathcal{O}(3.(2.N)(3.N+6)^2) + \mathcal{O}(3.(2.N)(3.N+6)^2) \\ &= \mathcal{O}(36.N^2 + 144.N + 144) + \mathcal{O}(54.N^3 + 216.N^2 + 216.N) \\ &\quad + \mathcal{O}(54.N^3 + 216.N^2 + 216.N) \\ &= \mathcal{O}(108.N^3 + 468.N^2 + 576.N + 144) \end{aligned} \quad (2.30)$$

La fonction $\mathcal{O}()$ désigne le coût opératoire d'une opération.

Pour le gain de Kalman, le coût opératoire est composé du calcul de la covariance de l'innovation et d'une multiplication de son inverse par le terme $\mathbf{P}^{k|k-1}_{(3.N+6,3.N+6)} \cdot \mathbf{H}_{k(3.N+6,2.N)}^T$ (ce terme est déjà calculé lors du calcul de la covariance de l'innovation). Le coût opératoire de l'inversion matricielle est estimé à $C_{inv} = \mathcal{O}(20.(2.N)^3) = \mathcal{O}(160.N^3)$ où le facteur 20 a été déterminé par rapport au coût opératoire d'une multiplication et de manière empirique avec le logiciel Matlab. Le coût opératoire du gain de Kalman est donc de deux multiplications matricielles et d'une inversion soit :

$$\begin{aligned} C_{G_k} &= \mathcal{O}(3.(3.N+6).(2.N).(2.N)) + \mathcal{O}(3.(2.N).(3.N+6).(2.N)) + \mathcal{O}(160.N^3) \\ &= \mathcal{O}(36.N^3 + 72.N^2) + \mathcal{O}(36.N^3 + 72.N^2) + \mathcal{O}(160.N^3) \\ &= \mathcal{O}(232.N^3 + 144.N^2) \end{aligned} \quad (2.31)$$

Soit un coût total pour une mise à jour du filtre de Kalman de :

$$\begin{aligned} C_{Total} &= \mathcal{O}(108.N^3 + 468.N^2 + 576.N + 144) + \mathcal{O}(232.N^3 + 144.N^2) \\ &= \mathcal{O}(340.N^3 + 612.N^2 + 576.N + 144) \end{aligned} \quad (2.32)$$

La mise à jour du filtre de Kalman croît donc de façon cubique avec le nombre d'amers. Cependant la mise en application brutale de ces équations n'est pas en accord avec notre philosophie de perception intelligente pour plusieurs raisons :

- On ne peut pas contrôler les ressources utilisées. En effet, cette procédure nécessite d'avoir mis en correspondance l'ensemble des points 3D de la carte avant de lancer la mise à jour du filtre.
- Il faut aussi calculer l'ensemble des jacobiennes de toutes les observations même si les points ne sont pas observés.
- On ne peut pas focaliser notre attention au fur et à mesure de notre avancée dans l'estimation de la pose.

Dans ce cas là, la flexibilité de notre algorithme n'est pas assurée. Toutefois, cette méthode présente un avantage lorsqu'un mauvais appariement est présent dans le calcul de l'innovation. Dans ce cas l'effet de ce mauvais appariement est lissé par les autres correspondances.

2.6.3 Méthode point par point

2.6.3.1 Principe

Les conclusions précédentes nous ont amenés à mettre en place un filtre de Kalman étendu dont les observations et les mises à jour se font une par une. Cette approche rend possible tout ce qui ne l'était pas en méthode globale que ce soit le contrôle des ressources ou la focalisation. Elle permet aussi d'insister sur l'estimation d'un amer particulier qui n'est pas défini de façon suffisamment précise par exemple ; ou bien encore de calculer la pose du véhicule un certain nombre de fois afin de valider l'état estimé. L'algorithme 2 présente la mise en œuvre du filtre en méthode point par point.

Algorithme 2 SLAM méthode Point par Point

- 1: Prédiction de l'évolution de l'état.
 - 2: **pour** $n = 1$ à N **faire**
 - 3: Calculer la jacobienne de l'observation n .
 - 4: Calculer la covariance de l'observation n .
 - 5: Appariement du point n observé dans la nouvelle image.
 - 6: Mise à jour de l'état et de sa covariance avec l'innovation n .
 - 7: **fin pour**
-

2.6.3.2 Recherche des appariements dans l'espace caméra

Comme nous l'avons vu pour la méthode globale nous pouvons utiliser la covariance de l'observation pour rechercher les appariements des points dans l'image. L'avantage avec une méthode point par point est que les mises à jour successives de l'état vont nous permettre de restreindre l'espace de recherche pour les futurs appariements de nos amers. Le fait de diminuer cet espace de recherche donne un double avantage : il nous permet d'éviter l'appariement de faux-positifs puisque la zone de recherche est affinée à chaque itération donc plus petite autour du vrai point. Le rapport signal/bruit est donc amélioré. En conséquence, le temps passé par le processus est diminué par rapport à la méthode classique puisque la zone se réduit à chaque mise à jour du filtre. La figure 2.17 montre ces ellipses en méthode point par point.

2.6. Perception Intelligente de l'environnement dans l'espace de la caméra 41

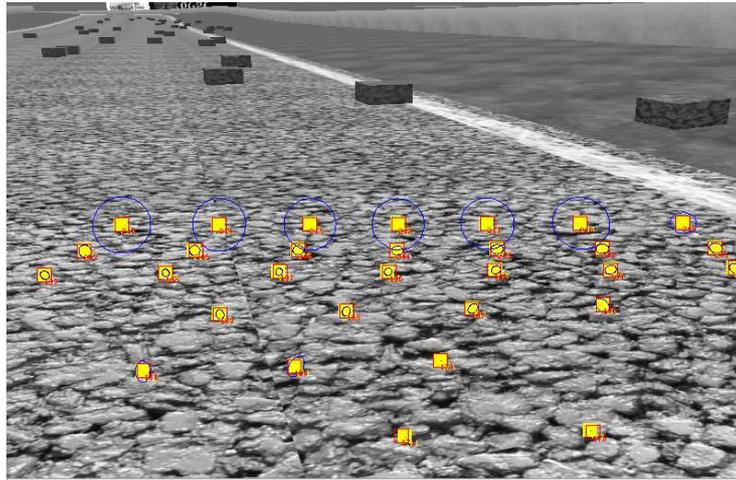


FIGURE 2.17 – La projection des ellipses d'incertitude dans l'espace caméra en méthode point par point (en bleu). Les ellipses deviennent de plus en plus petites avec chaque prise en compte d'un amer.

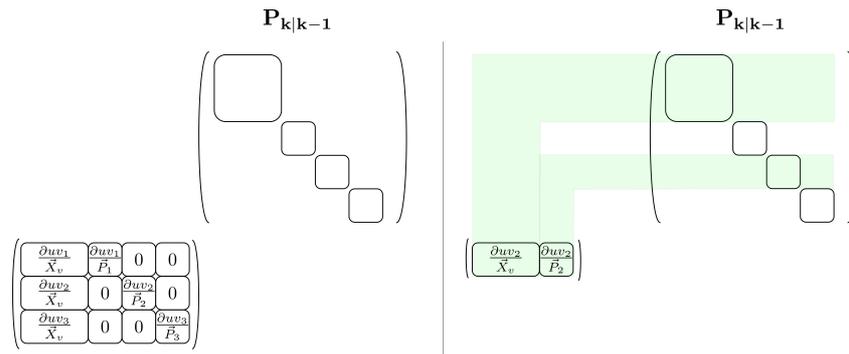


FIGURE 2.18 – Utilisation des matrices creuses en méthode point par point.

2.6.3.3 Coût opératoire de la méthode

Maintenant il nous faut regarder le coût de cette méthode pour la mise à jour du filtre puisque celle-ci va s'effectuer plusieurs fois par image. Dans ce cas, les N amers sont observés les uns après les autres donc la mise à jour se fait N fois. En utilisant le fait que les matrices jacobiniennes des observations sont creuses de façon prévisible on peut réduire cette matrice à une matrice pleine de taille $(2, 9)$ au lieu d'une matrice de taille $(2, 3.N + 6)$. Il suffit alors d'utiliser la jacobienne avec la partie de la matrice de covariance qui correspond (voir figure 2.18).

Pour une carte disposant de N amers le calcul d'une mise à jour de la matrice de covariance devient :

$$\mathbf{P}_{k|k(3.N+6,3.N+6)} = \mathbf{P}_{k|k-1(3.N+6,3.N+6)} - G_{k(3.N+6,2)} \cdot H_{k(2,9)} \cdot \mathbf{P}_{k|k-1(9,3.N+6)}$$

soit un coût opératoire pour N mise à jour :

$$\begin{aligned}
 C_{majcov} &= N. (\mathcal{O}(4.(3.N + 6)^2) + \mathcal{O}(3.2.9.(3.N + 6)) + \mathcal{O}(3.2.(3.N + 6)^2)) \\
 &= N. (\mathcal{O}(36.N^2 + 144.N + 144) + \mathcal{O}(162.N + 324) + \mathcal{O}(54.N^2 + 216.N + 216)) \\
 &= \mathcal{O}(80.N^3 + 522.N^2 + 684.N) \tag{2.33}
 \end{aligned}$$

Ce qui est environ 25% moins couteux en temps de calcul que la mise à jour de la covariance en méthode globale.

Pour le calcul du gain de Kalman, la démarche est la même que pour la méthode globale. Le coût opératoire des inversions de la covariance de l'innovation peut être estimé à $160.N$ pour la version point par point (N inversions de matrices $(2, 2)$). Le cout opératoire du calcul du gain de Kalman est donc de $2.N$ multiplications de matrices et de N inversions soit :

$$\begin{aligned}
 C_{G_k} &= N. (\mathcal{O}(3.2.9.2) + \mathcal{O}(3.(3.N + 6).2.2)) + \mathcal{O}(160.N) \\
 &= N. (\mathcal{O}(108) + \mathcal{O}(36.N + 72)) + \mathcal{O}(160.N) \\
 &= \mathcal{O}(36.N^2 + 340.N) \tag{2.34}
 \end{aligned}$$

Le total pour les N mises à jour du filtre est de :

$$\begin{aligned}
 C_{G_k} &= \mathcal{O}(80.N^3 + 522.N^2 + 684.N) + \mathcal{O}(36.N^2 + 340.N) \\
 &= \mathcal{O}(80.N^3 + 558.N^2 + 1024.N) \tag{2.35}
 \end{aligned}$$

Au final, les calculs sont théoriquement plus rapides que ceux proposée en méthode globale (la figure 2.19 montre les courbes de croissance du coût opératoire en fonction du nombre de points dans le vecteur d'état). Cette méthode permet également de maîtriser la mise à jour du filtre afin de lancer les calculs seulement si le point est visible et si l'application en a besoin. Si par erreur un faux-positif est utilisé par le processus de mise à jour cela a pour conséquence de faire échouer les mises en correspondances suivantes. Si trop d'échecs arrivent il suffit de reprendre une sauvegarde de l'état du véhicule à un moment précédent le mauvais appariement puis de continuer le processus en écartant les points ambigus de la mise à jour. Ce phénomène peut arriver notamment pour le premier point apparié puisque l'incertitude associée de l'observation dans la caméra est grande et donc le rapport signal sur bruit faible.

2.6.4 Bilan de la perception focalisée

Nous venons de présenter deux méthodes. La première méthode, classique, consiste à utiliser toutes les observations en même temps dans la mise à jour de l'EKF. En plus de proposer un temps de calcul moins important que la méthode

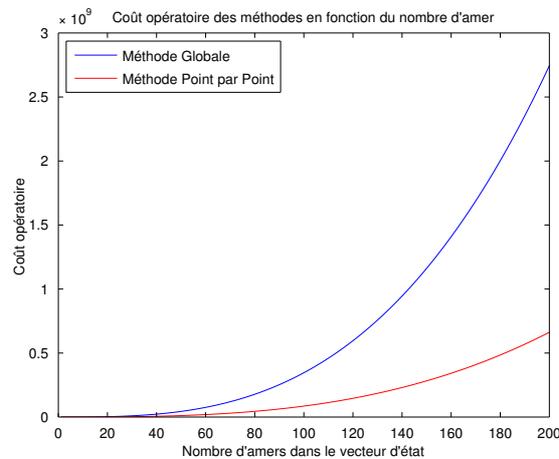


FIGURE 2.19 – La courbe d'évolution du coût opératoire de chaque méthode. En bleu : la méthode globale, en rouge : la méthode point par point.

globale, la méthode en point par point permet aussi de contrôler le temps passé dans la boucle de mise à jour de l'état. En effet, dans le cas de la méthode globale, aucune mise à jour n'est effectuée tant que les appariements ne sont pas tous finis. Si pour une raison quelconque le temps de calcul effectif dépasse le temps de calcul disponible il n'est pas possible d'arrêter la mise à jour. le système peut alors rater des images et/ou s'emballer.

Le processus de mise à jour est plus souple pour la méthode point par point car elle dispose d'une boucle au sein de la mise à jour. Cette boucle nous permet de contrôler le temps passé à faire la mise à jour du vecteur d'état et de s'assurer que ce temps ne dépasse pas un certain seuil. Ce seuil est réglé par le concepteur en fonction de la fréquence de la caméra, des autres processus en place dans le système complet etc ... Ce principe nous permet de focaliser nos ressources sur l'information pertinente à obtenir ; si un amer de l'environnement doit être précisé on peut se concentrer sur lui. Enfin, si l'estimée de la pose du robot est suffisamment précise nous pouvons stopper les calculs.

2.7 Correction de la dérive de la localisation absolue

L'ajout des données provenant du télémètre laser ainsi que la mise en place du processus de focalisation nous permettent de calculer efficacement l'odométrie visuelle associée au déplacement du véhicule comme montré en figure 2.20. La figure 2.21 montre les erreurs de localisation basées sur cette odométrie lors du suivi d'une trajectoire synthétisée sous Matlab correspondant à la figure 2.20. Au bout de 100 mètres, la localisation du véhicule présente une dérive réelle d'environ

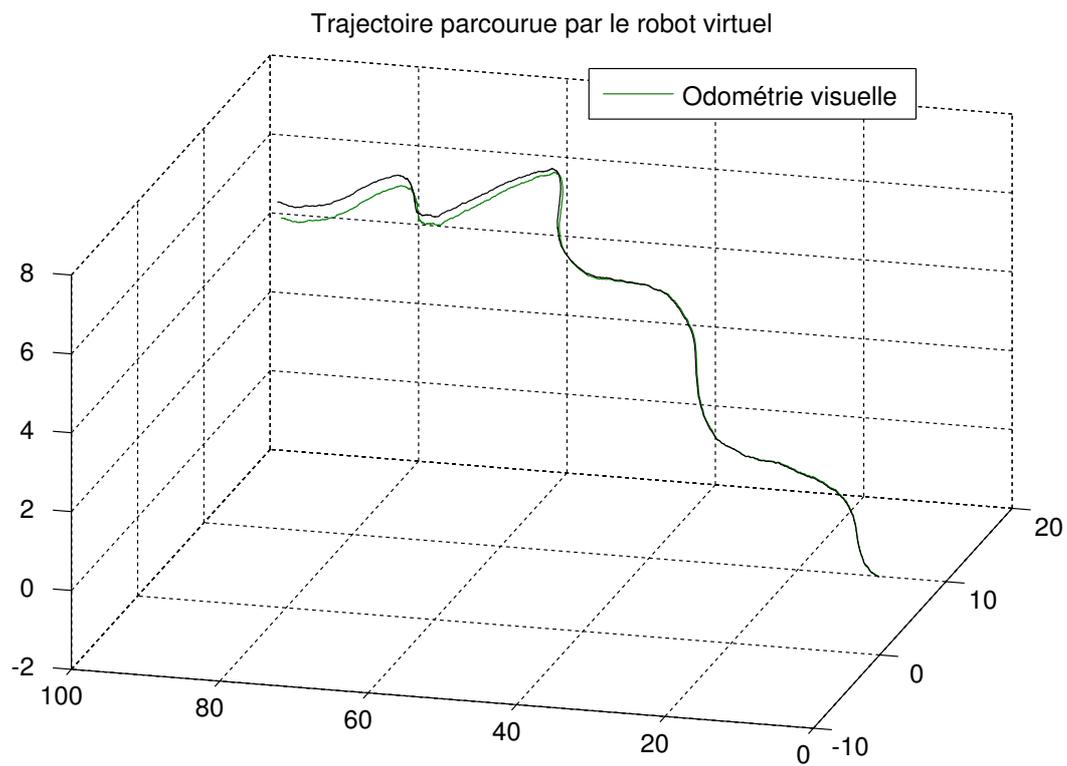


FIGURE 2.20 – Vue de dessus de la trajectoire effectuée par le robot virtuel (en noir) ainsi que la trajectoire estimée par l'odométrie visuelle (en bleu).

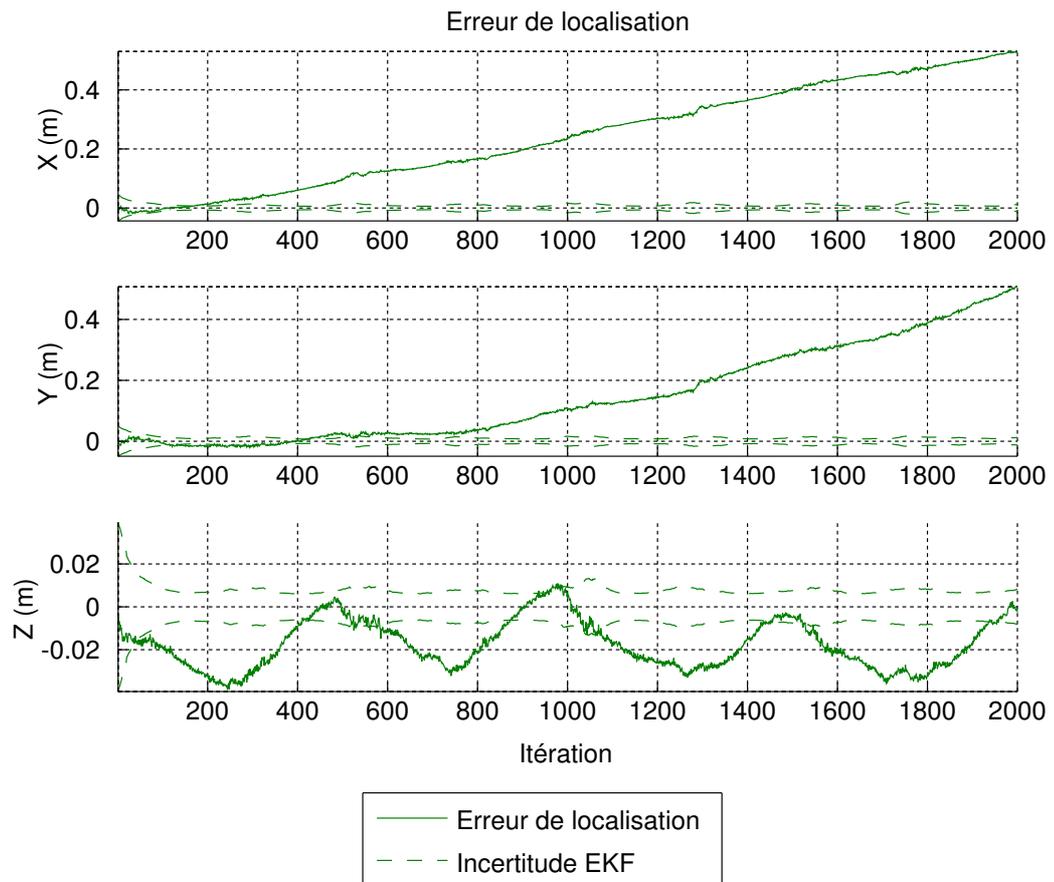


FIGURE 2.21 – Erreur de localisation en X, Y et Z de l'EKF par rapport à une trajectoire de référence et incertitudes associées.

25cm (ces résultats seront à nouveaux montrés dans le chapitre 4).

Cependant, l'estimation de l'erreur maximale calculée par le processus d'odométrie visuelle est de l'ordre de 2cm. Ce résultat montre clairement la non-intégrité du processus d'odométrie visuelle puisque l'erreur maximale estimée est inférieure à l'erreur réelle. La fonction première de notre système est de reconstruire l'environnement à l'avant du véhicule. Avoir une localisation intègre de notre véhicule est secondaire mais reste tout de même important pour un système de navigation.

En effet, ajouter des informations, même très robustes comme celles d'un télémètre laser ne changent pas le fait que l'odométrie visuelle par filtrage de Kalman étendu reste non-intègre. Cela provient du fait que :

- D'une part le système d'état est non-observable avec l'utilisation d'informations relatives à sa position puisque l'état est connu à une transformation rigide près (celle qui va du repère initial au repère absolu). Par exemple le cap initial est inconnu et aucune donnée ne nous permet de l'observer. Cette non-observabilité constitue un biais de localisation et sera notée \vec{B} .
- De plus Julier et Bailey ([Julier et Uhlmann, 2001, Bailey et al., 2006a]) ont montré que le filtre de Kalman étendu ne peut être intègre que ce soit en état stationnaire ou sur de longues distances. Les auteurs de [Huang et al., 2008] ont proposé une méthode de *First-Estimate Jacobian* (FEJ) où ils utilisent la première estimation faite d'un amer durant tout le processus SLAM pour calculer la jacobienne de l'observation. Ceci a pour but d'améliorer la consistance du filtre sans réellement garantir une intégrité. Cette inconsistance est due à la présence de bruit sur l'observation et des non-linéarités du système d'état. La dérive alors présente sur le système varie en fonction du temps et sera notée \vec{D}_t .

Ces deux grandeurs \vec{B} et \vec{D}_t traduisent la non-intégrité de notre système de localisation. Or, un véhicule autonome doit disposer d'une information intègre s'il veut pouvoir fonctionner en toute sécurité. En effet, même si le processus d'odométrie visuelle est très précis localement (d'une itération à l'autre) il doit l'être aussi globalement pour les besoins de l'application. Le système de guidage doit pouvoir disposer à chaque instant d'une localisation intègre, c'est à dire dont l'ellipse d'incertitude englobe la vraie position afin de pouvoir comparer cette position à son couloir de navigation. Illustrons cette contrainte : imaginons que le véhicule doit évoluer dans un terrain miné en suivant une trajectoire qui lui a été fournie préalablement. Cette trajectoire est incluse dans un couloir de navigation dans lequel le véhicule peut évoluer sans danger. Ce couloir de navigation est très restrictif puisque si nous ne sommes pas à l'intérieur de celui-ci nous pouvons à tout moment nous heurter à une mine. L'information d'intégrité est donc très importante. **Les techniques d'odométries visuelles n'étant pas consistantes nous devons rajouter une information qui garantisse l'intégrité de la localisation.**

Pour cela nous proposons de modéliser le système afin de quantifier comment

évolue cette dérive de localisation en fonction du déplacement. Une fois l'évolution du modèle estimée, nous proposons de corriger le biais grâce à une observation de la position dans le repère monde (par exemple en utilisant un GPS-RTK). Cette observation a l'avantage de nous fournir, en plus, la localisation absolue de notre système. Des travaux concernant la fusion de GPS et de SLAM existent ([Carlson, 2010, Schleicher *et al.*, 2009]) mais ne seront pas utilisés ici car ils ne proposent pas d'informations suffisamment consistantes pour englober toutes les possibilités de déplacement. La disponibilité du GPS n'étant pas assurée (perte de signal, manque de satellite), notre système servira d'odométrie visuelle lorsque les données GPS ne sont pas présentes.

2.7.1 État du système

Pour l'estimation de la localisation du véhicule, deux grandeurs doivent être considérées :

1. l'erreur de localisation du véhicule provenant de l'estimation fournie par l'odométrie visuelle. Cette localisation est assujettie à des problèmes de dérive dus notamment à l'inconsistance du filtre de Kalman étendu (\vec{D}_t).
2. l'erreur (\vec{B}) commise entre la localisation dans le repère monde (localisation absolue) et la position estimée de l'odométrie visuelle. Cette erreur est présente à cause de la mauvaise initialisation de la première position du robot (fixée à zéro) par rapport à un repère absolu.

Notre système vérifie l'équation 2.36 où \vec{X}_r est la vraie position du véhicule dans le repère monde, \vec{X}_{v-m} est l'estimée de cette position faite par le filtre de Kalman étendu et $\vec{B} + \vec{D}_t$ le biais total du système. Lorsque une estimation de ces biais sera faite, elle sera utilisée pour corriger l'état du système.

$$\vec{X}_r = \vec{X}_{v-m} + \vec{B} + \vec{D}_t$$

$$\begin{pmatrix} X_v \\ Y_v \\ Z_v \\ \alpha_v \\ \beta_v \\ \gamma_v \end{pmatrix} = \begin{pmatrix} \hat{X}_v \\ \hat{Y}_v \\ \hat{Z}_v \\ \hat{\alpha}_v \\ \hat{\beta}_v \\ \hat{\gamma}_v \end{pmatrix} + \begin{pmatrix} B_{X_v} \\ B_{Y_v} \\ B_{Z_v} \\ B_{\alpha_v} \\ B_{\beta_v} \\ B_{\gamma_v} \end{pmatrix} + \begin{pmatrix} D_{X_{v_t}} \\ D_{Y_{v_t}} \\ D_{Z_{v_t}} \\ D_{\alpha_{v_t}} \\ D_{\beta_{v_t}} \\ D_{\gamma_{v_t}} \end{pmatrix} \quad (2.36)$$

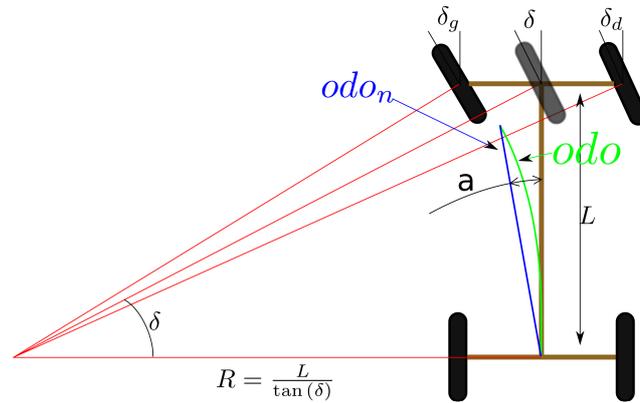


FIGURE 2.22 – Le modèle d'évolution du véhicule.

2.7.2 Estimation du biais $\vec{B} + \vec{D}_t$

2.7.2.1 Évolution du biais

Comme nous l'avons vu l'état \vec{X}_v du véhicule estimé par l'odométrie visuelle est inconsistant. Afin de fournir au processus de guidage une incertitude de localisation cohérente nous devons faire évoluer le biais de façon plausible. Pour cela le biais doit évoluer avec le temps et plus précisément en fonction du déplacement du véhicule.

Dans un filtre de Kalman étendu classique, l'évolution se déroule à travers une matrice d'évolution de la forme :

$$\vec{X}_{(k+1|k)} = f(\vec{X}_{(k|k)}, \vec{U}_k, \vec{Q}_k) \quad (2.37)$$

Le vecteur \vec{U}_k comporte dans notre cas l'angle de braquage du véhicule δ et la distance parcourue du véhicule en abscisse curviligne odo . La figure 2.22 montre les différentes grandeurs utilisées dans ce modèle d'évolution.

Les équations d'évolution utilisées par l'odométrie visuelle sont les suivantes :

$$\begin{aligned} \delta \neq 0 : \\ R &= \frac{L}{\tan(\delta)} \\ a &= \frac{odo}{R} \\ odo_n &= 2R \sin\left(\frac{a}{2}\right) \\ \delta = 0 : \\ a &= 0 \\ odo_n &= odo \end{aligned}$$

$$\begin{aligned}
\vec{X}_{v_{k+1}} &= f(\vec{X}_{v_k}, \text{odo}, \delta) \\
X_{v_{k+1}} &= X_{v_k} + \text{odo}_n \cdot \sin\left(\beta_v + \frac{a}{2}\right) \cdot \cos(\alpha_v) \\
Y_{v_{k+1}} &= Y_{v_k} - \text{odo}_n \cdot \sin(\alpha_v) \\
Z_{v_{k+1}} &= Z_{v_k} + \text{odo}_n \cdot \cos\left(\beta_v + \frac{a}{2}\right) \cdot \cos(\alpha_v) \\
\beta_{v_{k+1}} &= \beta_{v_k} + a
\end{aligned} \tag{2.38}$$

Avec R le rayon de braquage du véhicule, a la variation de cap du véhicule et odo_n la norme de la distance entre le point de départ du véhicule et le point d'arrivée (voir figure 2.22). En considérant que le véhicule ne subit pas de glissement, que son tangage est nul, qu'il évolue sur un sol totalement plat et que les commandes appliquées sont parfaitement connues la covariance prédite vaut alors :

$$P_{(k+1|k)} = F \cdot P_{(k|k)} \cdot F^T \tag{2.39}$$

où F est la jacobienne de la fonction d'évolution f par rapport à l'état \vec{X} .

Dans notre cas, l'erreur de localisation commise est due à la fois au problème d'initialisation du vecteur d'état dans le repère monde et à la fois à la dérive du processus d'estimation de la pose du robot.

Avant de poursuivre dans l'estimation du biais, regardons ce que donne l'utilisation d'une matrice d'évolution suivie de sa jacobienne dans un cas concret simple. Dans cet exemple nous ne considérerons qu'une erreur initiale sur le cap du véhicule. En effet, l'erreur initiale de position peut se limiter à un bruit blanc sur les trois coordonnées de localisation. Par contre, l'incertitude initiale sur le cap va avoir une influence à la fois sur l'estimation du cap du véhicule et sur l'estimation de sa position.

Considérons le véhicule lors de l'initialisation du processus. Sa position est donnée par un vecteur $(X_{v_{init}} \ Y_{v_{init}} \ Z_{v_{init}})^T$ connu. Prenons le cas où l'incertitude en cap est de π (cap inconnu) et le cap du véhicule initialisé arbitrairement sur l'axe des Z. L'angle de braquage du véhicule vaut exactement 0. De plus, considérons que le véhicule a avancé d'une distance D entre deux itérations du système. Le vecteur d'état du système à l'itération $k+1$ devient alors $(0 \ 0 \ D \ 0 \ 0 \ 0)^T$. À l'état k (état initial), la matrice de covariance vaut :

$$P_{(k|k)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

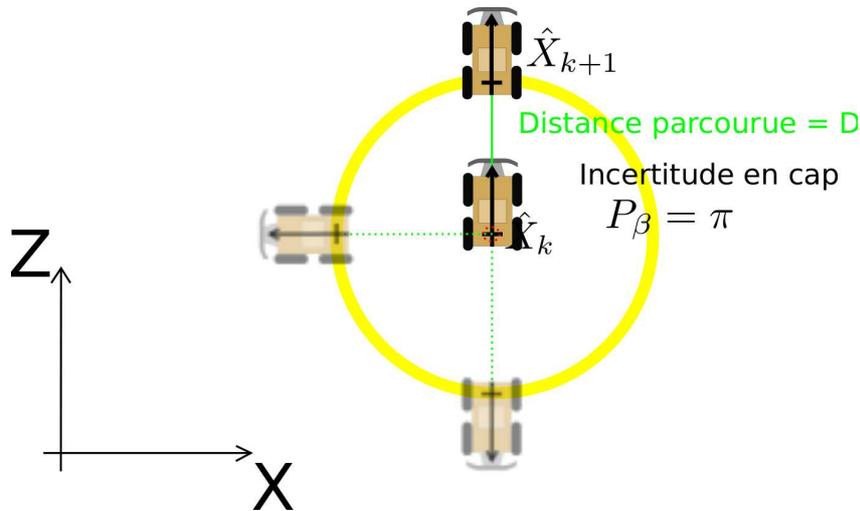


FIGURE 2.23 – Evolution de l'incertitude de position pour une incertitude en cap de π

La figure 2.23 montre cette configuration. On y voit dessiné en cercle jaune l'ensemble des positions possibles du véhicule lors d'une incertitude en cap de π . Ce cas de figure arrive à l'initialisation du système puisque nous ne pouvons pas connaître l'orientation initiale dans un repère global.

Pour cette configuration, la jacobienne F de la fonction d'évolution vaut :

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & D & 0 \\ 0 & 1 & 0 & -D & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Pour l'état prédit la covariance vaut donc :

$$P(k+1|k) = F \cdot P_{(k|k)} \cdot F^T = \begin{pmatrix} D^2\pi^2 & 0 & 0 & 0 & D\pi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ D\pi^2 & 0 & 0 & 0 & \pi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Cette covariance est représentée en figure 2.24 (ellipse bleue). On voit alors que l'on est loin d'englober toutes les possibilités de localisation du robot. Or, nous rappelons que pour être intègre, l'incertitude de notre biais prédit doit, à minima,

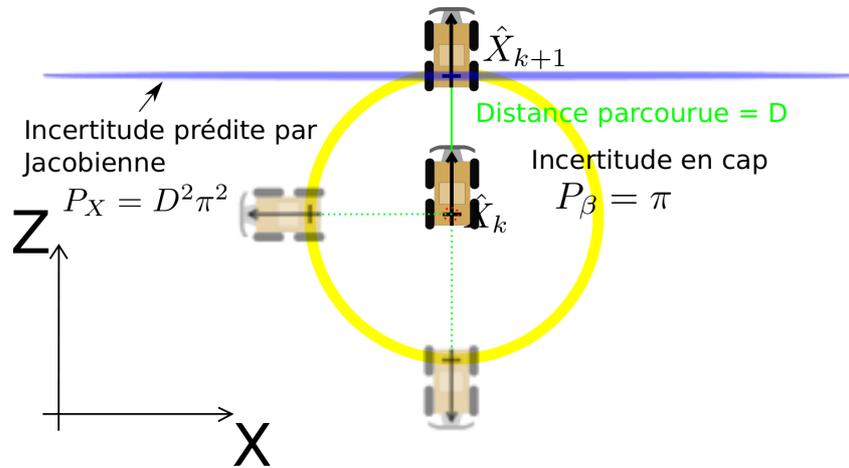


FIGURE 2.24 – Ellipse de l’incertitude en position en utilisant une jacobienne de l’évolution

englober tout le cercle des possibilités de positionnement du véhicule. Si cette intégrité de localisation n’est pas assurée nous ne pouvons pas garantir l’intégrité physique du véhicule.

Une solution serait de faire évoluer le système avec seulement un bruit blanc additif sur les composantes en x et z de variance D^2 . Cependant cette solution implique que la position prédite du système n’évolue pas avec le déplacement du véhicule. Or, les besoins du système de navigation font qu’un déplacement, aussi faux soit-il, doit être visible par le système de guidage. Il nous faut donc faire avancer notre robot avec la prédiction de son déplacement même en présence d’une incertitude de π sur son orientation. Dans ce cas, il faut que l’incertitude qui se situe bien à l’endroit estimé de notre position (en $\vec{X}_{(k+1)}$) englobe toutes les possibilités de notre déplacement selon un cap dont la valeur est connu à π près.

De ce fait, plusieurs ellipses d’incertitudes peuvent englober toutes les possibilités de position du véhicule en partant de la position prédite. Différentes possibilités sont montrées en figure 2.25. La première possibilité est un cercle de rayon $2D$. La deuxième est une ellipse de grand rayon $2D$ et de petit rayon $\sqrt{2}D$. Cependant il n’est pas souhaitable d’obtenir une ellipse trop grande puisque cela pourrait contraindre le système de navigation à s’arrêter prématurément. L’ellipse la plus petite est donc suffisante (Ellipse rouge de la figure 2.25). La matrice de covariance

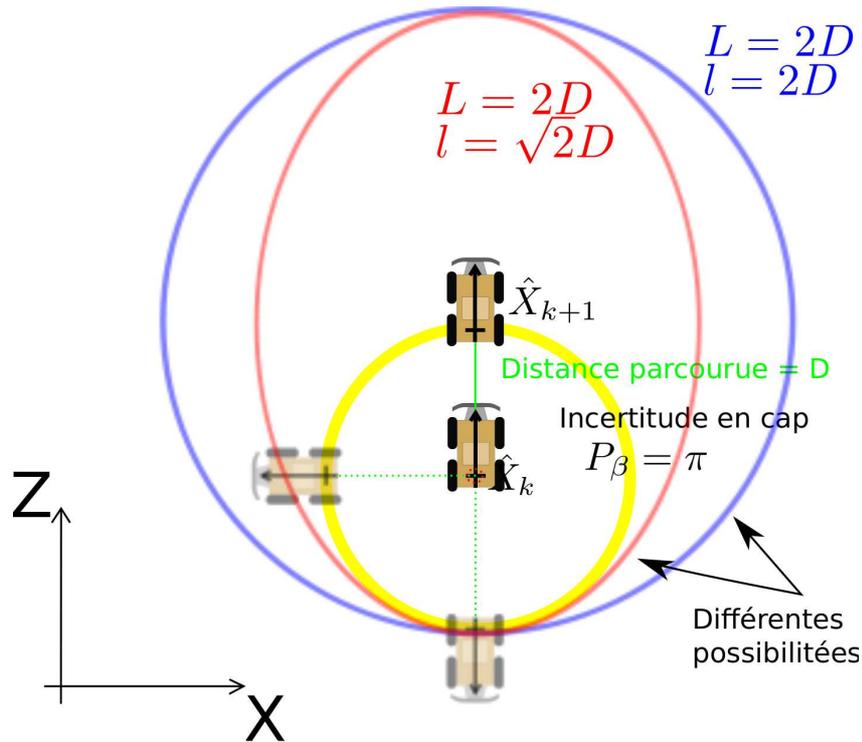


FIGURE 2.25 – Ellipses d'incertitudes possibles pour englober la position vraie.

associée à une ellipse de cette forme est représentée en équation 2.40.

$$P_{(k+1|k)} = \begin{pmatrix} (2D)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (\sqrt{2}D)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.40)$$

Cette matrice de covariance permet que notre incertitude artificielle soit constante quel que soit le biais en cap présent sur notre système. Nous devons donc constituer un système qui inclura cette caractéristique.

2.7.2.2 Évolution par addition

Dan l'exemple précédent, nous avons cherché à utiliser la fonction d'évolution du modèle cinématique du véhicule pour estimer l'incertitude en position lorsqu'un biais en cap de π est présent. Nous avons montré que nous ne pouvons pas passer par une jacobienne de l'évolution pour faire croître la covariance *a priori* de façon à avoir un positionnement intègre, c'est à dire qui englobe toutes les possibilités de déplacement du véhicule.

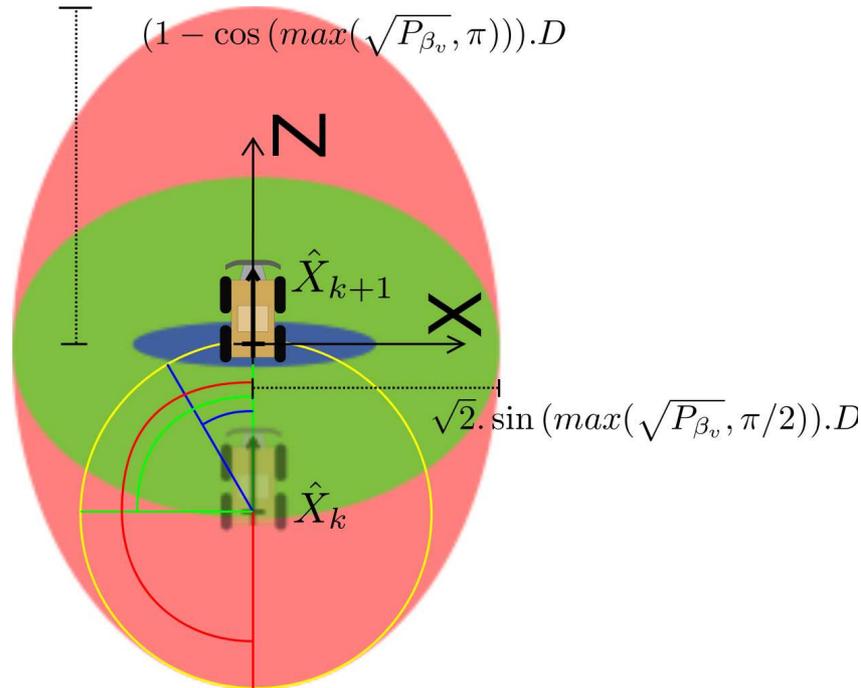


FIGURE 2.26 – Définition des incertitudes en fonction du biais en cap. Les ellipses en bleu, verte et rouge correspondent respectivement à des incertitudes en cap de $\pi/6, \pi/2$ et π .

Nous avons donc proposé d'utiliser une incertitude qui soit consistante. Maintenant il nous faut chercher le moyen d'obtenir cette incertitude de position en utilisant les incertitudes d'orientation du véhicule (cap, roulis, tangage). Le problème est le suivant : il faut pouvoir faire croître les covariances en position en fonction des incertitudes d'orientation (P_{α_v} P_{β_v} P_{γ_v}) et du dernier déplacement effectué du véhicule (Δ_X Δ_Y Δ_Z). Par exemple dans le cas de l'incertitude en cap, les incertitudes en X et Z doivent augmenter en fonction du dernier déplacement effectué. Plus précisément, la figure 2.26 montre comment les incertitudes en X et en Z sont créées à partir de l'incertitude en cap.

Dans cette figure, nous avons créé une ellipse dont les rayons sont contrôlés par l'incertitude en cap. Le rayon de l'ellipse sur l'axe X évolue avec le sinus de l'incertitude en cap du véhicule. Cette incertitude est seuillée à $\pi/2$, valeur à partir de laquelle le rayon de l'ellipse reste constant. De plus, pour englober toutes les possibilités d'évolution lorsque une incertitude de $\pi/2$ est présente sur le cap, un facteur $\sqrt{2}$ doit être rajouté. Au final nous obtenons comme équation $\delta_{P_{X_v}} = \sqrt{2} \cdot \sin(\max(\sqrt{P_{\beta_v}}, \pi/2)) \cdot \Delta_Z$. Le rayon sur l'axe Z évolue avec le cosinus de l'incertitude en cap. Plus précisément il évolue suivant $1 - \cos(\sqrt{P_{\beta_v}})$. Ici, l'incertitude est seuillée à π puisque l'espace de déplacement possible ne gran-

dit plus à partir de cette valeur. L'équation qui régit l'incertitude en X vaut donc $\delta_{P_{Z_v}} = (1 - \cos(\max(\sqrt{P_{\beta_v}}, \pi))).\Delta_Z$.

Pour résumer, si on prend le cas de l'incertitude en X , elle dépend seulement de l'incertitude en cap P_{β_v} et de l'incertitude en roulis P_{γ_v} . L'incertitude en Z dépend quant à elle de l'incertitude en cap P_{β_v} et de l'incertitude en tangage P_{α_v} . Si on effectue ce raisonnement sur toutes les incertitudes, on arrive à un δ incertitude à ajouter à la diagonale de la matrice de covariance en position dont l'écart type est de cette forme :

$$\begin{pmatrix} \delta_{P_{X_v}} \\ \delta_{P_{Y_v}} \\ \delta_{P_{Z_v}} \end{pmatrix} = (\sqrt{2} \sin \left(\max \left(\sqrt{\begin{pmatrix} 0 & P_{\gamma_v} & P_{\beta_v} \\ P_{\gamma_v} & 0 & P_{\alpha_v} \\ P_{\beta_v} & P_{\alpha_v} & 0 \end{pmatrix}}, \pi/2 \right) \right) + \left[1 - \cos \left(\max \left(\sqrt{\begin{pmatrix} P_{\beta_v} & 0 & 0 \\ 0 & P_{\gamma_v} & 0 \\ 0 & 0 & P_{\alpha_v} \end{pmatrix}}, \pi \right) \right) \right] + \left[1 - \cos \left(\max \left(\sqrt{\begin{pmatrix} P_{\gamma_v} & 0 & 0 \\ 0 & P_{\alpha_v} & 0 \\ 0 & 0 & P_{\beta_v} \end{pmatrix}}, \pi \right) \right) \right] \cdot \begin{pmatrix} \Delta_X \\ \Delta_Y \\ \Delta_Z \end{pmatrix} \quad (2.41)$$

Ainsi, nous avons mis en place une évolution de l'incertitude qui est consistante en fonction du biais présent sur les orientations du véhicule. Pour illustrer ce calcul, reprenons le cas où l'incertitude vaut π sur le lacet ($P_{\beta_v} = \pi^2$) et où on avance de 1 mètre sur Z ($(\Delta_X \ \Delta_Y \ \Delta_Z)^T = (0 \ 0 \ 1)^T$). Les valeurs de la diagonale de la matrice d'incertitude en position valent :

$$\delta = \begin{pmatrix} \delta_{P_X} \\ \delta_{P_Y} \\ \delta_{P_Z} \end{pmatrix} = \begin{pmatrix} \sqrt{2} \sin(\pi/2) \\ 0 \\ 1 - \cos(\pi) \end{pmatrix} = \begin{pmatrix} \sqrt{2} \\ 0 \\ 2 \end{pmatrix}$$

Nous voyons donc que ce calcul nous permet d'obtenir la matrice exposée dans le paragraphe 2.7.2.1 qui nous avait permis d'englober toutes les possibilités de position du véhicule à partir de de la position estimée.

Tous les calculs précédents ont été fait avec comme hypothèse une incertitude initiale de position nulle. Dans le cas contraire, l'incertitude à appliquer pour garder l'intégrité de l'estimation du vecteur d'état est dépendante de la covariance *a priori* déjà présente. Une étape supplémentaire doit être mise en place pour prendre en compte cette dépendance. Pour cela, il faut faire en sorte qu'une incertitude d'écart type δ fasse bien grandir l'ellipse d'incertitude de δ .

L'étape à mettre en place est de "corrélér" cette incertitude avec l'incertitude déjà présente, c'est à dire d'avoir :

$$\delta + \sqrt{P_{(k|k)}} = \sqrt{P_{(k+1|k)} + \varepsilon^2} \quad (2.42)$$

où δ est l'incertitude calculée par l'équation 2.41 et ε est l'incertitude réellement ajoutée à la covariance $P_{(k|k)}$. D'où :

$$\varepsilon^2 = \delta^2 + 2 \sqrt{P_{(k|k)}} \delta \quad (2.43)$$

A ce stade, la covariance en position évolue en fonction des incertitudes présentes sur les orientations. De plus, grâce à cette dernière étape, l'incertitude évolue de façon consistante. Finalement, pour faire évoluer les biais en orientations, un bruit blanc est ajouté sur ces composantes. Les écarts types de ce bruit blanc sont déterminés de façon empirique.

2.7.2.3 Mise à jour du biais

Maintenant que l'incertitude de l'état évolue en fonction du temps (et entre autre en fonction du déplacement du véhicule), il nous faut estimer le biais $\vec{B} + \vec{D}_t$. Ce biais ne peut être estimé qu'avec une observation consistante. Celle-ci se fait grâce à l'observation de la longitude, de la latitude et du cap fournie par un GPS-RTK. Le vecteur d'observation \vec{Y}_{gps} est donc :

$$\vec{Y}_{gps} = \begin{pmatrix} X_v \\ Z_v \\ C_v \end{pmatrix}$$

Cette observation correspond à la somme de l'état estimé par le processus SLAM \vec{X}_v et du biais de celui-ci $\vec{B} + \vec{D}_t$:

$$\vec{Y}_{gps} = \vec{X}_v + \vec{B} + \vec{D}_t = \begin{pmatrix} \hat{X}_v \\ \hat{Z}_v \\ \hat{\beta}_v \end{pmatrix} + \begin{pmatrix} B_{X_v} \\ B_{Z_v} \\ B_{\beta_v} \end{pmatrix} + \begin{pmatrix} D_{tX_v} \\ D_{tZ_v} \\ D_{t\beta_v} \end{pmatrix}$$

Cette observation est utilisée pour l'innovation dans le filtre de Kalman. La prédiction de l'observation est, dans ce cas, l'état estimé par l'odométrie visuelle plus le biais. La mesure de l'observation se fait grâce aux données du GPS-RTK. Maintenant il ne reste qu'à appliquer les équations du filtre de Kalman à savoir le calcul du gain optimal ainsi que la mise à jour de l'état et de la covariance.

Synchronisation de l'observation GPS avec l'état estimé

Nous venons de voir que l'innovation utilisée par le filtre de Kalman est basée sur des observations prédites et mesurées qui proviennent de deux systèmes différents. Cependant pour que cette mesure de l'innovation soit cohérente les deux

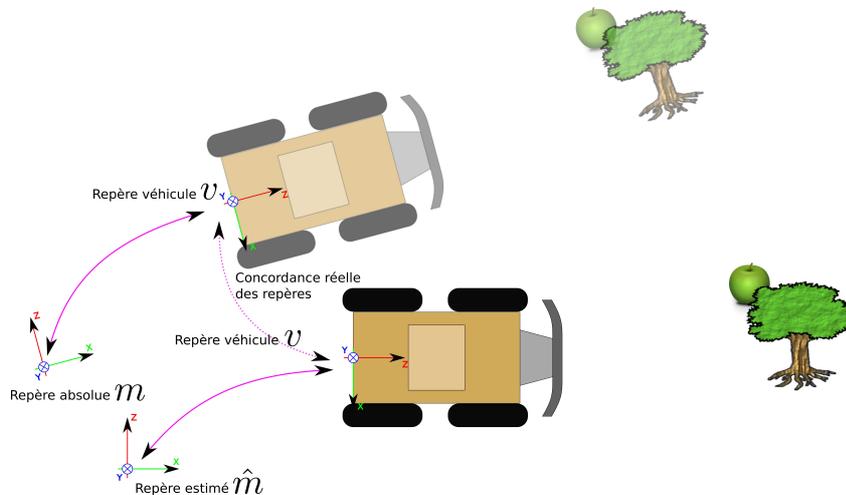


FIGURE 2.27 – Correction de l'odométrie visuelle : transformation rigide du repère estimé au repère absolu

données doivent provenir du même instant. Or les dates d'acquisitions des données GPS et les dates d'estimation de la pose par l'odométrie visuelle ne sont pas synchrones. Il nous faut donc synchroniser une des données par rapport à l'autre. Pour cela nous utilisons l'architecture d'ordonnement de données AROCCAM¹ ([Tessier *et al.*, 2006]) qui va nous permettre de déterminer une estimation de la pose du véhicule provenant de l'odométrie visuelle à la date du GPS. L'innovation du filtre peut être calculée une fois ces deux données "synchronisées".

2.7.2.4 Prise en compte du biais et correction de l'état estimé

Nous avons montré comment calculer le biais entre l'état estimé et la vraie position. Ce biais est connu après chaque mesure du GPS et contenu dans le vecteur d'état du filtre de Kalman. Plutôt que de laisser évoluer ce biais en utilisant une fonction d'évolution, il est beaucoup plus intéressant de venir corriger l'état estimé par l'odométrie visuelle avec ce biais afin de le rendre consistant.

Afin de corriger la pose estimée il suffit d'y ajouter le biais trouvé puisque c'est ainsi que nous l'avons pensé. Cependant pour les amers utilisés par l'odométrie visuelle il faut appliquer le changement de repère de l'ancien état estimé vers le nouveau repère de la localisation corrigée. Pour cela il faut trouver le changement de repère à appliquer partant de l'état estimé à l'état vrai. Dans ce problème, une information est commune aux deux repères, c'est la position des amers dans le repère véhicule. Ces deux repères sont identiques puisque même si les amers sont mal positionnés globalement, d'un point de vue local ils sont au même endroit. La

1. **AR**chitecture d'**O**rdonnement de **C**apteurs pour la **Cr**éation d'**A**lgorithmes **M**odulaires

figure 2.27 montre cette disposition. Nous disposons alors de deux informations :

- La pose estimée du véhicule dans le repère monde. C'est cette localisation qui est entachée d'un biais dû à l'inconsistance de l'EKF.
- La pose "vraie" du véhicule dans le repère monde. Cette localisation a été calculée avec la mesure du GPS-RTK.

Afin de faire passer les coordonnées des amers du repère estimé \vec{P}_n vers leur vraie position dans le repère monde \vec{P}_n nous allons passer par deux changements de repère successifs : M_{v-m} et $M_{\hat{m}-v}$. La correction des amers se fait donc par l'équation 2.44.

$$\vec{P}_n = M_{v-m} \cdot M_{\hat{m}-v} \cdot \vec{P}_n \quad (2.44)$$

2.8 Conclusion

Nous avons présenté dans ce chapitre une solution de localisation pouvant répondre aux besoins d'un processus de guidage. L'état du véhicule est estimé en fusionnant deux approches.

La première approche cherche à déterminer la pose du véhicule suivant 6 degrés de liberté. Pour cela un processus d'odométrie visuelle basé sur un filtre de Kalman étendu est utilisé. A ce calcul de pose est ajoutée une connaissance *a priori* de l'environnement en utilisant les données fournies par un télémètre. Ce télémètre nous procure l'information de profondeur nécessaire à l'initialisation des amers dans le filtre de Kalman étendu. Cette bonne initialisation nous permet de déterminer une métrique plus stable ainsi qu'un meilleur calcul de pose. Afin de rendre le système plus efficace, nous avons proposé une perception focalisante permettant les observations des amers de la caméra une par une profitant du fait que chacune de ces observations resserre la pose des appariements suivants.

Dans un deuxième temps, une estimation intègre de cette localisation est effectuée pour pallier l'inconsistance de l'EKF. Pour cela un système d'estimation et de correction du biais de localisation a été présenté. L'estimation est effectuée par un filtre de Kalman dont la mesure est fournie par un GPS. Une fois le biais estimé, il est utilisé comme correction dans le système d'odométrie visuelle.

Chapitre 3

Modèle Numérique de Terrain

3.1 Introduction

Afin de guider un véhicule jusqu'à son but il est évident que ce véhicule doit être capable de se localiser. De plus il est important que cette localisation soit à la fois précise et intègre pour garantir d'atteindre ce but en toute sécurité. Nous avons défini dans le chapitre précédent une méthode de localisation compatible avec cet objectif. Cependant cette localisation n'est qu'une partie des informations nécessaires. Kelly ([Kelly et Stentz, 1997a, Kelly et Stentz, 1997b]) a clairement exposé les besoins d'un système de navigation autonome pour réaliser sa tâche notamment en terme de perception. En effet la perception de l'environnement d'un véhicule doit également fournir des informations liées à la traversabilité de sa trajectoire. Cette problématique est plus souvent abordée sous une notion d'*obstacle*. Une autre information nécessaire au véhicule est donc la connaissance de ces éléments perturbateurs qui peuvent empêcher le fonctionnement du guidage automatique ou encore nuire à l'intégrité physique du véhicule et/ou de ses occupants. Cependant cette dernière notion est subjective, elle dépend de plusieurs éléments :

- **Le gabarit du véhicule** influe sur la définition d'un obstacle puisque suivant la taille de ses roues notamment il pourra plus ou moins franchir un obstacle (voir figure 3.1).
- **La dynamique du véhicule** influe également car suivant sa vitesse un élément de l'environnement pourra être défini comme un obstacle ou pas.
- **La trajectoire du véhicule** vis-a-vis de cette géométrie du sol va aussi modifier la traversabilité de la trajectoire de l'environnement (un trottoir pris en travers a beaucoup moins d'impact que de face).

Il est donc difficile de poser une étiquette *obstacle/pas obstacle* simplement à partir d'une information perçue par des capteurs. La traversabilité peut être déterminée uniquement lorsque toutes les informations concernant le véhicule et l'environnement sont connues. Il est alors évident qu'il faut connaître la géométrie de



FIGURE 3.1 – Être ou ne pas être un obstacle.

l'environnement dès lors que l'on se trouve en milieu ouvert. De plus il faut disposer de cette géométrie aux endroits pertinents par rapport au véhicule pour pouvoir être efficace, c'est à dire sur la trajectoire réellement à parcourir (voir figure 3.2). Ce chapitre est consacré à la reconstruction de la géométrie d'un milieu ouvert servant de support à la trajectoire d'un véhicule évoluant de manière autonome en milieu ouvert.

Pour pouvoir se déplacer en toute sécurité dans un environnement ouvert il nous faut connaître la géométrie à l'avant du véhicule. Dans un contexte général de robotique mobile autonome, cet environnement est inconnu. Nous considérons donc que nous n'avons pas de connaissance *a priori*. Sa reconstruction passe par la perception de celui-ci par des capteurs extéroceptifs. Cependant seul l'environnement "proche" du véhicule et qui correspond aux besoins exprimés par le processus de guidage est nécessaire.

Pour reconstruire la géométrie de l'environnement il nous faut une série de coordonnées de points 3D. Cette série de points peut être référencée de deux façons différentes. On peut soit les référencer par rapport aux capteurs soit par rapport au monde. Il est plus évident de référencer les données par rapport à la structure qui les a acquises, c'est à dire par rapport au robot lui-même. Le passage du repère du véhicule au repère monde se fait en utilisant le positionnement du véhicule dans le repère monde. Cependant nous avons vu dans le chapitre précédent que ce

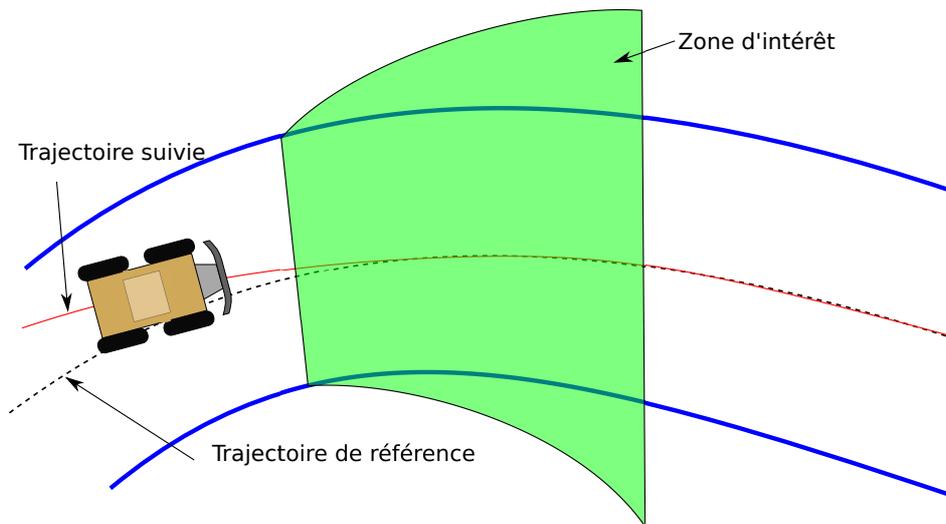


FIGURE 3.2 – La partie de l’environnement utile pour le guidage du véhicule.

positionnement pouvait être faussé. Dans ce cas la correction de positionnement doit pouvoir être appliquée aux données capteurs. Dans un premier temps, les données traitées seront donc référencées par rapport au capteur. Puis un changement de repère sera appliqué pour les avoir dans le repère monde.

Maintenant que nous connaissons les besoins de notre système il nous faut savoir de quels moyens nous disposons pour retrouver cette géométrie de l’environnement.

3.2 Approches de reconstruction de terrain

3.2.1 Introduction

Pour pouvoir réaliser notre tâche de navigation autonome il nous faut percevoir et reconstruire notre environnement. Cette reconstruction est faite, la plupart du temps, grâce à des capteurs extéroceptifs. Dans cette section nous allons faire une liste non-exhaustive des capteurs ou techniques de reconstruction utilisés pour les principaux systèmes de navigation.

3.2.2 Reconstruction de l’environnement par Stéréo-Vision

La stéréo-vision est basée sur l’analyse de deux images provenant de deux caméras synchronisées. Ces deux caméras sont liées l’une à l’autre par un lien rigide qui les maintient en place. On utilise alors le champ de vision commun aux deux caméras (voir figure 3.3) pour extraire une image de disparité. Cette image de dis-

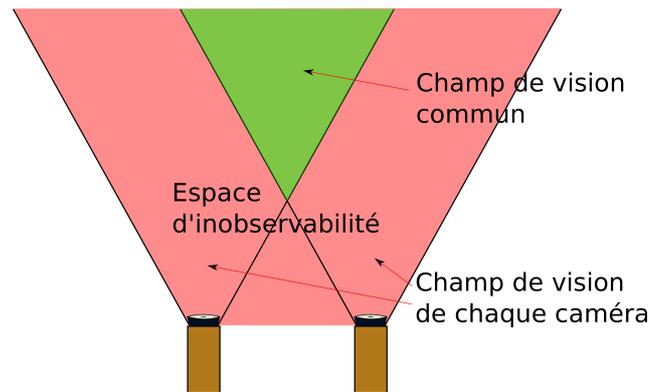


FIGURE 3.3 – Une paire stéréo et le champ de vision commun aux 2 caméras.

parité est une carte de profondeur de la vue stéréo. Mise à part la calibration des paramètres intrinsèques de chaque caméra, plusieurs étapes sont nécessaires :

- La calibration des deux caméras ensemble va permettre de connaître les paramètres extrinsèques d'une caméra par rapport à l'autre.
- Ensuite une rectification des images permet une simplification des algorithmes qui seront appliqués par la suite. La rectification consiste à transformer les deux images de sorte que les images modifiées appartiennent au même plan. Ce plan est parallèle à la droite formée par les centres optiques et choisi de façon à ce que les droites épipolaires soient horizontales dans les nouvelles images. Cette étape amène une simplification d'appariement entre les deux images puisque le correspondant d'un point d'une image sera sur la même ligne dans l'autre image (voir figure 3.4).
- Ensuite, il faut appairer les points des images de la caméra de gauche avec ceux de la caméra de droite.
- Finalement la disparité est calculée en utilisant la formule $z = -\frac{B}{d}$ où B est la distance entre les centres optiques des caméras, d est la distance qui sépare deux points appariés dans les images rectifiées et z est la profondeur calculée du point.

Au final une carte de disparité est produite (un exemple de carte de disparité est montrée en figure 3.5). Cette image de disparité a l'avantage d'être une reconstruction 3D de l'environnement proche du véhicule à un instant t . Jung ([Jung et Lacroix, 2003, Lacroix *et al.*, 2002]) utilise ces images de disparité combinées à un calcul de pose pour obtenir une carte d'élévation de l'environnement à partir d'images stéréo prises par un ballon dirigeable en basse altitude. Cette carte de disparité peut aussi être directement utilisée pour détecter des obstacles. [Broggi *et al.*, 2005] se sert de cette information pour détecter des obstacles de types "arbres", "murs", "personnes", etc en séparant les informations de disparité qui définissent le sol du reste. [Dubbelman *et al.*, 2007] fait sensiblement la même

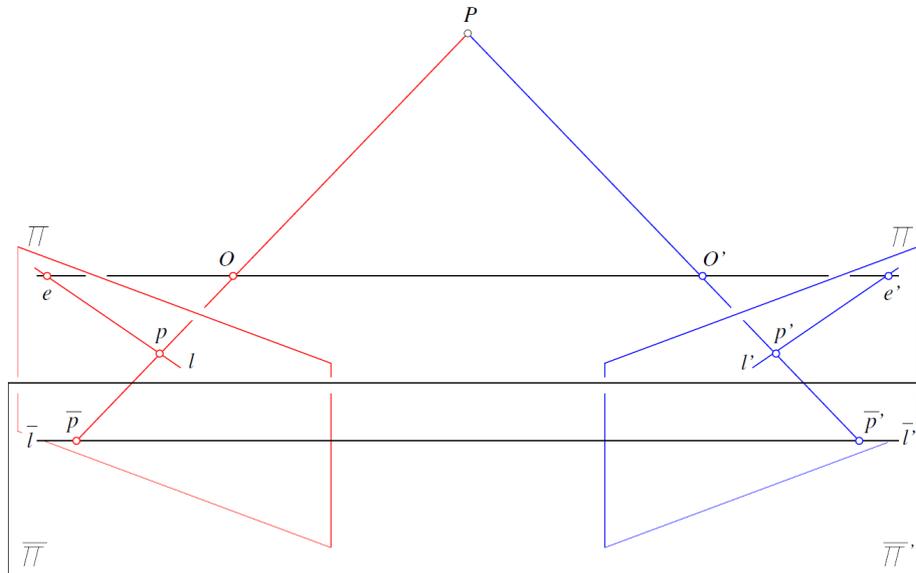


FIGURE 3.4 – La rectification de deux images stéréo. Les images des caméras qui se trouvent sur les plans π et π' sont transformées pour être exprimées dans un plan commun au deux caméras $\bar{\pi}$ (source [Forsyth et Ponce, 2002]).

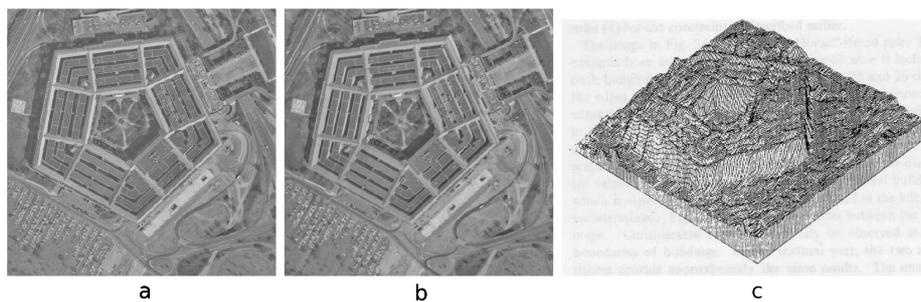


FIGURE 3.5 – Une carte de disparité produite avec une paire stéréo (résultat obtenu dans [Ohta et Kanade, 1985] réimprimé dans [Forsyth et Ponce, 2002]). (a) Image issue de la caméra de gauche. (b) Image issue de la caméra de droite. (c) Carte de disparité.

chose avec un algorithme lui permettant de détecter les obstacles en nocturne.

Cependant même si cette technique paraît intéressante, plusieurs points sont contraignants. Tout d'abord afin d'avoir une carte de disparité correcte la calibration doit être faite très précisément. De plus les perturbations subies par le véhicule à cause de l'environnement peuvent dérégler cette calibration. Ensuite, la précision des mesures de profondeur acquises décroît en fonction du carré de la distance et est dépendante de la distance entre chaque caméra de la paire stéréo. Cependant, plus cette distance est importante plus l'espace d'inobservabilité grandit (voir figure 3.3). Ce qui signifie que les possibilités atteignables par le véhicule vont dépendre de la configuration initiale du banc stéréo. Le système est donc figé. De plus, l'obtention de la carte de disparité passe par plusieurs étapes assez coûteuses en temps de calcul. Il y a aussi des problèmes pour calculer la disparité lorsque les textures des images sont mauvaises car la mise en correspondance ne se fait pas. La qualité de la reconstruction est donc dépendante de l'environnement. Finalement cette information obtenue à partir de caméras stéréo nous renseigne uniquement sur la géométrie du terrain. Le système a besoin aussi de géo-localiser cette environnement, des étapes supplémentaires sont alors nécessaires pour combler un système déjà très consommateur de ressource.

3.2.3 Reconstruction de l'environnement par télémètre 3D dense

Ces dernières années, de nouvelles solutions technologiques de laser 3D sont apparues comme par exemple le HDL-64E de la société *Velodyne* ([*Velodyne, 2007*]). Ces technologies ont l'avantage de produire un nuage de points 3D de façon "instantanée" (c'est à dire que les 3 coordonnées d'un point sont déterminées en même temps). Pour cela, plusieurs lasers 2D sont embarqués au sein d'un même dispositif. Ensuite, par rotation autour d'un axe, ces lasers produisent des observations sur 360°. De ce fait ces capteurs sont capables de fournir plusieurs milliers de points 3D par seconde. Il n'y a donc pas de traitement à effectuer pour obtenir une carte d'élévation. Cependant le prix de ces appareils est prohibitif (le fameux HDL-64E de chez Velodyne coûte 75000 dollars). Aussi en considérant une éventuelle commercialisation d'une solution de reconstruction de l'environnement complet, le prix de ce capteur seul peut dépasser le prix du véhicule à équiper. De plus, tout comme les paires stéréo, ce capteur ne fait que procurer un nuage local de points référencés par rapport au véhicule (dont beaucoup se situent à l'arrière du véhicule, donc très peu utiles dans notre contexte). Il faut donc, en plus des mesures effectuées, calculer le déplacement du véhicule d'un instant à l'autre. Pour cela des algorithmes de point le plus proche (Itérative Closest Point : ICP) sont utilisés pour retrouver le changement de repère d'une mesure à l'autre. Rusinkiewicz ([*Rusinkiewicz et Levoy, 2001*]) a comparé l'algorithme ICP suivant

différents types de sélection de points dans le nuage et de mise en correspondance de ces points. Nuechter ([Nüchter *et al.*, 2005]) par exemple utilise cette méthode pour estimer la pose de son robot en utilisant le nuage de points généré par un télémètre 2D rotatif. Même si ce capteur est suffisant si il est couplé à un système de localisation comme celui montré en chapitre 2, son défaut est principalement économique.

3.2.4 Reconstruction de l'environnement par SLAM

Le géo-référencement de la carte de l'environnement est primordial pour réaliser la navigation autonome du véhicule. Nous avons vu dans ce chapitre que notre odométrie visuelle était capable d'estimer la position d'amers 3D référencés dans un repère monde. C'est le principe même de la localisation et cartographie simultanées (SLAM). Dans leurs articles, Durant ([Durrant-Whyte et Bailey, 2006]) et Bailey ([Bailey et Durrant-Whyte, 2006]) ont synthétisé les avancées de la recherche en ce qui concerne les méthodes de SLAM. Cependant les coûts opératoires engendrés par le SLAM sont importants quelle que soit la méthode utilisée (EKF, filtre à particules, etc...). Ces méthodes sont limitées à une centaine de points dans la carte avec les systèmes informatiques actuels. Ensuite, lorsque cette carte atteint une taille maximum, elle est détachée du système et forme une carte locale. En prenant l'intégralité de ces cartes on obtient la carte de l'environnement parcouru.

Si on oublie les problèmes de temps de calcul, ces méthodes remplissent bien le rôle que l'on attend à un détail près. Le SLAM est très souvent utilisé dans un contexte d'exploration. Pour cela on utilise des amers remarquables pour décrire l'environnement comme expliqué en section 2.5. Ces amers ne sont pas définis de façon assez dense pour calculer la traversabilité de la trajectoire parcourue. De plus ces amers sont initialisés par des points d'intérêts dans une image. Dans notre cas ces points ne sont pas toujours disponibles ou ne sont pas localisés de façon appropriée pour notre application.

3.2.5 Bilan des méthodes de reconstruction de l'environnement

Nous venons de voir les principales méthodes de reconstruction de l'environnement utilisées par des systèmes actuels de navigation. Dans un premier temps, les systèmes de mesure par télémètre 3D semblent les plus intéressants puisque qu'ils permettent une mesure directe de l'environnement proche sans pour autant nécessiter de moyens supplémentaires. Cependant si l'on souhaite géo-référencer, mettre à jour la carte à l'avant du véhicule lorsqu'il se déplace ou encore combiner cette carte locale avec une carte globale existante des moyens supplémentaires doivent

être employés. De plus le coût de ces appareils sont un frein pour des éventuelles applications grand public.

En stéréoscopie, les informations de l'environnement proche sont extraites à partir des correspondances des deux images. Puis, de la même façon que pour un télémètre 3D, il faut utiliser d'autres moyens pour remonter à une carte géo-référencée. Cependant les informations ne sont pas disponibles aussi directement que pour les télémètres 3D.

Finalement les systèmes SLAM permettent de créer une ou des cartes géo-référencées. Cependant les coûts en puissance de calcul sont beaucoup plus importants que pour des systèmes de stéréoscopie ou de Lidar 3D.

3.3 Approche de modèle numérique de terrain par couplage caméra télémètre

Nous avons vu que le processus de guidage a besoin, pour se déplacer, d'une carte du terrain à l'avant du véhicule relativement dense. La finalité de cette carte d'élévation est de calculer la traversabilité de la trajectoire prédite afin de déterminer si oui ou non le véhicule pourra atteindre son but en maintenant son intégrité physique. Comme nous l'avons vu le processus d'odométrie visuelle par filtre de Kalman étendu nous permet d'estimer dans le même processus la position 3D d'un amer et la pose du véhicule. Dans notre application nous disposons d'un télémètre mono-nappe orienté vers le sol qui est utilisé pour nous aider à la localisation. Ce télémètre effectue une mesure de distance au sol dans le repère du véhicule. Aussi, comme le véhicule se déplace, cette mesure est différente à chaque acquisition. Si on est capable de connaître parfaitement le déplacement du véhicule entre deux périodes d'échantillonnage du télémètre on peut donc obtenir une carte 3D discrétisée de l'environnement constituée de l'ensemble des nappes lidar en utilisant seulement un télémètre 2D. Dans ce cas cette carte 3D est très précise.

Au chapitre précédent nous avons proposé une méthode de localisation précise et intégrée se basant sur la combinaison de plusieurs capteurs. En combinant cette méthode de localisation 6D avec l'intégralité des informations perçues par le télémètre nous pouvons accéder à une reconstruction 3D de l'environnement à la fois dense et précise.

Cette idée nous permet alors d'avoir un système complet de positionnement et de reconstruction d'environnement basé sur seulement deux capteurs au coût somme toute modeste. Ce système basé sur des capteurs complémentaires va nous permettre de satisfaire les besoins du processus de navigation.

3.3.1 Positionnement Espace-Temps d'une donnée Lidar dans le Modèle Numérique de Terrain

Afin d'utiliser les données télémétriques comme informations principales dans notre reconstruction d'environnement nous avons besoin de positionner ces données dans le référentiel de notre odométrie visuelle. Ce problème est le même que celui présenté en section 2.5.2.3 lorsque nous devons repositionner dans le temps les données issues du télémètre à une date valide pour la caméra. Cette fois ci, il ne nous est pas nécessaire de remettre les données dans le repère de la caméra. Il nous suffit seulement d'estimer la pose du véhicule à l'instant de la donnée Lidar pour calculer le changement de repère à effectuer pour positionner la nappe télémétrique entière en utilisant l'équation (2.25). Nous pouvons donc à partir du même raisonnement que celui de la section 2.5.2.3 effectuer un positionnement espace-temps d'une donnée lidar pour l'estimation de localisation et pour reconstruire le monde.

3.3.2 Mise à jour du MNT lors d'une correction de la dérive de localisation

Tout comme pour la section 3.3.1, l'influence de la correction de la dérive lors de la prise en compte de données intègres est la même que ce soit pour corriger les points 3D du vecteur d'état ou les points du MNT. Lorsque l'on met à jour notre position dans le repère global notre nuage de points 3D (qui est toujours le même localement) doit être corrigé "globalement" pour rester au même endroit "localement". Cette correction se manifeste par un changement de repère de l'ancien repère *biaisé* vers le nouveau repère intègre. Pour cela nous allons utiliser le même principe que celui exposé dans la section 2.7.2.4 où nous appliquions la correction de l'état sur les points 3D de notre vecteur d'état. Cette fois-ci ce sont les points 3D dans la carte d'élévation qui doivent être corrigés. Cela est effectué en utilisant l'équation 2.44 qui nous a servi à corriger les points du vecteur d'état de l'odométrie visuelle. L'équation (3.1) réalise donc cette correction. Une fois de plus, le même principe est utilisé pour réaliser deux tâches dont le but est différent : la localisation du véhicule et l'estimation du MNT.

$$\vec{P}_{MNT_X} = \mathbf{M}_{(v-X)} \cdot \mathbf{M}_{(\hat{X}-v)} \cdot \vec{P}_{MNT_{\hat{X}}} \quad (3.1)$$

3.4 Stratégie de gestion de la carte de l'environnement

A ce stade, nous avons les moyens d'accumuler des nappes télémétriques dans une carte d'élévation de l'environnement. Cependant l'accumulation en perma-

nence de données dans le modèle numérique de terrain amène une consommation de ressource très importante par rapport au reste du système. En effet le système de guidage a besoin des points de la carte dans le repère local du véhicule. Un changement de repère est donc nécessaire pour ramener ces points du repère monde vers le repère local. Ce changement de repère est une simple roto-translation des coordonnées du point. Cependant lorsqu'il est effectué sur plusieurs dizaines de milliers de points le temps de calcul de cette opération devient vite prépondérant sur tout le reste.

Suivant les besoins de l'application, plusieurs possibilités s'offrent à nous pour la gestion de ces points. Dans notre application de robotique mobile autonome, nous cherchons à rejoindre une destination géo-référencée. Cette carte a pour but d'être consultée par le processus de guidage afin d'y trouver les informations pertinentes pour réaliser sa tâche. Il n'est donc pas nécessaire de numériser le "monde entier" pour accomplir cette mission. Seul l'environnement proche du véhicule sur la trajectoire parcourue est pertinent. De plus les phénomènes de fermeture de boucle n'apparaissent *a priori* pas ou très peu dans ce contexte particulier. Dans la littérature, les détections de fermeture de boucle sont très souvent faites dans un contexte d'exploration.

Une de nos stratégies consiste alors à supprimer les points devenus inutiles dans le MNT c'est à dire ceux qui sont derrière le véhicule (figure 3.6). Les points destinés à être supprimés sont choisis suivant la relation :

$$(0 \ 0 \ 1) \cdot \mathbf{R}_{(v-m)}^T \cdot (P_{MNT} - \vec{T}_{(v-m)}) < 0 \quad (3.2)$$

$$\mathbf{R}_{(v-m)_3}^T \cdot P_{MNT} < \mathbf{R}_{(v-m)_3}^T \cdot \vec{T}_{(v-m)} \quad (3.3)$$

où $\mathbf{R}_{(v-m)_3}^T$ représente la 3^{ème} ligne de la matrice de rotation $\mathbf{R}_{(v-m)}^T$.

De plus, l'ajout de nouveaux points dans le MNT se fait en ne gardant que les points du télémètre qui satisfont la contrainte du processus de guidage. C'est à dire qu'ils doivent appartenir à la zone d'observation définie par le système de commande.

Le nombre de points du télémètre dans le sens horizontal nous assure une densité suffisante de notre carte dans ce même sens. Cependant la densité dans le sens de déplacement du robot n'est pas automatiquement assurée puisque elle est dépendante du contexte d'évolution du véhicule.

3.4.1 Interaction du processus de guidage pour les besoins de perception

Lors de l'acquisition du MNT, certaines configurations comportent des parties de l'environnement qui ne sont pas observables par le télémètre. On peut notamment citer la partie de l'environnement masquées par un arbre ou encore lorsque

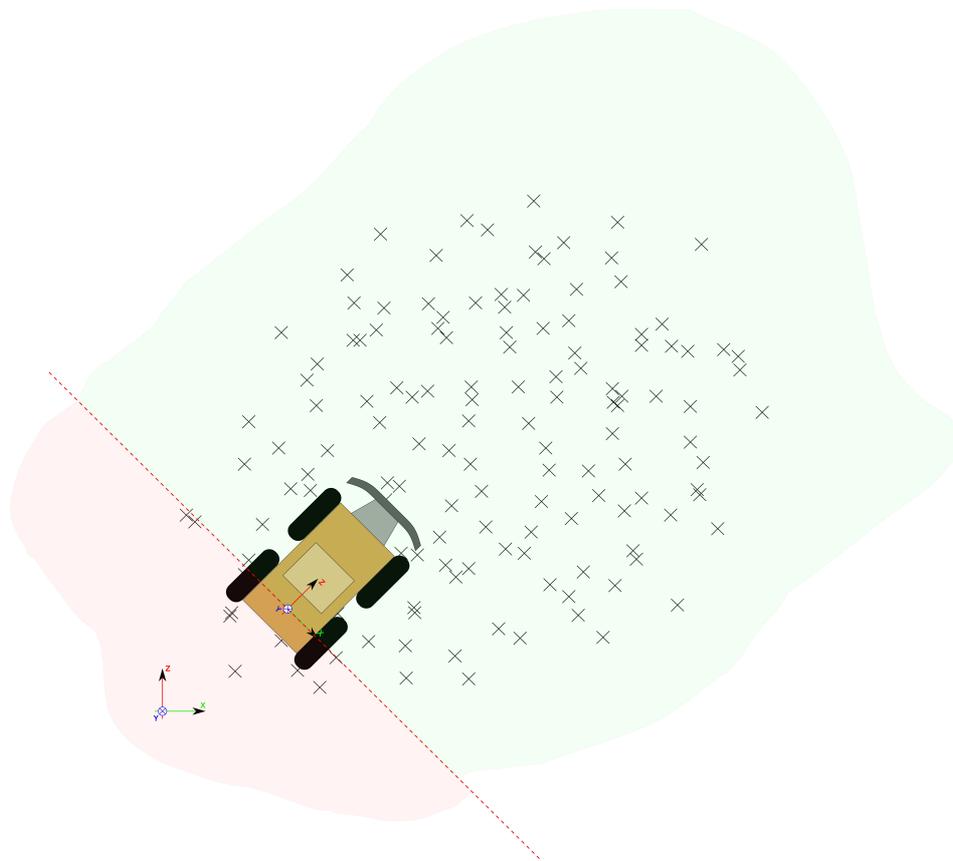


FIGURE 3.6 – Sélection de points : les points qui passent derrière le véhicule sont supprimés de la carte.

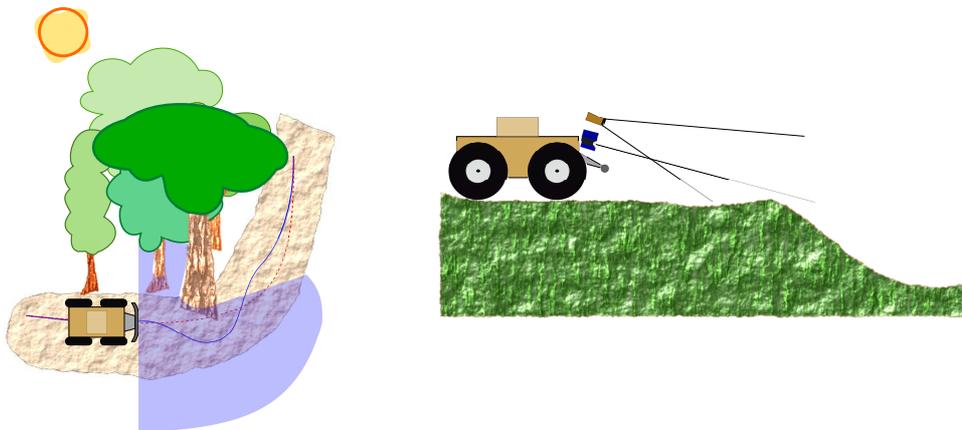


FIGURE 3.7 – Configurations où des espaces non-observables de l'environnement apparaissent.

le véhicule arrive vers une pente descendante ou si tout simplement la vitesse du véhicule devient trop grande vis-a-vis de la fréquence d'acquisition du télémètre. La figure 3.7 montre certaines de ces configurations.

Cependant la trajectoire de référence peut passer proche ou même sur ces parties. Il existe alors un espace qui doit être analysé pour être exploité par la tâche de guidage alors qu'il n'est pas observé par notre système de perception. Cet espace ne peut pas rester inobservé sous peine de voir échouer la mission de guidage automatique. Lorsque cette configuration arrive, le système décisionnel (qui connaît quelles sont les zones comprenant son déplacement) prend en compte cette absence d'information et demande au processus de perception de focaliser son attention sur cette zone. D'un point de vue applicatif, cette zone doit être échantillonnée en plusieurs points. Le nombre de points est déterminé par le processus décisionnel au regard du contexte (vitesse, taille du véhicule, etc...). Ces points sont connus en longitude et attitude puisque c'est l'information attendue. Il reste donc à déterminer l'altitude de ces points que nous estimerons grâce au processus d'odométrie visuelle exposé au chapitre 2. Dans ce cas, les points ne sont pas initialisés par des mesures Lidar (c'est à dire avec une incertitude initiale très faible) mais de façon plus "classique" avec une incertitude importante. La complémentarité caméra-télémètre nous sert donc dans ce cas à répondre aux exigences non-encore satisfaites du système de navigation. Ici c'est donc la caméra qui va assurer la densité du MNT en estimant l'élévation des points 3D réclamés par le système de guidage.

L'interaction se fait aussi lorsque le système de perception devient insuffisant. Dans le cas où le système de perception n'est pas capable de fournir des données suffisantes au processus de guidage ce dernier peut alors prendre les dispositions nécessaires à la sécurisation du déplacement du véhicule. Ces dispositions sont la libération de ressources de calcul ou le ralentissement du véhicule par exemple.

On peut imaginer alors que lorsque la vitesse du véhicule devient très grande la densité du MNT demandé par le système de navigation dépasse de beaucoup les capacités de perception. Il faut dans ce cas faire savoir au processus de guidage que l'on est incapable de fournir les informations demandées. Dans ce cas le véhicule va ralentir de lui même et va implicitement placer son point de fonctionnement de façon optimale au vue des capacités de perception/action. C'est cette interaction permanente entre le système de perception et le système de navigation qui doit rendre robuste, intègre et efficace notre démarche et permet d'atteindre l'objectif fixé en toute sécurité.

3.5 Conclusion

Dans ce chapitre nous avons exposé une méthode de reconstruction de la géométrie de l'environnement. Le couplage d'un télémètre 2D avec le système de localisation 6D décrit au chapitre précédent nous a permis de remonter à un nuage de point 3D dense. Pour cela un système de fusion de données a été utilisé. Ce système de perception permet de retrouver la carte d'élévation de façon précise et à moindre coût. De plus, il est possible d'estimer la géométrie de l'environnement aux endroits invisibles du télémètre grâce à la présence de la caméra. Une stratégie intelligente de guidage permet alors de mettre en œuvre cette perception de façon à satisfaire les besoins de la navigation autonome et d'informer cette dernière sur les capacités du système de perception.

Chapitre 4

Résultats expérimentaux

Dans ce chapitre, nous allons exposer les résultats que nous avons obtenus avec notre système de perception. Les résultats de la partie localisation par fusion de données et ceux de reconstruction de carte d'élévation seront montrés et discutés. Plusieurs approches seront expérimentées. La faisabilité du système sera faite en utilisant le système de calcul matriciel Matlab. Ensuite nous évaluerons les performances de notre système sur un système proche d'un système réel grâce au simulateur de robotique que nous avons développé. Finalement les résultats des données réelles faites sur notre plateforme expérimentale **AROCO**¹ seront présentés.

4.1 Résultats théoriques sur MATLAB

Dans un premier temps, la faisabilité du système de localisation par fusion caméra-télémètre doit être démontrée. De plus, afin de montrer la pertinence de notre approche, la comparaison doit être faite entre notre système de fusion et un système sans fusion de données. Nous allons aussi comparer les résultats de localisation obtenus par un EKF avec des systèmes basés sur des techniques de géométrie épipolaire et de triangulation analytique.

4.1.1 Procédure de test

Dans ces résultats nous allons comparer plusieurs approches :

- Une localisation par géométrie épipolaire. Pour cela nous utilisons les correspondances de points d'une image à l'autre pour calculer la rotation et translation entre chaque itération. Nous nous appuyerons sur le code fourni par Stewénius [Stewénius *et al.*, 2006].

1. **AROCO** : Autonomous Robot - Off road COntrol applications

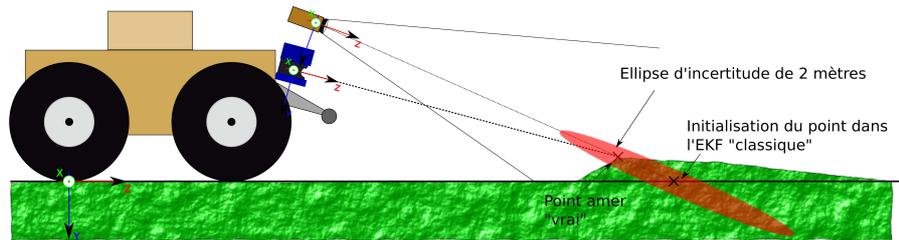


FIGURE 4.1 – L’initialisation d’un amer dans l’EKF "classique" à partir d’un point télémètre.

- Une localisation par triangulation de la position. En partant de l’hypothèse que les coordonnées des points sont connues nous allons chercher à calculer la pose de la caméra par rapport à leur projection dans l’image. Le code présenté par Araujo [Araujo *et al.*, 1998] sera utilisé.
- Une localisation par EKF et fusion avec données télémétriques. Les différentes méthodes d’observation des points dans la caméra seront testées.
- Une localisation par EKF "classique". Dans cette méthode les amers sont initialisés comme des points caméra classiques. Pour cela les amers proviendront aussi du télémètre mais seront transformés pour être situés sur un sol plan par rapport au véhicule ($P_{y_n} = 0$) tout en gardant leur projection dans l’image au même endroit (voir figure 4.1). L’incertitude associée est fixée à 2 mètres en profondeur. Les différentes méthodes d’observation des points dans la caméra seront testées.
- Une localisation par odométrie où seul le modèle d’évolution du véhicule est utilisé pour estimer la position du véhicule.

Afin de comparer les résultats obtenus nous devons disposer d’une même procédure de test pour notre système et les autres approches. Pour cela, la trajectoire utilisée avec les différents systèmes sera la même. De plus, nous devons utiliser les mêmes amers de référence pour ne pas favoriser une approche par rapport à une autre. Aussi les points image utilisés pour mettre à jour chacune de ces méthodes seront les projections des vrais amers. Pour réaliser cette procédure de façon équivalente pour chaque méthode nous utiliserons donc comme "base" un système où les points proviennent d’un télémètre. Puis, ces points télémétriques seront soit transformés, soit bruités, soit utilisés tels quels afin de correspondre à la méthode utilisée. L’incertitude associée à ces points sera faite de façon à correspondre à la méthode utilisée.

Le robot "virtuel" suit une trajectoire préalablement définie. En même temps, les différentes méthodes évaluent le chemin parcouru en fonction des observations données par le robot. Le scénario qui est joué par cette partie simulation pure décrit une trajectoire en 'S'. La trajectoire comporte aussi plusieurs dénivelés entre le

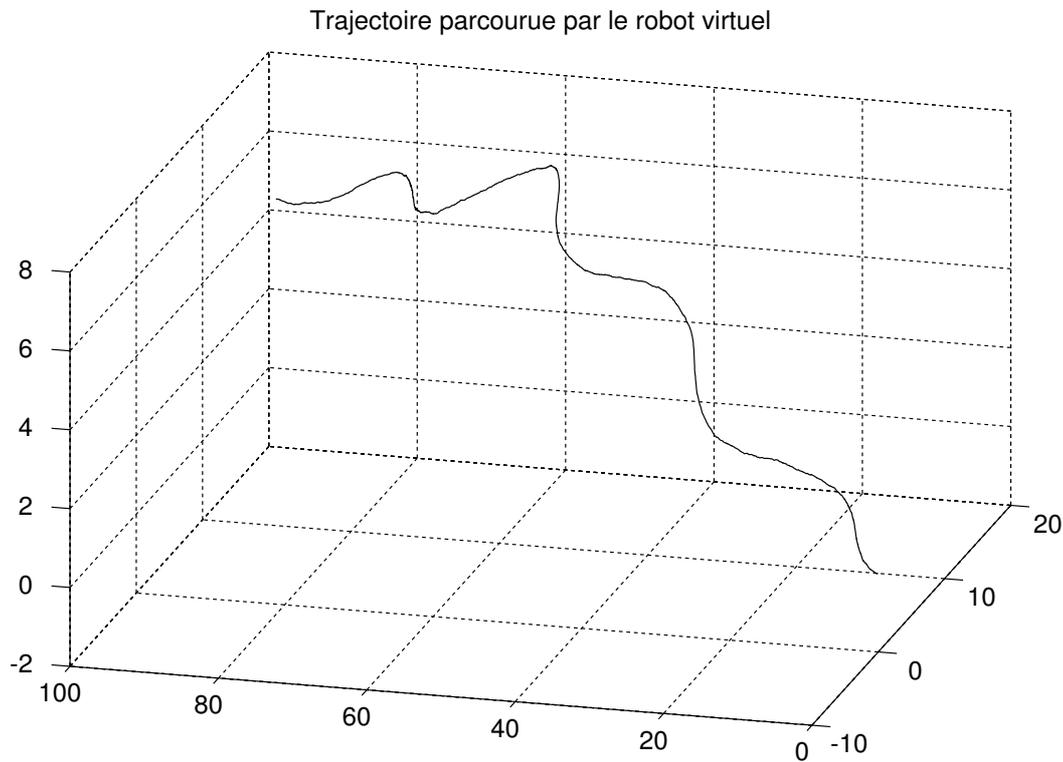


FIGURE 4.2 – Vue de la trajectoire effectuée par le robot virtuel.

début du parcours et la fin. Le lacet et le tangage du véhicule évoluent de façon sinusoïdale. Le véhicule virtuel avance par pas de 5cm (la norme de la distance entre deux itérations), toutes les 10 itérations une mesure télémétrique est ajoutée à la carte de sorte qu'à chaque instant toutes les méthodes possèdent les mêmes points dans leur carte. Seules leurs initialisations auront été faites différemment. Les mesures télémétriques sont des mesures de profondeur faites sur un sol bosselé (écart type 30cm). Les différentes méthodes disposent d'une odométrie bruitée par un bruit blanc gaussien d'écart type $\sigma_{odo} = 1cm$ entre chaque itération. Finalement, un bruit blanc gaussien est ajouté sur les 6 composantes de la pose du robot pour matérialiser des imperfections du chemin emprunté. La distance parcourue par le robot fait donc environ 100 mètres. La trajectoire empruntée est montrée en figure 4.2. Cette procédure est réitérée 100 fois afin de comparer statistiquement plusieurs réalisations de chaque processus de localisation.

Les amers sont observés par une caméra virtuelle de résolution 800x600 d'angle de vue de $\pi/4$ horizontalement dont la matrice de paramètres intrinsèques K est notée en équation 4.1. Le bruit de mesure sur cette caméra est seulement représenté par l'arrondi au pixel de l'amer observé. Ces paramètres ont été choi-

sis pour coïncider avec les paramètres de la caméra utilisée dans notre simulateur (voir section 4.2.1.1) afin de mettre en correspondance les résultats théoriques et les résultats pratiques.

$$\mathbf{K} = \begin{pmatrix} 724.264 & 0 & 400.5 \\ 0 & 724.264 & 300.5 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

4.1.2 Test des différentes méthodes de positionnement

Dans cette partie, nous allons comparer la méthode d'estimation de pose du véhicule par fusion caméra-télémetre aux différents algorithmes discutés dans le chapitre 2. Pour garder la lisibilité des figures nous montrons seulement les résultats de la méthode globale pour les méthodes par EKF. Dans la figure 4.3 on peut voir les résultats de chacun des procédés pour accomplir cette tâche de localisation (les erreurs de localisation sur les 100 réalisations des processus sont montrées en figure 4.4).

Les résultats de localisation des différentes approches sont montrés en figures 4.4, 4.5, 4.6, 4.7. Dans un premier temps nous pouvons constater la plus value qu'apporte notre système de fusion de données par rapport à un EKF classique (l'erreur de localisation dans la figure 4.4 est plus petite en prenant en compte les données télémétriques).

Le fait d'avoir une information précise sur la position initiale d'un amer apporte une grande précision de l'algorithme en terme de métrique. Cette précision a ensuite un impact sur les estimations de la pose du véhicule notamment en terme d'estimation de l'orientation comme nous pouvons le voir en figure 4.6. Finalement l'erreur en terme de position est réduite comme prévu (figure 4.5 et 4.4). On voit par exemple que l'erreur moyenne de position à la fin du parcours est d'environ 40m pour un système sans utilisation de données Lidar. Le système par fusion de données ajoutant le télémetre propose une erreur de localisation en fin de parcours inférieure à 50cm. La méthode de localisation par géométrie épipolaire a été effectuée en utilisant un algorithme robuste de type RANSAC. Généralement cette méthode produit une bonne estimation du déplacement du véhicule. Pour ce qui est du calcul de pose par triangulation les points 3D utilisés sont les coordonnées des points télémétriques. De ce fait le calcul de pose est très bon lui aussi mais reste inférieur à la méthode par filtre de Kalman étendu. De plus cette méthode ne met pas à jour les coordonnées 3D des points utilisés. La figure 4.7 montre le pourcentage de l'erreur de métrique entre la vraie distance parcourue et les estimations (le tableau 4.1 montre les valeurs numériques de ce tableau). On peut voir que l'apport du télémetre diminue grandement les erreurs d'estimations de métrique du filtre de Kalman étendu. La métrique calculée par le filtre de Kalman sans les données du

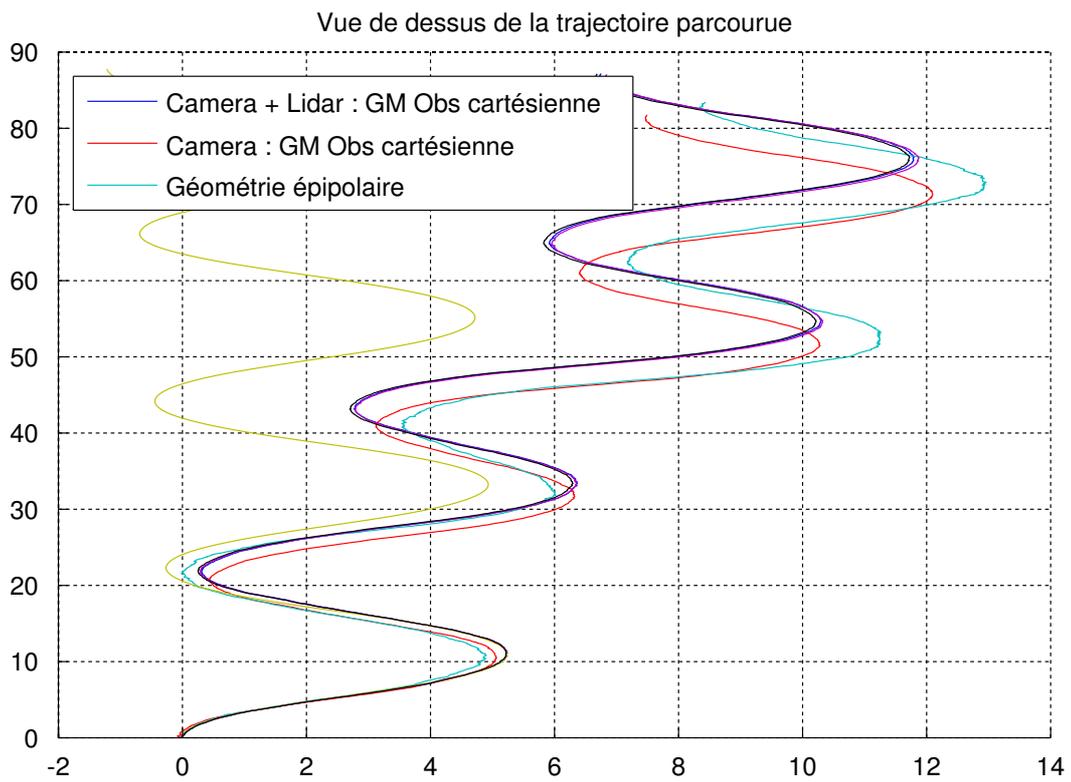


FIGURE 4.3 – Vue de dessus de la trajectoire effectuée par le robot virtuel ainsi que les trajectoires estimées par les différents types de localisation (GM : Méthode globale).

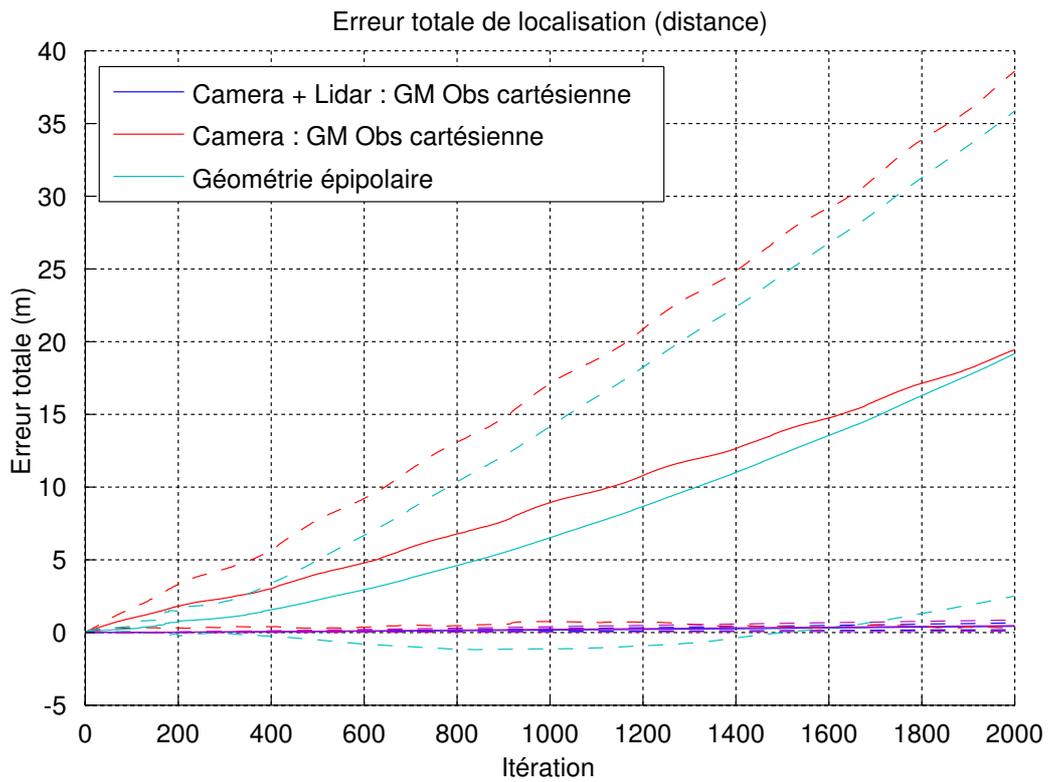


FIGURE 4.4 – Erreur totale moyenne des 100 réalisations des processus pour les différents types de localisation. Les pointillés représentent la courbe à \pm un écart type. 2000 itérations correspondent à environ 100 mètres.

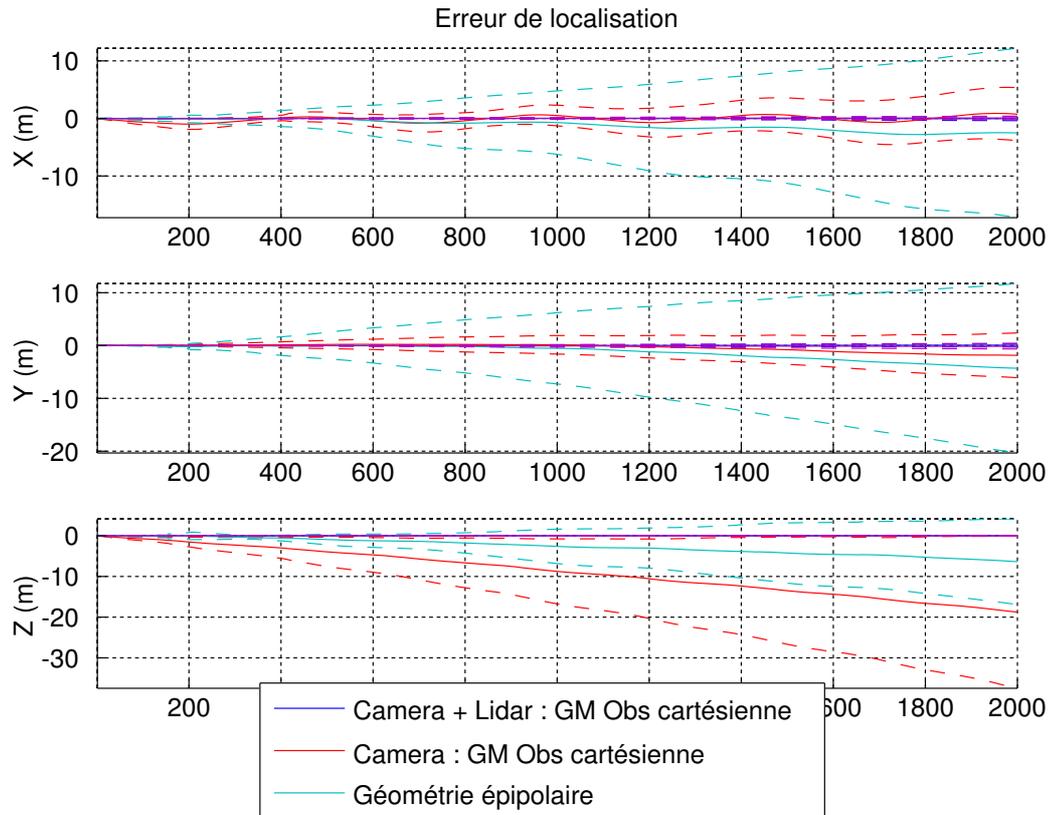


FIGURE 4.5 – Erreur moyenne de localisation des 100 réalisations en X, Y et Z de chaque méthode par rapport à la trajectoire de référence. Les pointillés représentent la courbe à \pm un écart type.

télémetre est mauvaise (jusqu'à 40% d'erreur de métrique estimée) ce qui entraîne une erreur de localisation en distance parcourue importante (visible notamment sur la coordonnée en Z de la figure 4.5).

4.1.3 Test des méthodes globales et point par point

Dans un deuxième temps, nous avons comparé les approches par méthode globale (nommée GM) et point par point (nommée PPM) proposées en section 2.6 du chapitre 2 en utilisant l'observation cartésienne. Nous rappelons que dans la méthode globale nous utilisons toutes les observations faites des amers dans la caméra en une seule fois alors qu'en méthode point par point les observations sont utilisées une par une. Les erreurs commises par chacune des méthodes sont montrées en figure 4.10. On peut y voir que la précision de la méthode point par point est similaire à la méthode globale. La figure 4.9 nous montre que l'orientation du vé-

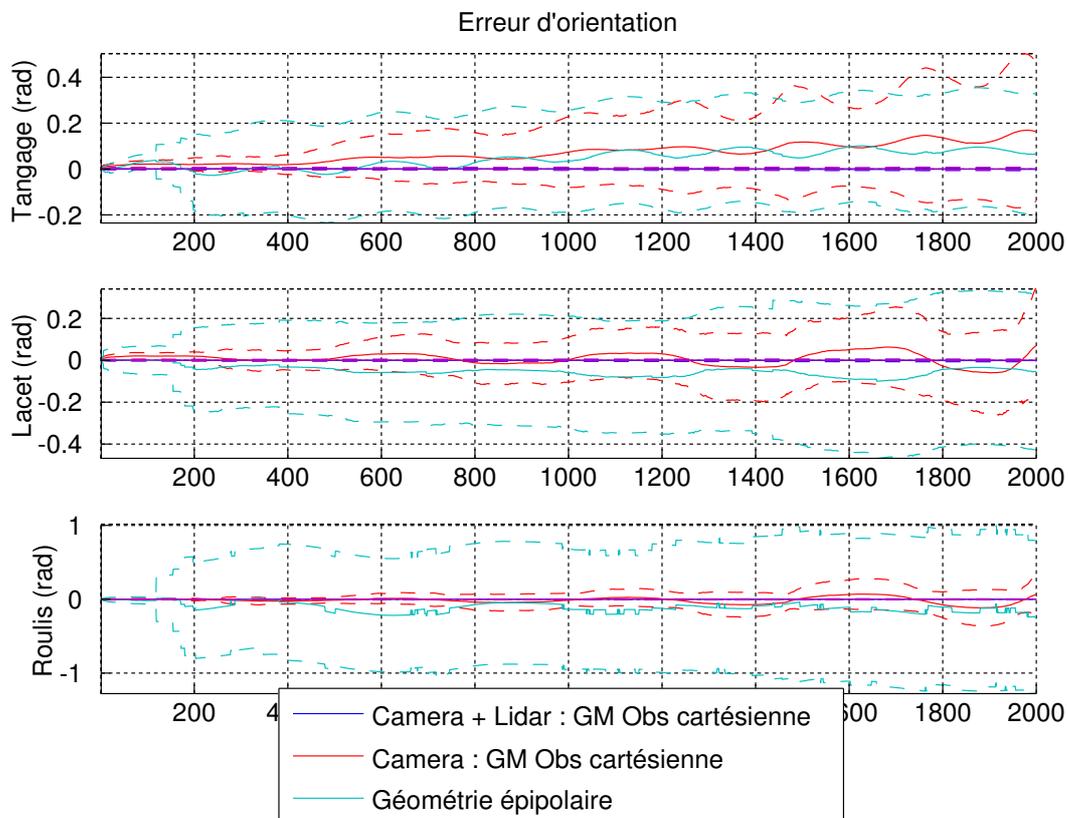


FIGURE 4.6 – Erreur moyenne d'orientation des 100 réalisations en tangage, lacet, roulis (α , β et γ) de chaque méthode par rapport à l'orientation de référence. Les pointillés représentent la courbe à \pm un écart type (pour les méthodes par caméra + lidar et par triangulation les courbes sont confondues).

Méthode	Erreur de métrique (%)		
	moyenne	maximale	écart type
Caméra + Lidar : GM Obs cartésienne	3	-3.4031	5
Caméra : GM Obs cartésienne	6	-22.1031	5
Géométrie épipolaire	4	109.983	5
Triangulation	9	3.3248	7
Odométrie seule	6	7.9615	1

TABLE 4.1 – Récapitulatif du pourcentage d'erreur de métrique suivant les différentes méthodes.

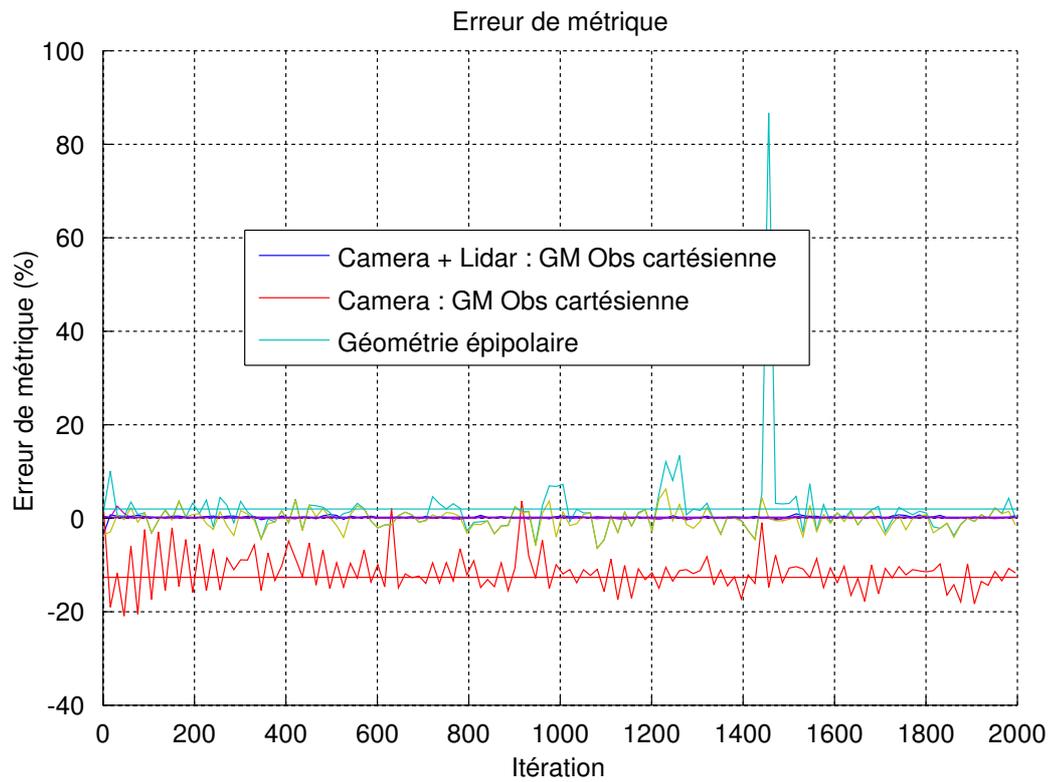


FIGURE 4.7 – Erreur moyenne commise sur la métrique des 100 réalisations de chaque méthode. Les données précises du télémètre diminuent grandement les erreurs de métrique.

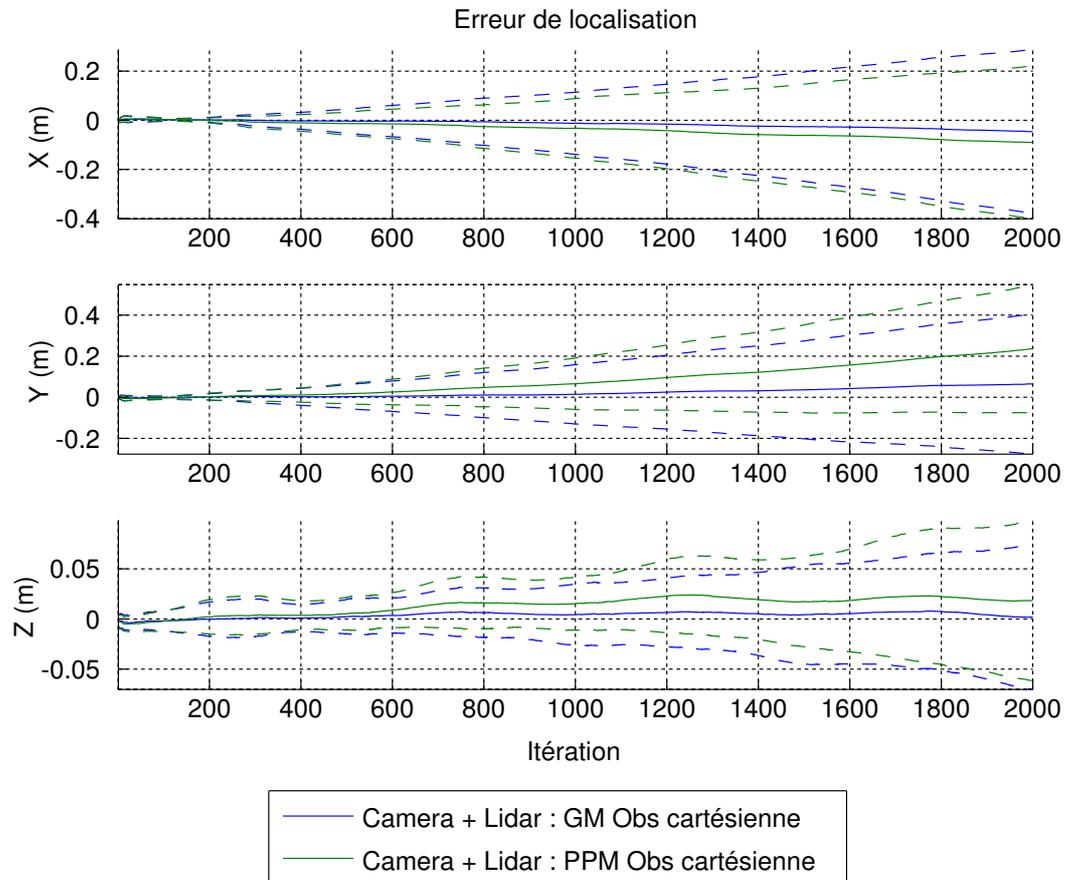


FIGURE 4.8 – Erreurs moyenne de localisation des 100 réalisations en X, Y et Z en méthode globale et point par point.

hicule est estimée en méthode point par point aussi correctement qu'en méthode globale. Les estimations de localisation du véhicule en figure 4.8 sont donc très proches d'une méthode à l'autre.

Nous avons montré que la méthode point par point pouvait estimer la pose du véhicule avec une précision équivalente à la méthode globale. L'aspect intéressant dans cette méthode est de pouvoir réduire le champ de recherche des mises en correspondance au fur et à mesure de l'estimation de l'état véhicule.

4.1.4 Test des différentes méthodes d'observation

Ici nous allons montrer les résultats obtenus en utilisant les différentes méthodes d'observation décrites au chapitre 2 en méthode point par point. La comparaison sera faite entre l'observation cartésienne et l'observation sphérique en ob-

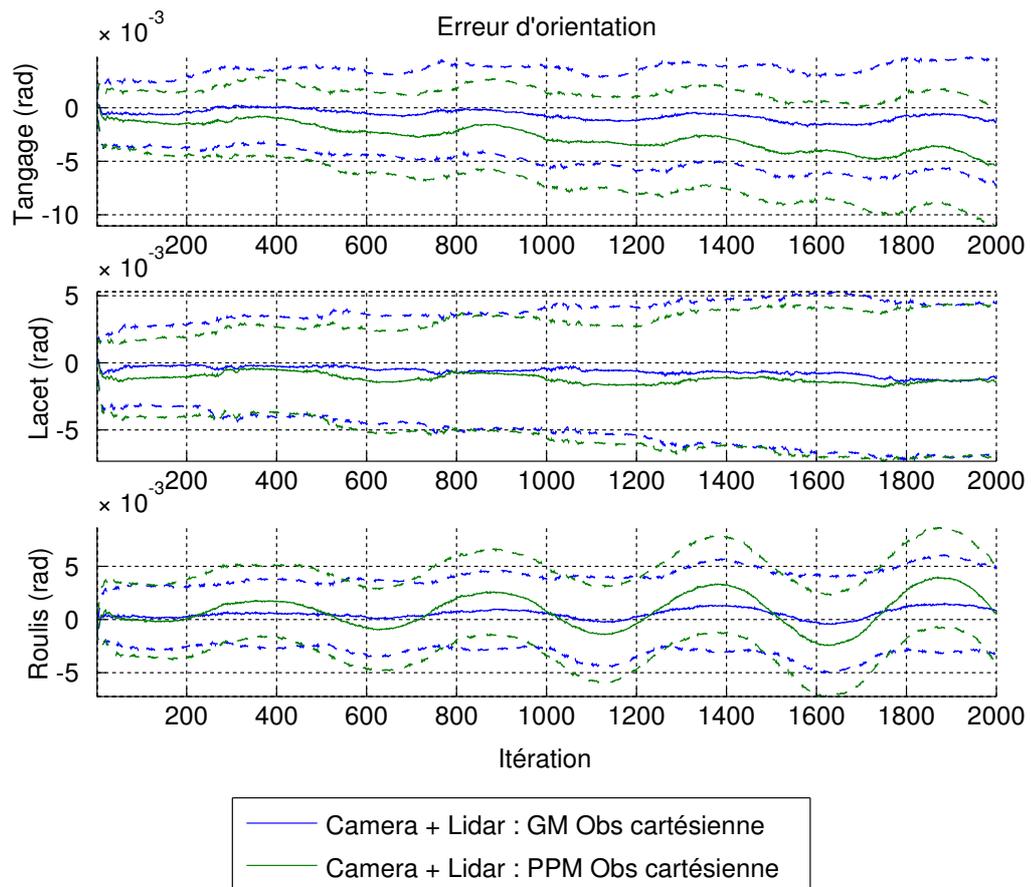


FIGURE 4.9 – Erreurs moyenne d'orientation des 100 réalisations en tangage, lacet et roulis (α , β et γ) en méthode globale et point par point.

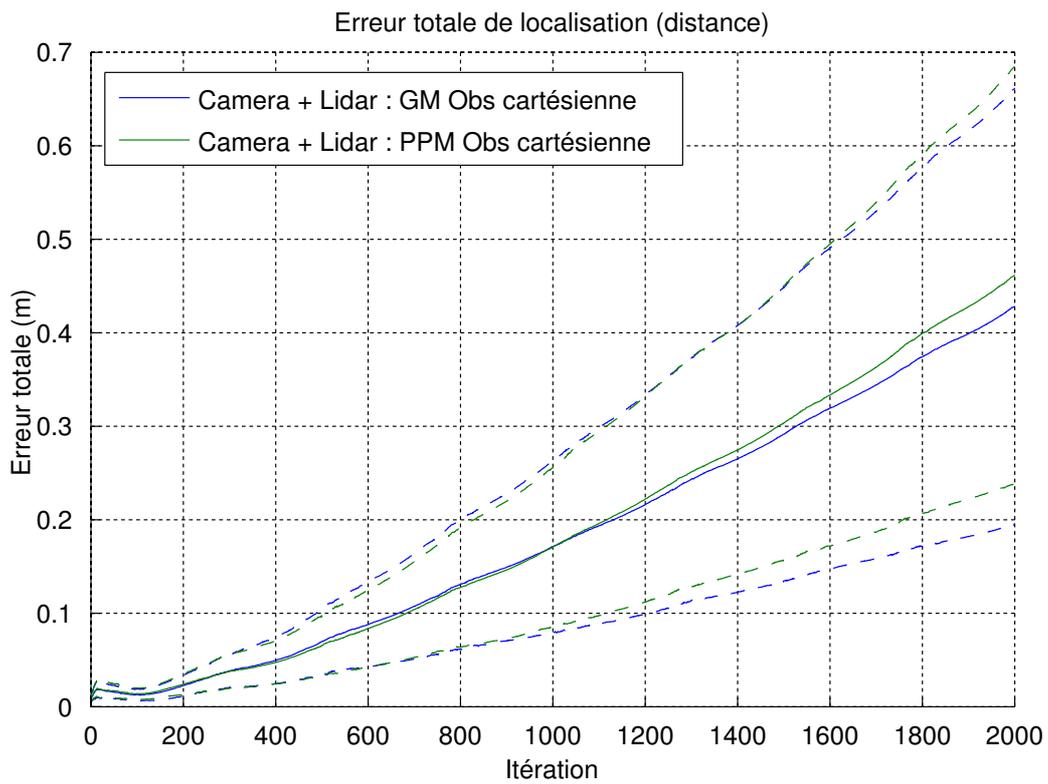


FIGURE 4.10 – Erreur totale moyenne des 100 réalisations en méthode globale (GM) et point par point (PPM). Les deux méthodes sont similaires en terme de résultats.

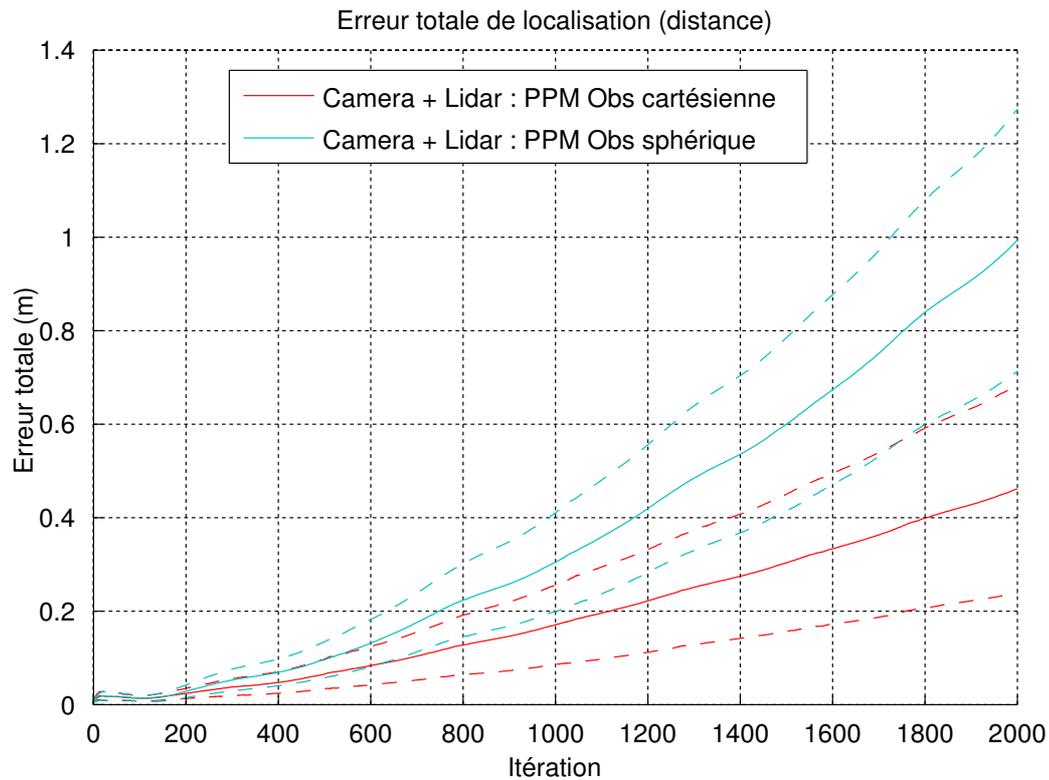


FIGURE 4.11 – Erreur totale moyenne des 100 réalisations suivant chaque méthode d’observation.

servation point par point. La figure 4.11 montre l’erreur totale obtenue suivant ces deux méthodes. On voit que la méthode d’observation "sphérique" propose une précision de localisation moins bonne que l’observation "cartésienne". Ce constat est le même en regardant les figures 4.12 et 4.13 qui correspondent aux erreurs de localisation et d’orientation sur chaque variable de l’état du véhicule. Les estimations des angles de l’orientation du véhicule et par conséquent la position du robot sont moins bien estimée en observation "sphérique".

4.2 Résultats en environnement simulé

4.2.1 Résultat de localisation

Dans la section précédente nous avons pu voir la faisabilité d’un système de fusion caméra-télemètre pour la localisation d’un véhicule. Maintenant il nous faut valider cette approche d’un point de vue plus expérimental avec toutes les contraintes que cela entraîne. Avant d’implanter notre approche sur un véhicule

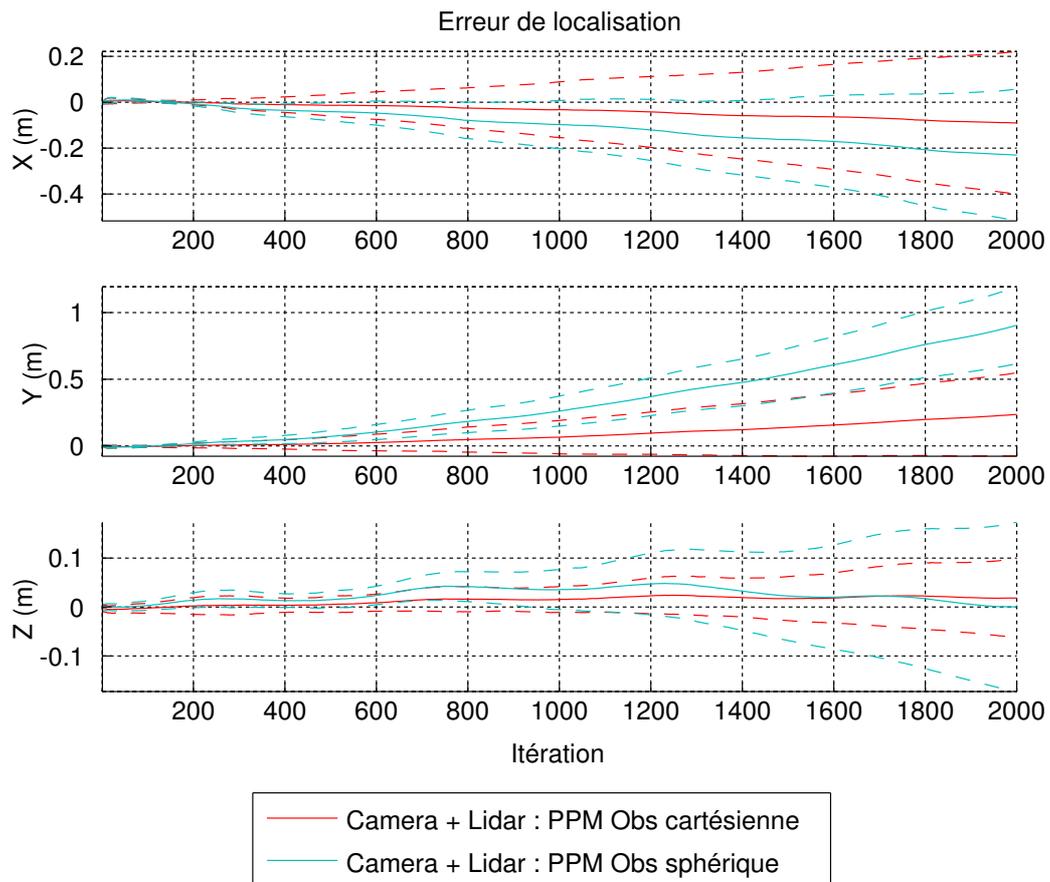


FIGURE 4.12 – Erreurs moyenne de localisation des 100 réalisations en X, Y et Z suivant chaque méthode d'observation.

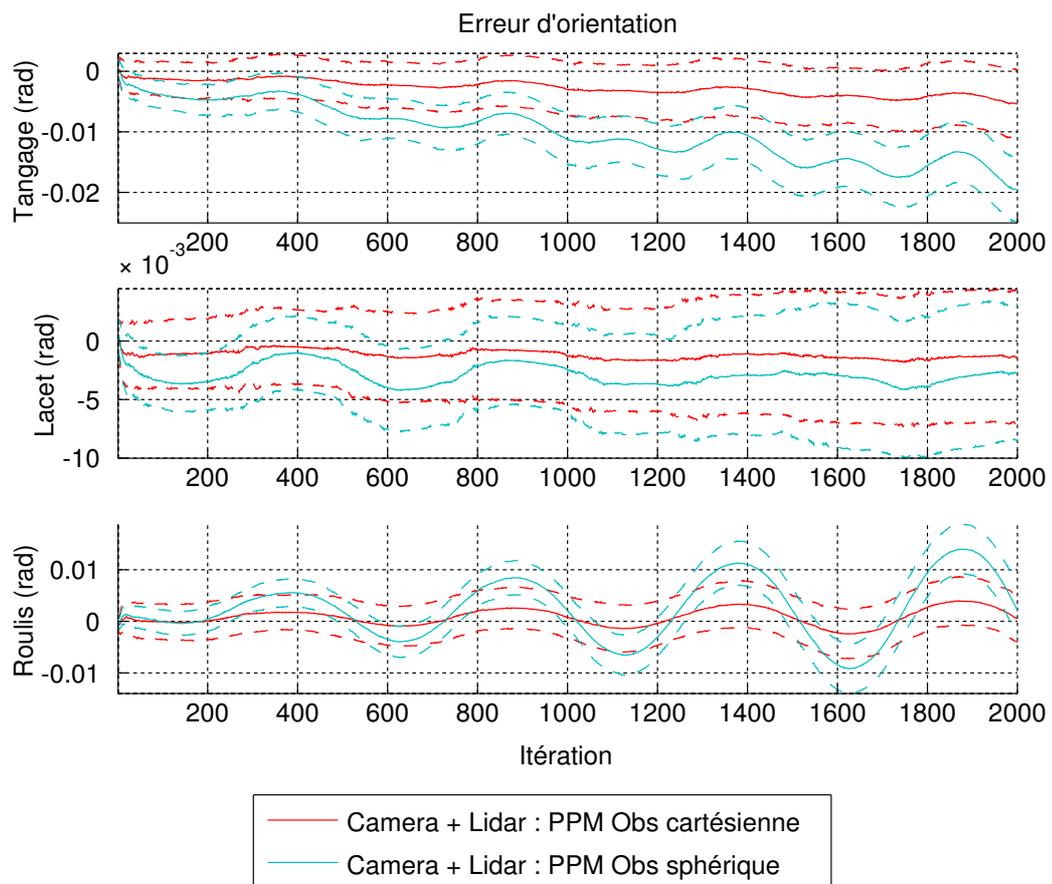


FIGURE 4.13 – Erreurs moyenne d'orientation des 100 réalisations en tangage, lacet et roulis (α , β et γ) suivant chaque méthode d'observation.

réel nous avons choisi de passer par une étape intermédiaire de simulation d'environnement. Lors de cette étape nous vérifierons les résultats obtenus en utilisant la localisation du véhicule par EKF. Nous comparerons les résultats obtenus en utilisant la méthode globale ou la méthode point par point suivant l'observation cartésienne uniquement. Pour cela nous avons testé notre algorithme dans un contexte maîtrisé que nous fournit notre simulateur **ASROCAM**.

4.2.1.1 Introduction du simulateur ASROCAM

ASROCAM² est un simulateur temps réel de capteurs et de véhicules dans un environnement virtuel. Il s'appuie sur un moteur de rendu visuel et un moteur physique. Les données générées par ce simulateur sont transmises au destinataire des données par protocole UDP/IP. La bonne gestion de la datation des données est assurée par le middleware AROCCAM. Toutes les données sont donc valides à leurs dates de perception. Du côté utilisateur, les données arrivent sous forme de trame contenant les données à traiter. L'avantage est de permettre d'utiliser la même application que ce soit avec des données réelles ou simulées. Seul le média de communication change entre les deux approches. Cependant ce changement est assuré par le système d'abonnement d'AROCCAM.

Dispositif expérimental sous simulateur Afin d'être proche de notre dispositif expérimental AROCO³, le modèle physique du véhicule a été modélisé à partir des données du véhicule réel. Sur le véhicule simulé, tous les capteurs nécessaires au déroulement du processus de perception ont été pris en compte. De plus des capteurs ont été ajoutés pour nous fournir une vérité terrain.

La caméra La caméra a été modélisée de façon générique avec les mêmes caractéristiques que celles que l'on pourrait trouver dans le commerce. Elle délivre une image en niveau de gris avec une résolution de 800x600 à 25 images/seconde avec un angle de vue de 45°. Aucune distorsion n'est appliquée à l'image.

Le télémètre Le télémètre aussi a été modélisé de façon générique. Les données qu'il procure sont cadencées à 12.5 Hz. La plage d'acquisition s'étend sur 180° avec une résolution d'une mesure par degré. Afin de mieux correspondre à un système réel, les données acquises sont arrondies au centimètre près.

Le véhicule et ses capteurs Le véhicule en lui-même est instrumenté de la même façon que le robot réel. Sur chaque roue se trouve un capteur d'odométrie fournis-

2. Architecture de Simulation de **RO**bots et **CA**pteurs **MO**dulaire

3. Véhicule expérimental de la FR TIMS. Dimensions (1.4 x 2.2), poids (600kg).

sant une donnée cadencée à 10 Hz. De plus, chaque train directeur (avant et arrière) dispose d'un capteur d'angle de direction. La dynamique du véhicule est assurée par le moteur physique ODE⁴ qui intègre masse, suspension et liaisons des différents éléments du véhicule.

La vérité terrain Afin de vérifier les performances de notre système, le véhicule est instrumenté par des capteurs fournissant la vérité terrain du véhicule et de son environnement. La localisation est assurée par un capteur qui fournit les 3 paramètres de positionnement du véhicule ainsi que les 3 paramètres d'orientation.

Calibration du système caméra-télémetre Afin de tester le système dans le simulateur il nous faut calibrer la caméra et le télémetre pour être dans les mêmes conditions que le système réel. Pour cela nous avons utilisé la méthode décrite en annexe A.2. Pour comparer les résultats de calibration, le simulateur peut nous fournir directement la vraie matrice de calibration $\mathbf{M}_{(t-c)(true)}$. Les résultats de calibration sont montrés en équation 4.2.

$$\mathbf{M}_{(t-c)} = \begin{pmatrix} 0.9999927 & 0.0037263 & -0.0007748 & 0.0003115 \\ -0.0038029 & 0.9864540 & -0.1639935 & 0.1755090 \\ 0.0001532 & 0.1639952 & 0.9864611 & 0.3309977 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2)$$

$$\mathbf{M}_{(t-c)(true)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9865754 & -0.1633057 & 0.1731318 \\ 0 & 0.1633057 & 0.9865754 & 0.3256153 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.3)$$

Soit une erreur en position de $(-0.0003115 \quad -0.0023771 \quad -0.0053823)^T$ (exprimée en mètres) et une erreur en orientation de $(-0.0006972 \quad 0.0007748 \quad 0.0037263)^T$ (exprimée en radians). Ces erreurs de calibration ont un impact faible sur le positionnement des points du télémetre dans l'image. Dans le cas de nos expérimentations, le faisceau laser du télémetre touche le sol environ 3 mètres devant lui. En comparant la calibration calculée avec la vraie nous obtenons une moyenne des erreurs de position 3D des points provenant du télémetre de 6.4 mm ainsi qu'une erreur de reprojection moyenne de ces points dans l'image de 0.944 pixels.

Modèle d'évolution du véhicule Le filtre de Kalman a besoin d'une prédiction de l'évolution de son état pour fonctionner. Cette étape permet "d'orienter" la valeur initiale du vecteur d'état avant d'utiliser les observations pour mettre à jour

4. ODE : Open Dynamics Engine <http://www.ode.org/>

le filtre. L'évolution la plus triviale que peut faire un système est le *random-walk* où tous les paramètres sont décorrélés et aléatoires. Cependant notre système ne peut pas évoluer de la sorte. Notre véhicule est non-holonome puisque il possède moins de capacités de commande que de degrés de liberté. Son évolution peut être prédite en utilisant un modèle (modèle d'Ackermann utilisé par exemple dans [Thuilot *et al.*, 2004]). Aussi nous utilisons un modèle de prédiction dont les équations sont données en 2.38 dans la section 2.7.2.1.

La covariance aussi doit être prédite en utilisant ce modèle d'évolution, pour cela il nous faut utiliser les jacobiniennes de cette fonction par rapport à l'état et par rapport à la commande.

$$\begin{aligned} \mathbf{F}_1 &= \frac{\partial f}{\partial \vec{X}_{k-1}} \\ \mathbf{F}_2 &= \frac{\partial f}{\partial odo} \\ \mathbf{F}_3 &= \frac{\partial f}{\partial \delta} \end{aligned} \quad (4.4)$$

En supposant que les mesures de l'odométrie et de l'angle de direction soient entachées d'un bruit blanc Gaussien nous obtenons alors :

$$\mathbf{P}_{\vec{X}_{k|k-1}} = \mathbf{F}_1 \cdot \mathbf{P}_{\vec{X}_{k-1|k-1}} \cdot \mathbf{F}_1^T + \mathbf{F}_2 \cdot \sigma_{odo}^2 \cdot \mathbf{F}_2^T + \mathbf{F}_3 \cdot \sigma_{\delta}^2 \cdot \mathbf{F}_3^T \quad (4.5)$$

où $\mathbf{P}_{\vec{X}}$ est la matrice de covariance associée à l'état du véhicule, σ_{odo} est l'écart type de l'incertitude associée à la mesure de l'odométrie et σ_{δ} est l'écart type de l'incertitude associée à la mesure de l'angle de direction.

4.2.1.2 Résultats sans correction de la dérive

Dans cette section nous allons présenter les résultats de notre système en terme de localisation du véhicule dans un environnement simulé. Le scénario que nous allons utiliser est une portion de circuit sur lequel nous avons placé des boîtes de taille $0.5 \times 0.5 \times 0.2m$. La trajectoire réalisée effectue un slalom entre ces boîtes. Un aperçu du scénario est montré en figure 4.14.

Méthode Globale Dans un premier temps, nous allons vérifier les performances de notre algorithme lorsque toutes les correspondances de points sont utilisées en même temps (méthode globale : GM). Nous vérifierons aussi que l'incertitude artificielle proposée en section 2.7.2.1 rend bien intègre notre estimation de la position et de l'orientation du véhicule. Dans ce scénario, des points du télémètre sont ajoutés au vecteur d'état à raison d'un point sur six tous les 10 cm parcourus environ. Tout d'abord, l'erreur totale pour chaque type d'observation est montrée en figure



FIGURE 4.14 – Vue externe du scénario présenté sur le simulateur.

4.15. La vue de dessus du parcours effectué avec chacune des méthodes est montré en figure 4.16. Nous pouvons voir que l'erreur commise dérive très peu avec la distance parcourue ($\approx 9\%$ de dérive). Plus particulièrement les erreurs de positionnement sur chaque composante ainsi que les erreurs d'orientation sont montrées en figure 4.17 et 4.18. Les incertitudes à un écart type de la pose du véhicule estimée par l'EKF et l'incertitude artificielle sont montrées respectivement par des pointillés fins et des pointillés larges. Il est important de noter que les erreurs de localisation sont principalement dues à ces erreurs d'orientations qui s'accumulent tout au long de la trajectoire. On peut aussi remarquer que nous obtenons des résultats largement inconsistants. Cependant l'incertitude artificielle remplit bien son rôle.

A ce stade, notre approche de localisation 6D par fusion de donnée caméra et télémètre fonctionne. Les résultats obtenus sont différents des prévisions faites en calcul théorique pur sous MATLAB à cause de la présence de glissement des patches utilisés pour effectuer la mise en correspondance des points dans la caméra. Une réflexion sur ce phénomène sera présentée dans le bilan de cette section.

Méthode Point par Point Nous avons vérifié le fonctionnement de notre algorithme de fusion de données. Nous pouvons maintenant tester et vérifier la précision de notre système lorsque nous utilisons les observations les unes après les autres. Pour cela nous utilisons le même jeu de données que précédemment. Les résultats sur l'erreur de positionnement sont montrés en figure 4.19. Dans un premier temps nous pouvons remarquer que l'estimation de la pose du véhicule en utilisant une méthode point par point est au moins aussi précise qu'avec une méthode globale (8 mètres d'erreur contre 9 pour la méthode globale). Les résultats de cette méthode

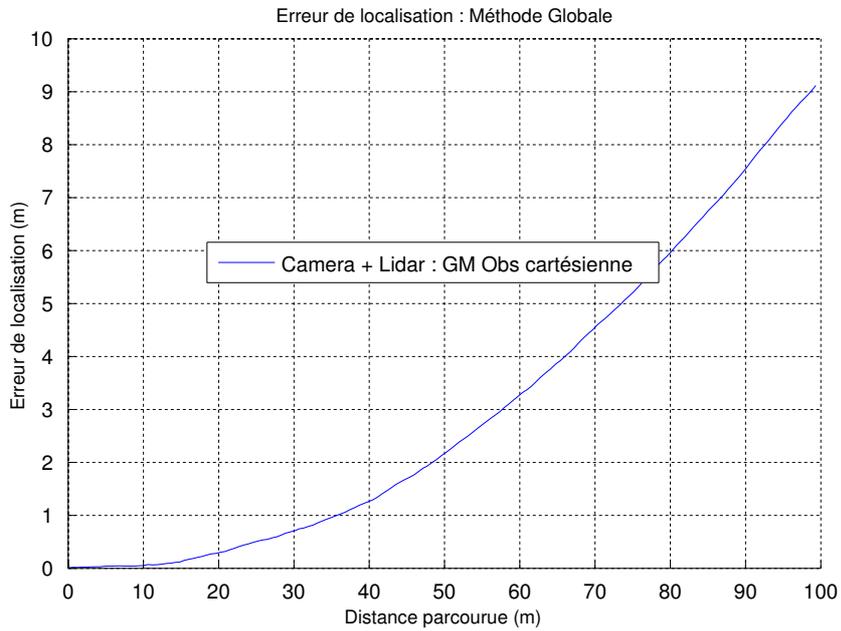


FIGURE 4.15 – Erreur totale de localisation en utilisant la méthode globale.

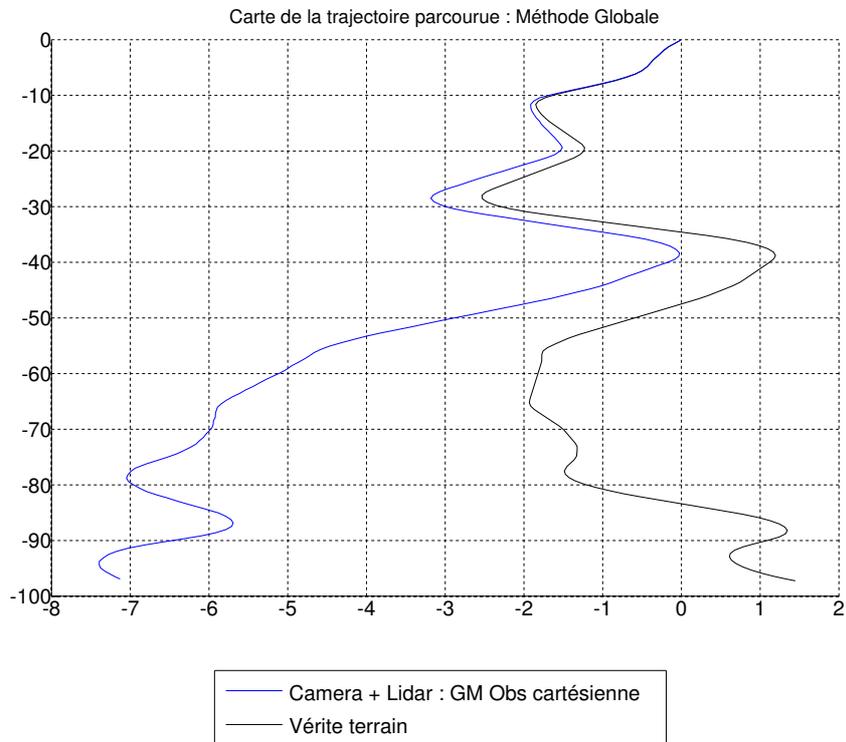


FIGURE 4.16 – Vue du dessus de la trajectoire calculée en méthode globale.

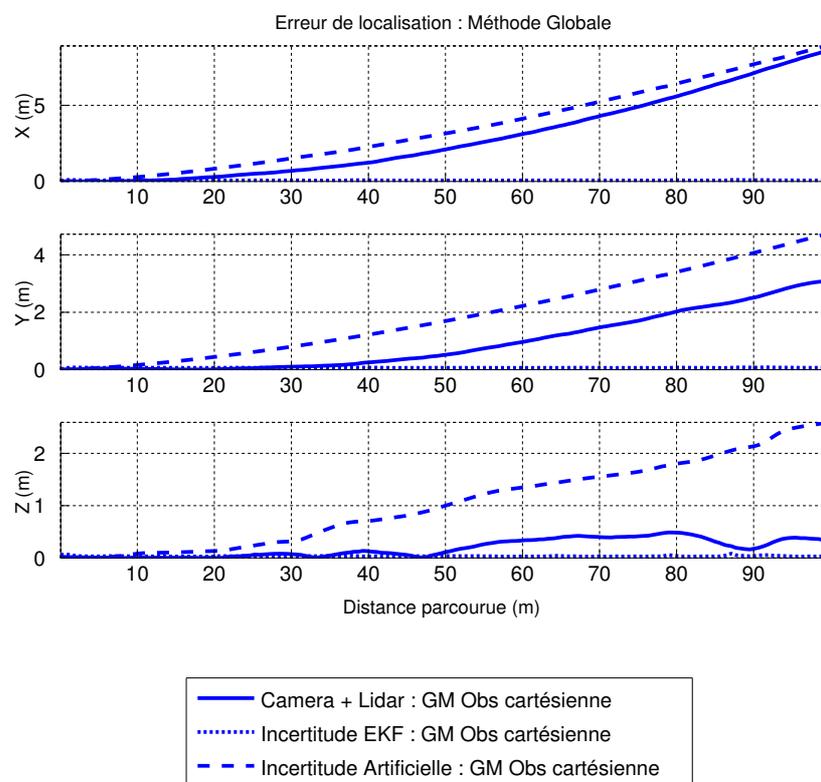


FIGURE 4.17 – Erreur de localisation suivant chaque variable en méthode globale.

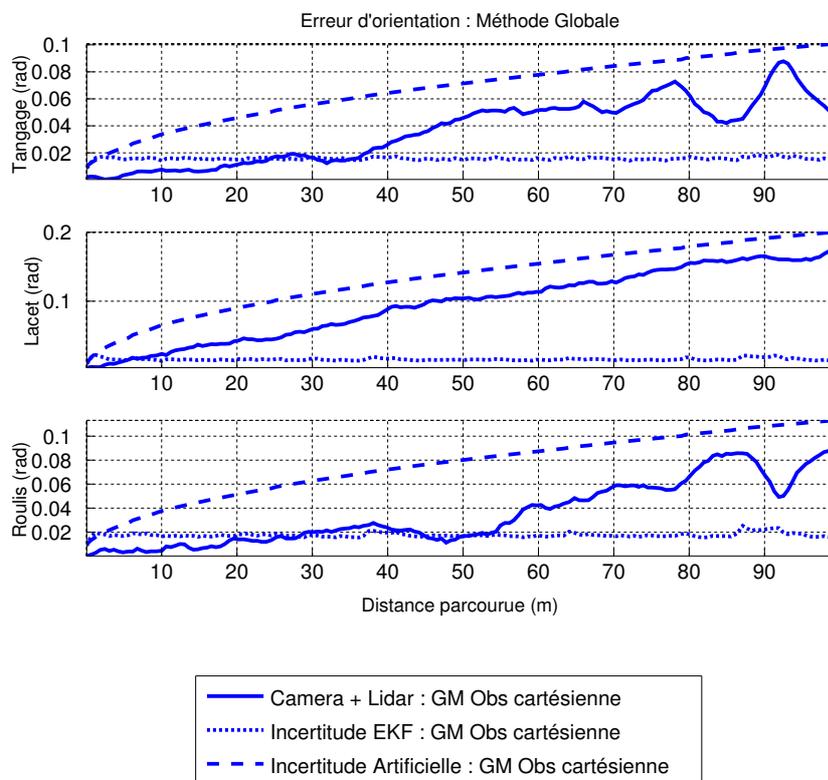


FIGURE 4.18 – Erreur d'orientation suivant chaque variable en méthode globale.

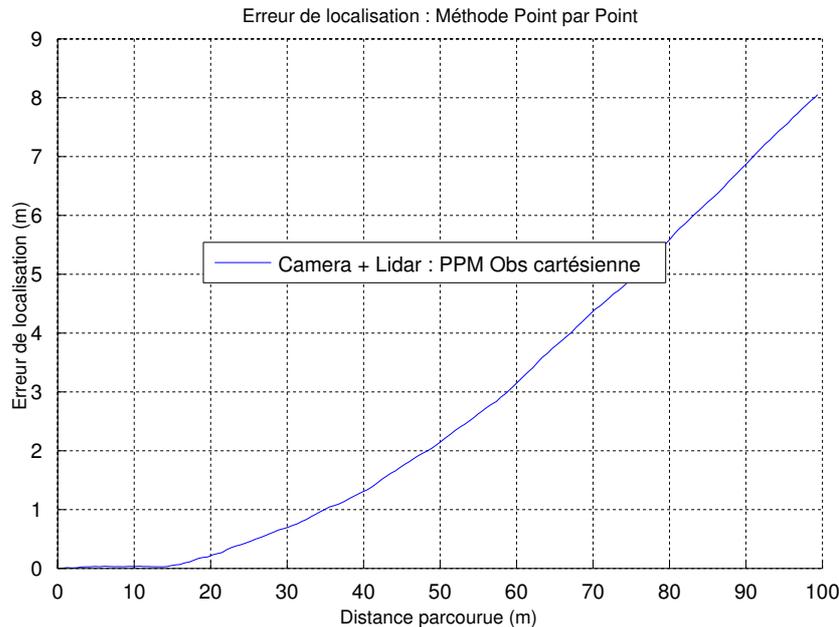


FIGURE 4.19 – Erreur totale de localisation en utilisant la méthode point par point.

sont inconsistants tout comme pour la méthode globale. Cependant l'incertitude artificielle permet de garder une estimation de localisation intègre comme on peut le voir dans les figures 4.21 et 4.22 où les erreurs d'estimations sont bien comprises dans l'incertitude.

Dans cette section nous avons vu la faisabilité de notre système de localisation par fusion de données. De plus nous avons montré la qualité des résultats lorsque les observations sont considérées les unes-après les autres (méthode point par point).

4.2.1.3 Résultats avec correction de la dérive

Ici, nous vérifions le fonctionnement du système lorsqu'une mesure GPS est utilisée pour corriger le calcul de pose. Pour cela, nous allons utiliser notre système de localisation par fusion de données avec l'observation cartésienne et en utilisant la méthode point par point. A ce système est ajouté la méthode d'estimation du biais de localisation proposée en section 2.7.2.3. La procédure de test se fera dans un premier temps sur des données simulées puis sur des données réelles.

Pour ce dispositif de test sous simulateur, les observations de la pose du véhicule par GPS sont faites toutes les 2 secondes. Pour cela nous utilisons les données de positionnement du robot fournies par le simulateur. Les bruits d'observation de la mesure sont supposés Gaussiens de moyenne nulle et d'écart type 5cm pour la position et 0.05rad pour le cap. Ces valeurs ont été choisies pour se rapprocher des erreurs habituelles d'un GPS-RTK.

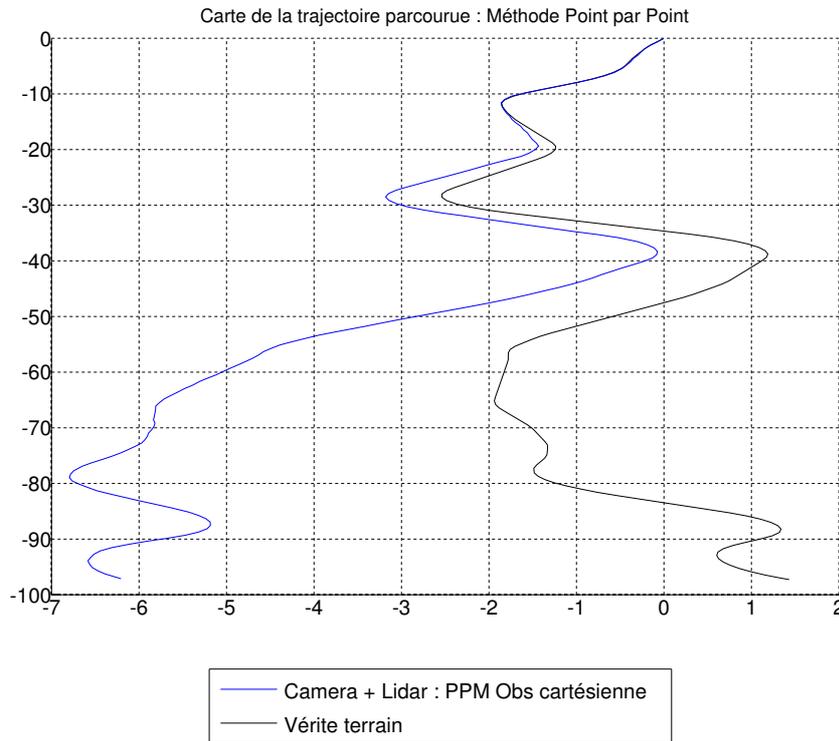


FIGURE 4.20 – Vue du dessus de la trajectoire calculée en méthode point par point.

Le jeu de données est exactement le même que celui utilisé en section 4.2.1. Les résultats peuvent être comparés directement. La figure 4.23 montre une vue de dessus de la trajectoire parcourue après correction de la dérive par une mesure GPS. On peut voir que cette trajectoire est bien corrigée par rapport au plan Oxz . En effet l'erreur reste inférieure à $5cm$ sur l'ensemble de la trajectoire comme on peut le voir en figure 4.24. Naturellement, l'erreur de positionnement sur l'axe y n'est pas corrigé puisque aucune information d'altitude n'est utilisée dans le calcul de biais. Dans les figures 4.24 et 4.25 on peut voir l'effet de la mise à jour sur l'incertitude artificielle qui réagit comme attendu en diminuant en fonction de la précision de la mesure.

4.2.1.4 Bilan intermédiaire

Nous venons de voir les résultats obtenus avec notre algorithme de localisation par fusion caméra télémètre en environnement simulé. Si on compare ces résultats avec ceux obtenus en section 4.1 on s'aperçoit qu'il y a des différences en terme de précision. La différence de nos algorithmes d'odométrie visuelle entre la version théorique sous MATLAB et la version plus pratique sur notre simulateur est la nécessité de mettre en correspondance des points dans un flux d'images. De plus les

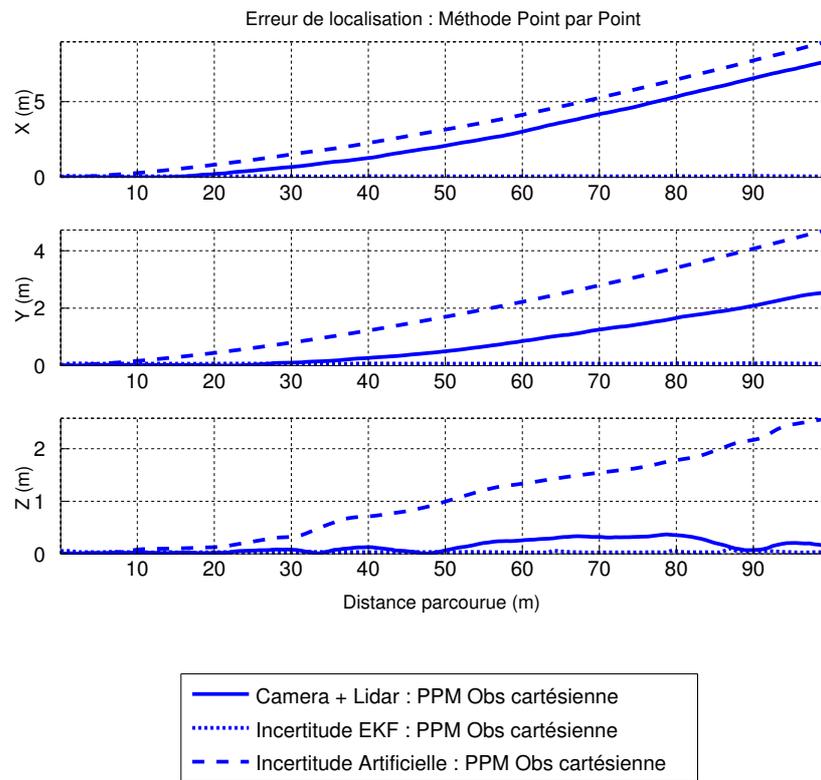


FIGURE 4.21 – Erreur de localisation suivant chaque variable en méthode point par point.

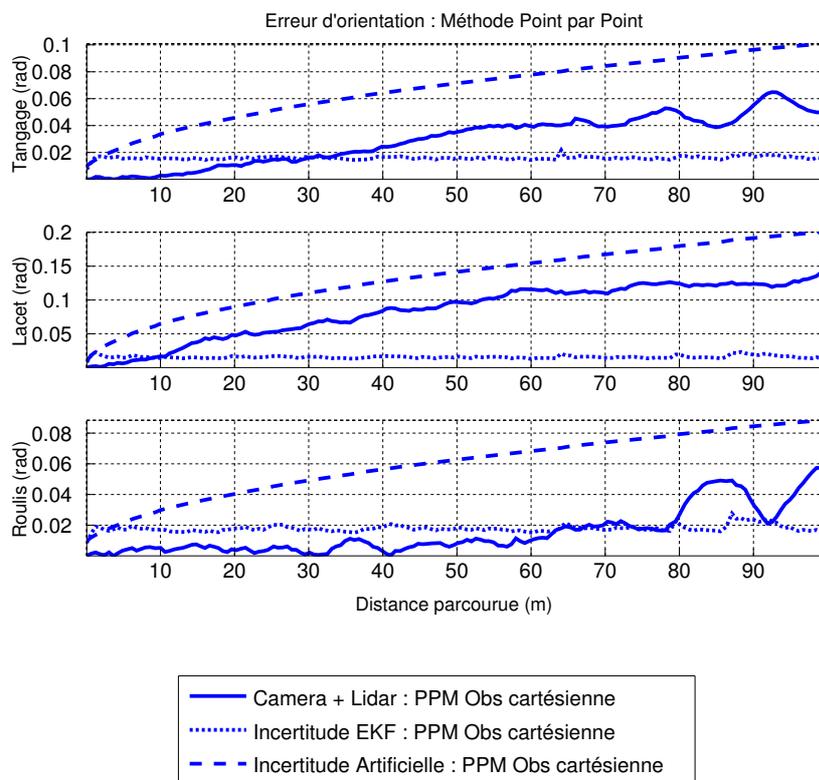


FIGURE 4.22 – Erreur d'orientation suivant chaque variable en méthode point par point.

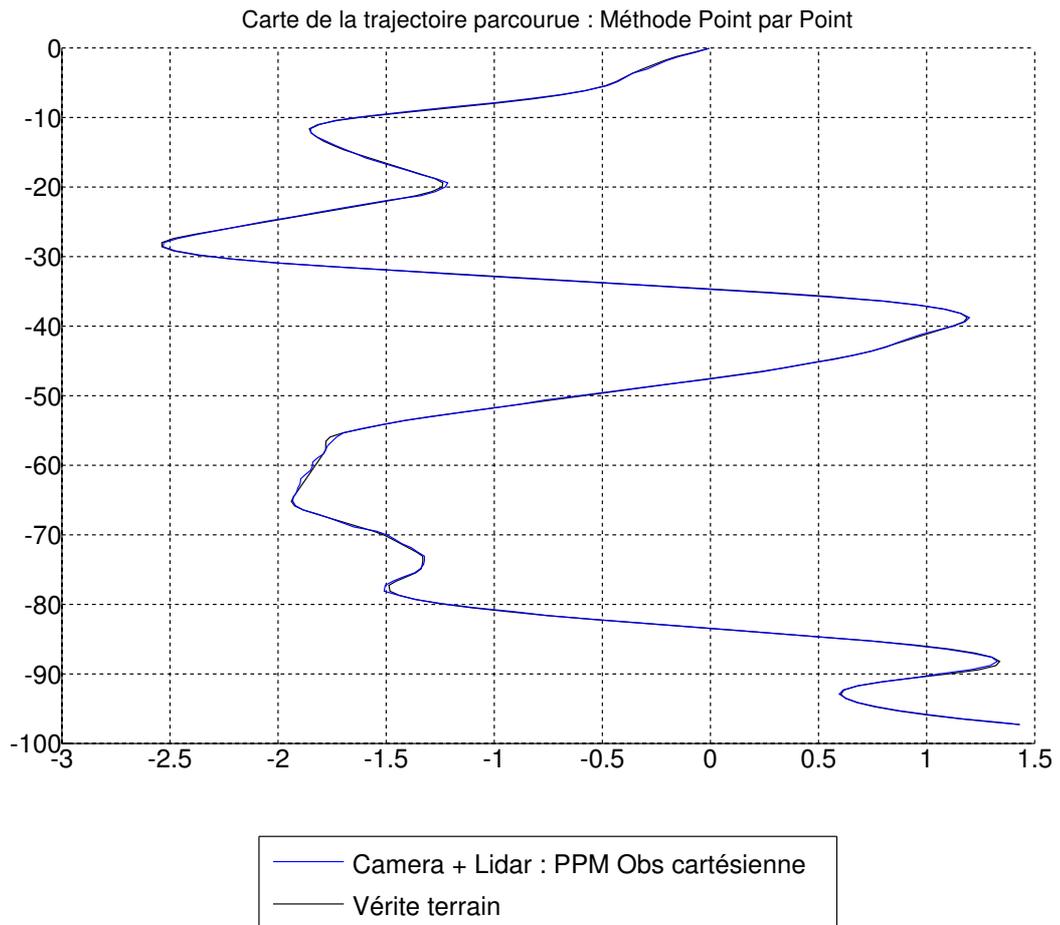


FIGURE 4.23 – Vue du dessus de la trajectoire parcourue en environnement simulé après correction de la dérive.

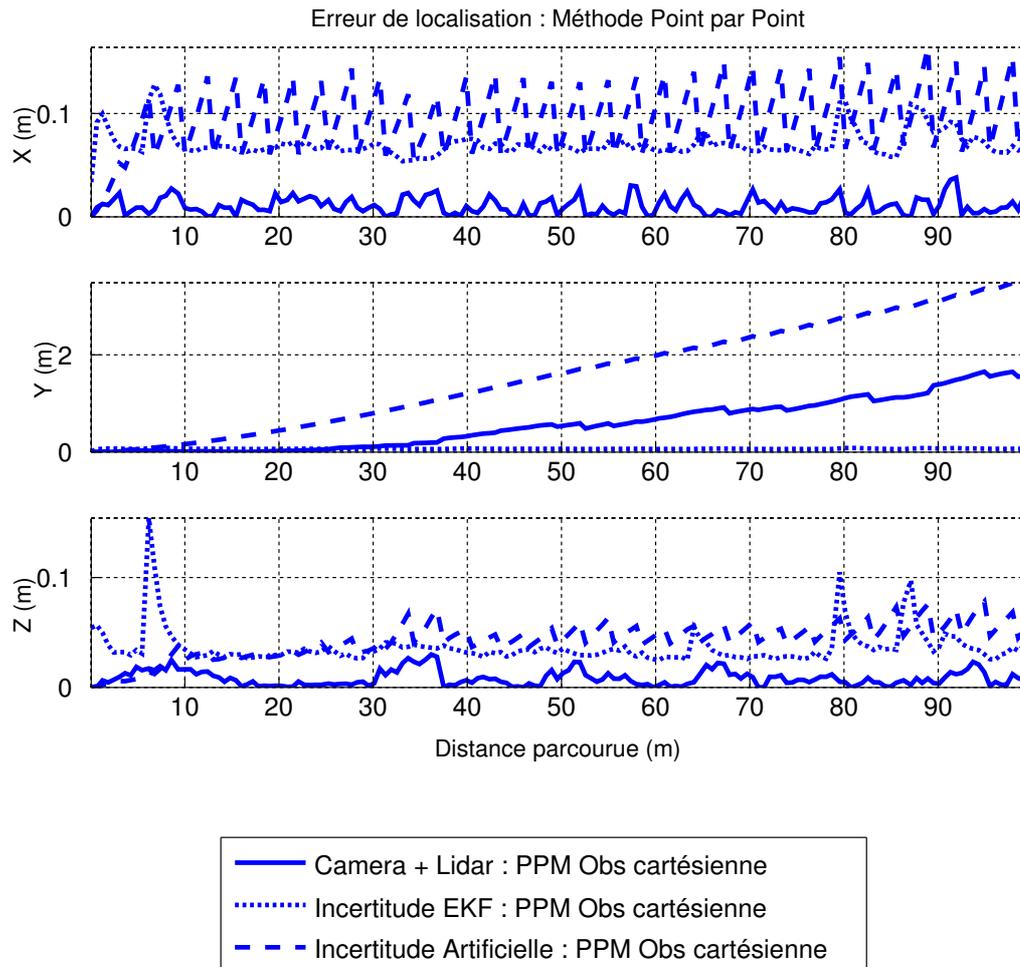


FIGURE 4.24 – Erreur de localisation suivant chaque variable après correction de la dérive.

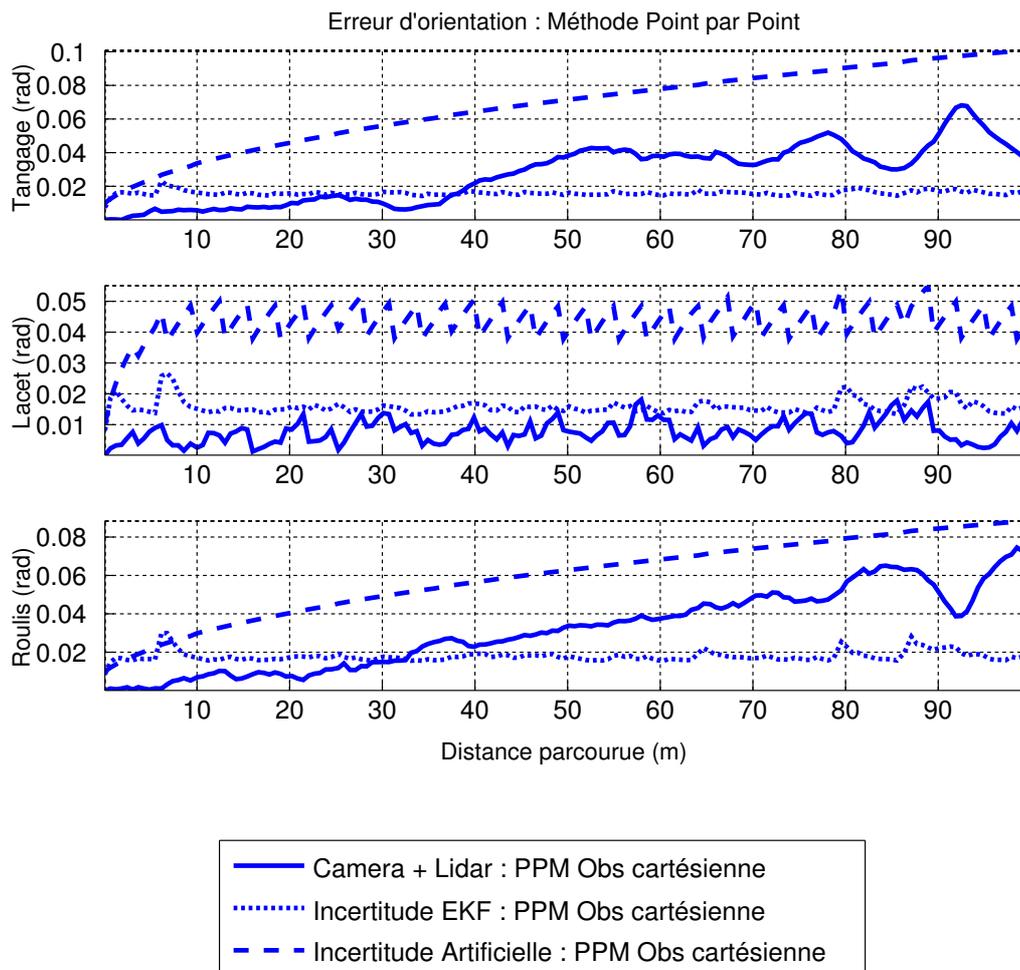


FIGURE 4.25 – Erreur d'orientation suivant chaque variable après correction de la dérive.

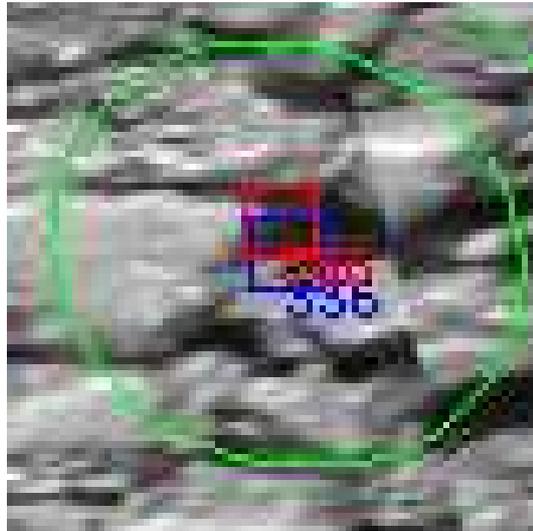


FIGURE 4.26 – Le phénomène de glissement pour l'appariement des points d'une image à l'autre. En bleu : la projection exacte de l'amer du vecteur d'état. En rouge : l'appariement effectué

points utilisés comme amers sont positionnés sur le plan de la route (à une hauteur de 0 mètre) ce qui est connu pour être un mauvais conditionnement en géométrie épipolaire par exemple. Or si on observe le comportement du système en environnement simulé on peut remarquer des phénomènes de glissement de la mise en correspondance des points. La figure 4.26 montre un exemple de ces glissements.

Dans cette figure, tous les paramètres du système ont été remplacés par leurs vraies valeurs (calibration du système caméra-télémètre et pose estimée du véhicule) fournies par le simulateur. Dans ce cas là notre système cherche donc à appairier exactement les vraies projections des points 3D. On s'aperçoit alors que les mises en correspondance successives ont amené une différence entre la projection réelle de l'amer (en bleu) et l'appariement trouvé (en rouge). Les différences entre les vraies projections des amers et leurs appariements dans le jeu de données simulé sont montrées en figure 4.27. On voit que le glissement peut dépasser aisément plus de 10 pixels sur les axes u ou v de la caméra. Il est important de noter qu'il ne s'agit pas de faux appariements ou des conséquences des erreurs de calibration mais bien de glissement de la mise en correspondance. Ces erreurs sont très pénalisantes puisque qu'elles provoquent une dérive du système de localisation que ce soit par la méthode globale ou point par point.

La connaissance initiale précise des amers contenus dans le vecteur d'état est alors à la fois un atout et un handicap. En effet, le fait d'avoir cette information précise amène une grande confiance de l'amer dans le filtre de Kalman et donc une covariance de l'observation associée très faible. Ceci va avoir une conséquence sur

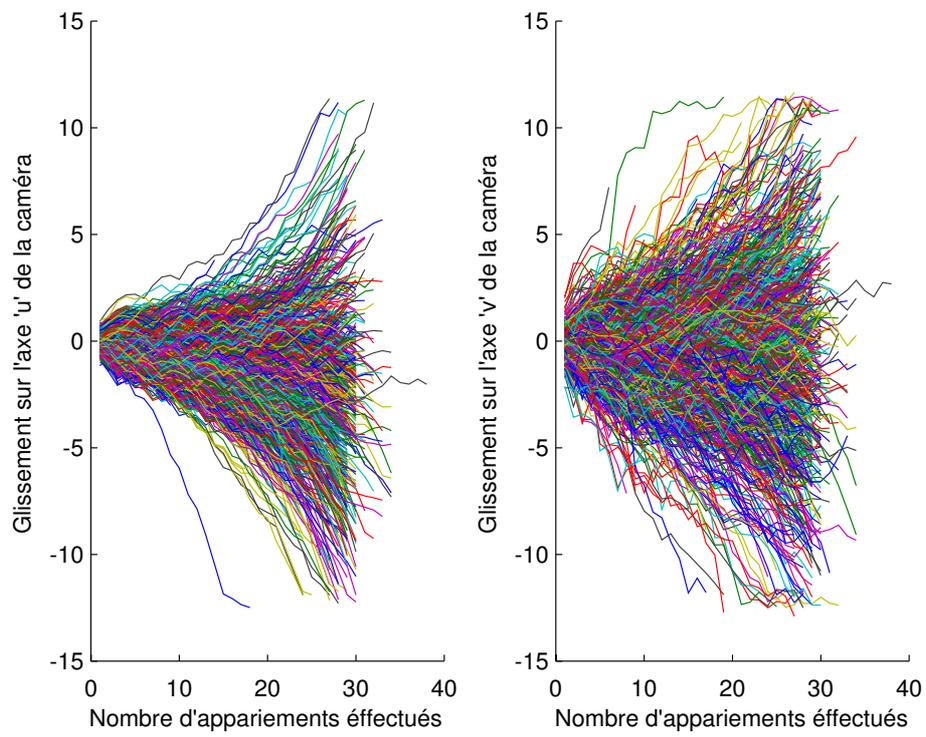


FIGURE 4.27 – Glissements en u et v obtenus par rapport aux vraies projections des amers

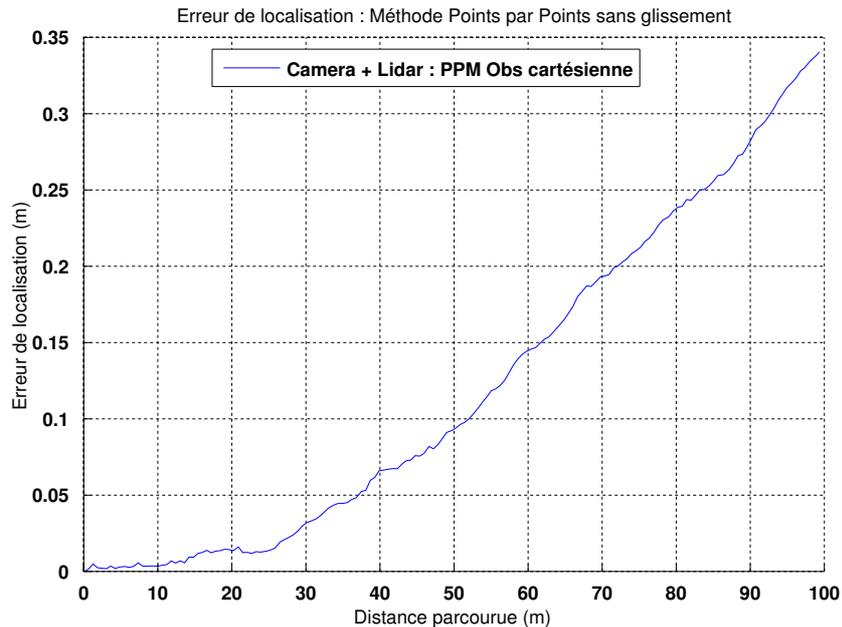


FIGURE 4.28 – Erreur totale de localisation en utilisant la méthode point par point en absence de glissement.

le gain de Kalman qui sera plus important avec des points initialisés avec cette grande précision plutôt que d'autres initialisés plus classiquement et donc de façon moins précise. De ce fait, le glissement sera plus ressenti avec des points provenant du télémètre qu'avec des points initialisés de façon ordinaire.

Afin de vérifier que les erreurs de notre système de localisation viennent bien de ce glissement nous avons supprimé artificiellement ce problème. Pour cela nous avons utilisé les informations vraies de la pose du véhicule provenant du simulateur pour fournir les observations de nos amers dans la caméra. Cette configuration permet de s'affranchir des glissements en évitant la mise en appariement des amers dans la caméra. La figure 4.28 montre l'erreur de localisation obtenue sans glissement.

Plusieurs explications peuvent justifier ce glissement :

- La première est que les patches se déforment beaucoup entre plusieurs itérations. Les patches sont les textures récupérées de l'image et qui correspondent à un point de l'environnement de normale différente de la prise de vue de la caméra. Ces patches subissent donc une homographie entre plusieurs itérations.
- La deuxième explication vient de la façon de traiter la mise en correspondance dans notre approche. Il y a deux façons de faire pour la mise en œuvre de l'appariement.

1. Premièrement, on peut initialiser le patch avec la première représen-

tation du point trouvé puis garder ce patch durant tout le processus de mise en appariement. Cette technique ne permet pas de mettre en correspondance les amers pendant une longue séquence d'images (quelques itérations au plus) car le phénomène de déformation de patch cité précédemment empêche une bonne mise en correspondance.

2. Deuxièmement, on peut mettre à jour le patch correctement apparié avec la texture de l'amer dans la nouvelle image. Cette technique permet de mettre en correspondance l'amer pendant plus longtemps puisque d'une itération à l'autre le patch se déforme très peu. Cependant comme la position du patch dans la nouvelle image est une position arrondie au pixel de la projection de l'amer, ces simplifications successives peuvent provoquer un glissement important du patch.

Afin de contrer ce phénomène, nous avons mis en place un bruit de mesure dans la caméra qui dépend de chaque amer et qui augmente avec les itérations du système. Les résultats montrés dans ce chapitre tiennent compte de cette modification. Malgré tout, ce problème peut tout à fait être contenu par l'utilisation de l'incertitude artificielle afin de garder une pose du véhicule intègre. De plus, ces glissements font seulement dériver le système de localisation à long terme mais ne remettent pas en cause ni l'utilisation du télémètre comme source d'informations supplémentaires ni l'aptitude du système de perception à reconstruire un MNT local dense et précis.

4.2.2 Résultat de construction du MNT global

Dans cette partie nous allons chercher à estimer le MNT obtenu lorsque l'on garde tous les points qui proviennent du système de fusion de données. Dans un premier temps, la figure 4.29 représente la carte d'élévation lorsque l'on utilise une localisation parfaite avec les données télémètre. Dans un deuxième temps, la figure 4.30 montre les résultats obtenus en utilisant la localisation faite par notre système de fusion de données en méthode point par point. On peut y voir une carte qui a tendance à descendre. Ce phénomène est présent à cause de la dérive du processus de localisation qui entraîne la même dérive sur le MNT. Cependant on peut constater qu'il n'y a pas de cassure dans la carte qui pourrait venir d'une estimation d'orientation aberrante par exemple.

Dans notre application, l'information importante qui va nous permettre de déterminer si l'environnement est navigable ou pas est contenue dans la cartographie locale par rapport au véhicule. Même si la carte globale est différente de l'environnement réel à cause de la dérive du système de localisation, les informations locales qu'elle contient sont très précises et ne comportent pas de dérive comme nous le verrons dans le paragraphe suivant.

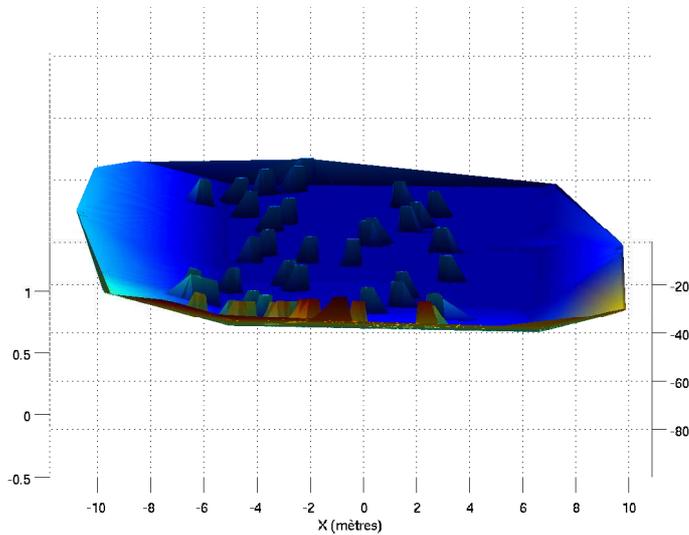


FIGURE 4.29 – Modèle numérique de terrain obtenu à partir des points télémétriques en utilisant la vraie pose du robot. La hauteur de l'environnement est codée en fonction de la couleur, du bleu (0cm) au rouge (80cm).

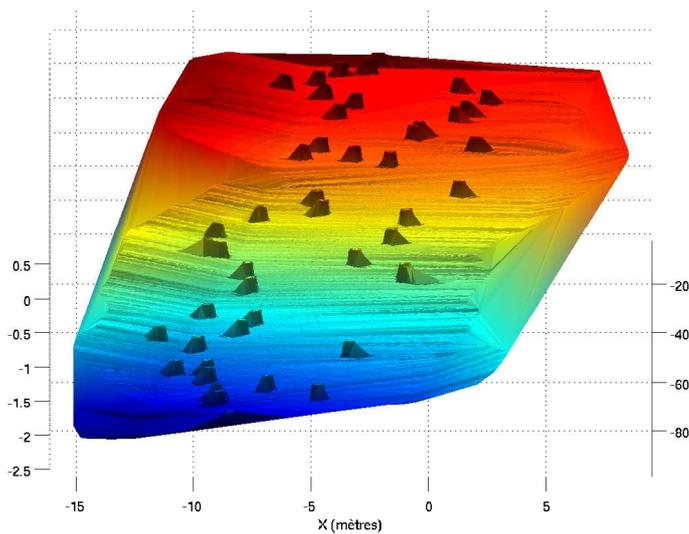


FIGURE 4.30 – Modèle numérique de terrain obtenu à partir des points télémétriques en utilisant l'observation cartésienne en méthode point par point. La hauteur de l'environnement est codée en fonction de la couleur, du bleu (-2m) au rouge (0m).

4.2.3 Résultat de construction du MNT local

Dans la section précédente nous avons vu que le MNT global reconstruit a subi un effet indésirable à cause de la dérive du processus de localisation. Ce qui nous intéresse pour mettre en œuvre le processus de guidage au complet est d'extraire de cette carte globale les informations locales et pertinentes de cette carte. Ces informations locales ne subissent que très peu ce phénomène de dérive et donc le MNT devrait être très satisfaisant.

Avant de parler des résultats de reconstruction du MNT local à proprement parlé, intéressons-nous aux besoins exprimés par le système de navigation pour achever sa tâche. Lorsque nous avons évoqué la reconstruction de terrain par système SLAM au chapitre 3 nous avons dit que la puissance de calcul des ordinateurs actuels n'était pas suffisante pour avoir une densité en rapport avec les besoins de notre système de guidage. Concrètement, avec notre véhicule expérimental AROCO dont les roues ont un diamètre de $\phi = 80\text{cm}$ il est admis de dire que le sol deviendra infranchissable à partir $\Delta h = \phi/3$. La carte doit donc présenter au minimum un point 3D tous les 20cm environ pour être suffisamment dense. Il faut au véhicule une distance de 12.5m pour s'arrêter pour une vitesse de 5m/s en ayant une décélération de 1m/s^2 . En ayant une zone d'observation de deux fois la largeur du véhicule ($2 \times 1.5 = 3\text{m}$) la carte doit contenir au minimum $\frac{12.5}{0.2} \cdot \frac{3}{0.2} = 938$ points. Cette réflexion sur le nombre de points nécessaire au système de guidage avait orienté nos recherches vers une autre source principale d'informations pour construire le modèle numérique de terrain. Dans la suite de cette partie, nous évaluerons la capacité de notre système à créer une carte d'élévation avec la fusion caméra-télémetre.

Dans cette partie nous allons montrer les résultats obtenus pour la reconstruction du MNT dans le repère local du véhicule en utilisant des données provenant du simulateur ASROCAM. Dans un premier temps, nous montrerons les résultats de reconstruction lorsque seul le télémetre est utilisé. Le calcul de pose sert alors à repositionner chaque nappe télémetre dans le repère local du véhicule.

Dans un deuxième temps nous montrerons la reconstruction du terrain lorsque les nappes télémetres sont insuffisantes. Dans ce cas, les informations supplémentaires sont estimées par le processus d'odométrie visuelle en ne focalisant que sur la trajectoire à suivre.

4.2.3.1 Reconstruction par Lidar seul

Ici nous allons montrer les résultats obtenus lorsque seul le télémetre est utilisé pour reconstruire le MNT. Pour cela nous ajoutons dans notre carte d'élévation les mesures télémétriques lorsque une certaine distance a été parcourue. Cette action a pour but d'éviter de surcharger la carte lorsque le véhicule est à l'arrêt. Les don-

nées de la carte ont été repositionnées dans le repère local du véhicule pour être affichées. Les figures 4.31 et 4.32 montrent la carte obtenue pour certaines itérations du processus. On peut voir sur cette figure que cette carte produite est dense et précise grâce à la bonne localisation du robot localement.

Grâce à l'environnement simulé, nous pouvons comparer le MNT local obtenu par notre système de perception avec le vrai MNT fabriqué avec la vraie pose du véhicule fournie par le simulateur. Le résultat de cette comparaison entre carte estimée et réelle est montré en figure 4.33. On voit sur cette figure que la dérive de localisation de notre système ne fait pas dériver le MNT local. De plus, grâce aux informations lidar, le MNT reconstruit est très précis (erreur moyenne des points reconstruits inférieure à 10 centimètres sur toute la séquence). La figure 4.34 montre l'histogramme des erreurs obtenu sur la dernière itération du processus.

4.2.3.2 Reconstruction densifiée par la caméra

Il peut arriver lors d'une navigation que le télémètre ne produise pas de données à certains endroits pertinents de l'environnement. Par exemple lorsque le véhicule franchit un obstacle, il se peut qu'une partie de l'environnement soit invisible du télémètre ou si la fréquence d'échantillonnage du capteur devient trop lente vis-à-vis de la vitesse du véhicule. Si ce scénario arrive, le processus de guidage ne peut réaliser sa tâche de navigation autonome. Dans ce cas là, il est possible pour le système d'avoir recours à une demande explicite d'estimation d'amers au sein de la carte d'élévation. Ces amers, même s'ils ne sont pas visibles dans l'espace du télémètre, peuvent être accessibles dans l'image caméra. C'est donc ce capteur qui prend le relai pour enrichir la carte.

Nous allons montrer les résultats obtenus lorsque le télémètre est complété par la caméra pour reconstruire un MNT dense. Pour cela nous allons limiter artificiellement la prise en compte des données télémétriques dans la carte et nous allons la compléter avec autant de points caméra que le permet notre système. La figure 4.35 montre la carte locale obtenue pour une itération du processus lorsque peu d'informations télémétriques sont disponibles. La figure 4.36 montre la carte locale obtenue pour la même itération lorsque l'on augmente la quantité d'informations avec des données de la caméra. Dans ce cas de figure on voit que le processus d'odométrie visuelle est tout à fait capable de combler les espaces inobservés par le télémètre qui sont pertinents pour le processus de navigation.

4.2.4 Bilan

Dans cette section, nous avons confronté notre système de perception multi-sensorielle à un environnement simulé réaliste. Cette procédure a permis de valider notre approche de fusion de capteurs caméra et télémètre pour déterminer la pose

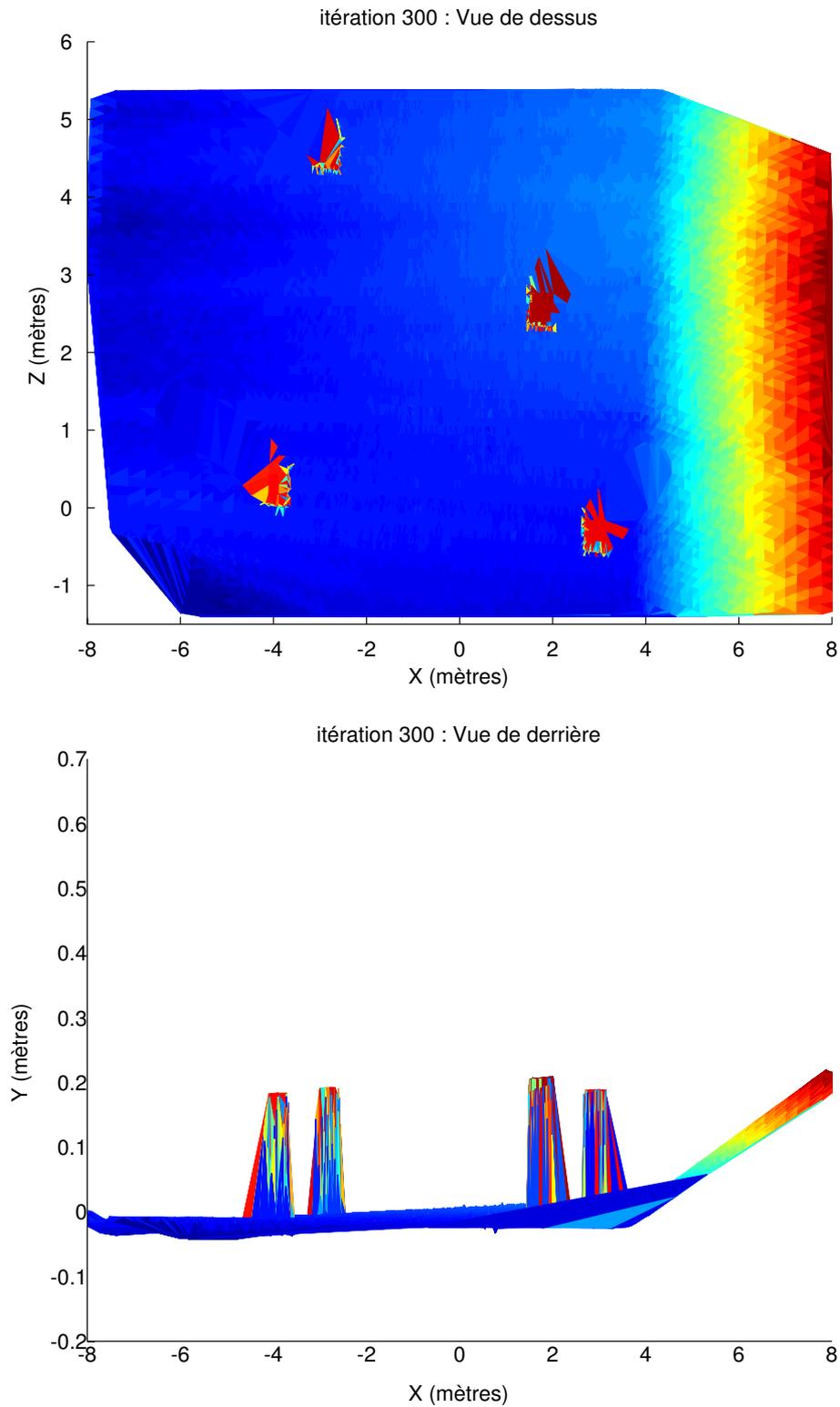


FIGURE 4.31 – Modèle numérique de terrain local obtenu à partir des points télémètres seuls pour l'itération 300. La hauteur de l'environnement est codée en fonction de la couleur, du bleu au rouge.

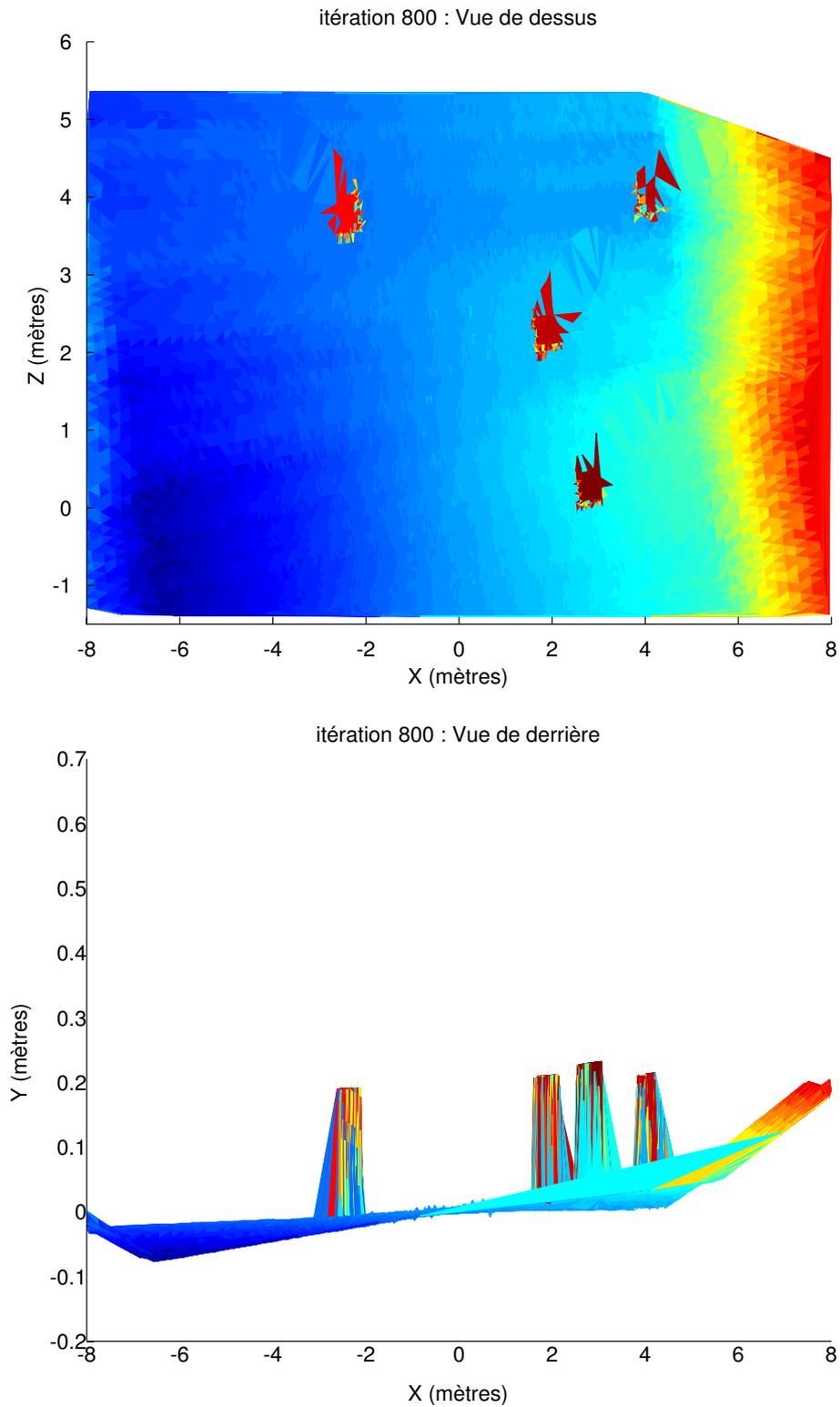


FIGURE 4.32 – Modèle numérique de terrain local obtenu à partir des points télémètres seuls pour l'itération 800. La hauteur de l'environnement est codée en fonction de la couleur, du bleu au rouge.

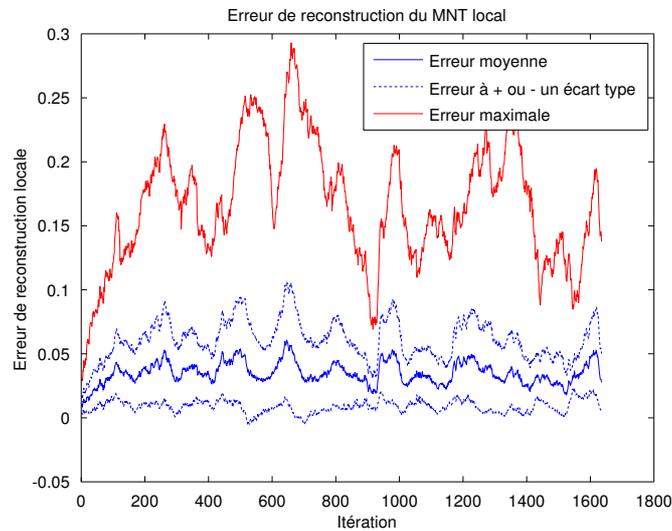


FIGURE 4.33 – Erreur de reconstruction locale obtenue sur l'ensemble de la trajectoire simulée. L'erreur moyenne est affichée en bleu. L'erreur à \pm un écart type est affichée en pointillé bleu, l'erreur maximale est affichée en rouge.

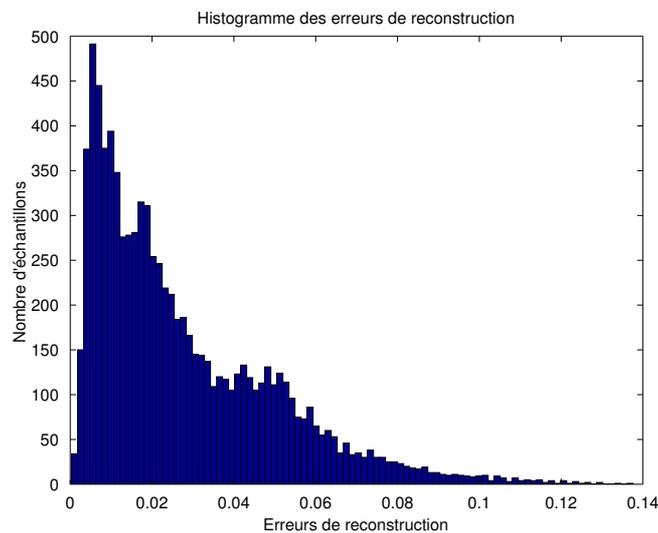


FIGURE 4.34 – Histogramme des erreurs de reconstruction obtenu sur une itération du processus.

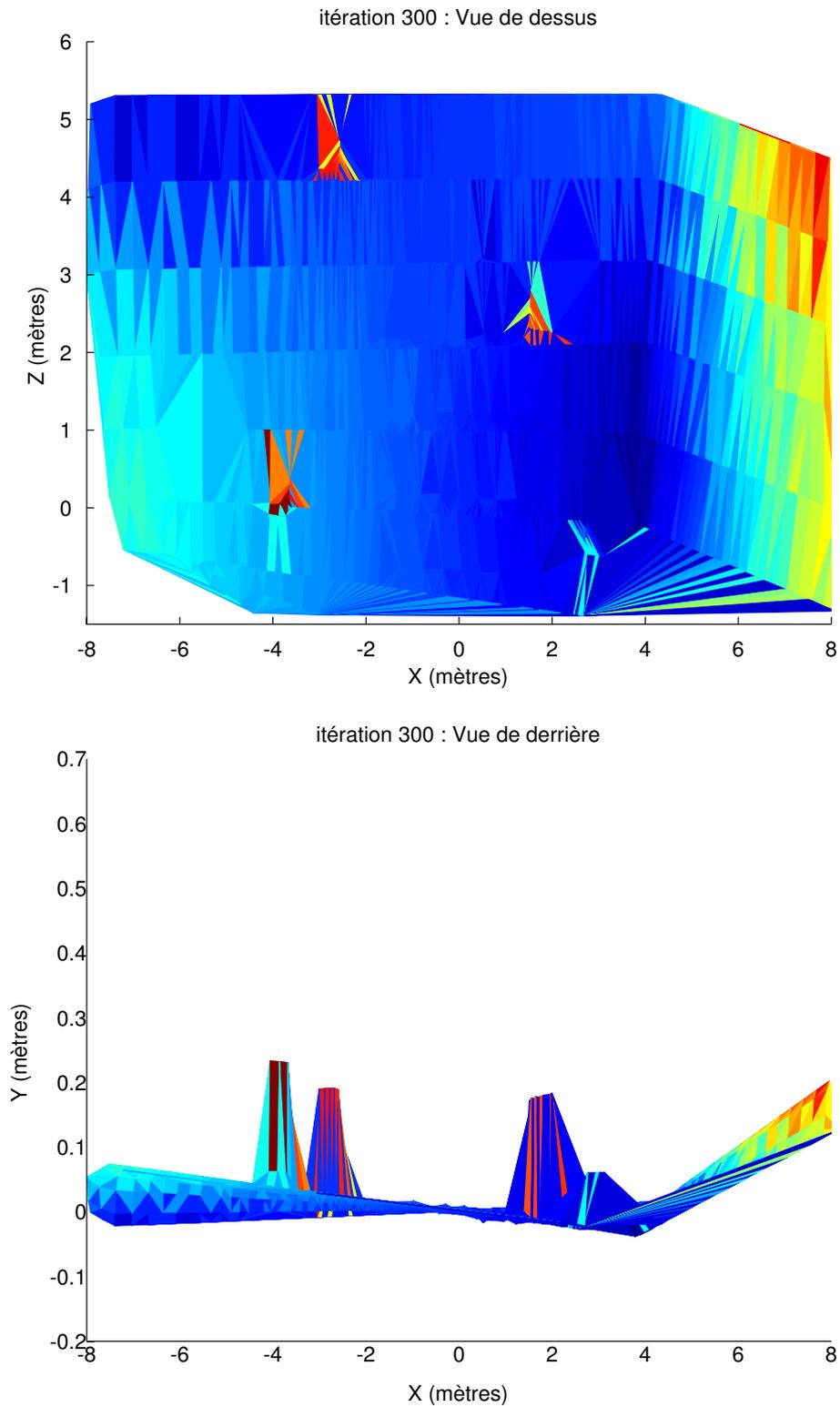


FIGURE 4.35 – Modèle numérique de terrain local obtenu pour l'itération 300 avec peu de données télémétriques (une acquisition tout les mètres).

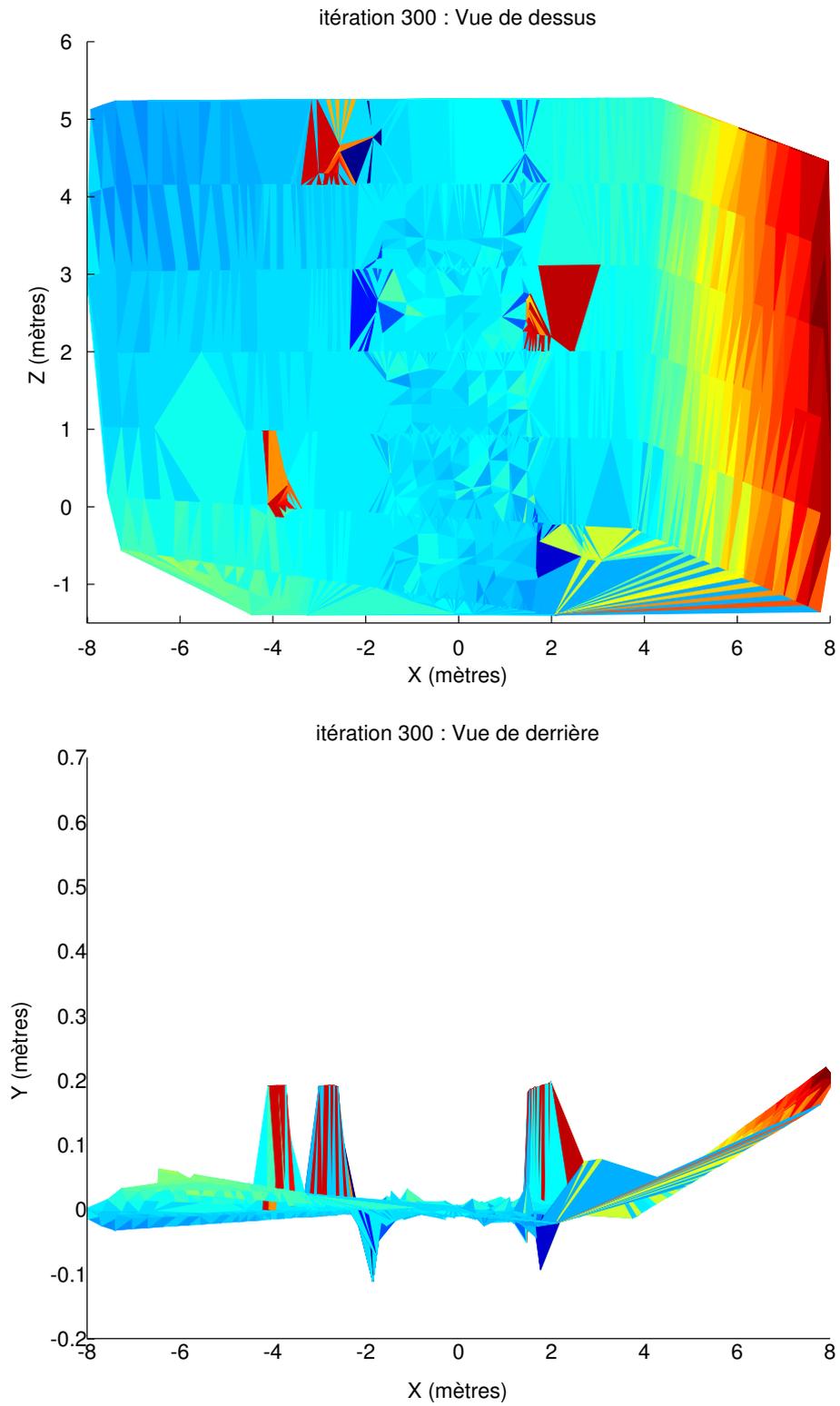


FIGURE 4.36 – Modèle numérique de terrain local obtenu pour l'itération 300, augmenté avec les données caméra.



FIGURE 4.37 – Notre véhicule d’expérimentation AROCO lors du Symposium ROBOTICS 2010.

d’un véhicule et la carte d’élévation de l’environnement d’un point de vue pratique. Ce scénario a aussi mis en évidence un problème d’appariement de points dans une image caméra. Ce phénomène a comme conséquence de faire diverger le calcul de pose du véhicule à long terme. Cependant, le MNT local reconstruit est très peu influencé par cette dérive. De plus cette carte est construite de façon dense et précise grâce à l’utilisation des informations provenant du télémètre.

4.3 Résultats en environnement réel

4.3.1 Résultats de localisation

4.3.1.1 Dispositif expérimental

Notre approche de fusion de données ayant été validée sur notre simulateur, nous pouvons l’expérimenter sur notre véhicule AROCO (voir figure 4.37). Sur cette plate-forme se trouvent les différents capteurs dont nous avons besoin, c’est à dire ceux qui ont été utilisés lors des expérimentations sur simulateur. Il s’agit de :

La caméra La caméra est une F-131B de la société MARLIN disposant d’un capteur CCD monochrome de matrice 1280×1024 fonctionnant à 15 Hz. Les paramètres intrinsèques et de distorsion de la caméra ont été calibrés en utilisant la

boite à outils Matlab développée par Jean-Yves Bouguet ⁵.

Le télémètre Le télémètre est un LMS221 de la marque SICK fonctionnant à 40 Hz et fournissant un point tous les demi-degré sur 180°. La précision du capteur donnée par le constructeur est d'un centimètre.

Le véhicule et ses capteurs L'instrumentation du véhicule comporte par défaut l'odométrie sur chacune des roues ainsi que l'angle des trains directeurs (avant et arrière). Toutes ces informations sont transmises à une cadence de 10 Hz.

4.3.1.2 Résultats sans correction de la dérive

Dans cette section, nous montrons les résultats obtenus en terme de localisation dans un contexte réel avec notre approche de fusion caméra - télémètre laser. Pour cela, le système caméra-télémètre a été calibré en suivant la méthode décrite en annexe A.2. Le modèle d'évolution du véhicule est le même que celui énoncé en section 4.2.1.1. La vérité terrain sera donnée par l'utilisation d'un GPS-RTK disposé au milieu de l'essieu arrière (situé donc à l'origine du repère du véhicule). La méthodologie de prise en compte des observations point par point s'est révélée aussi précise dans les tests précédents en simulation que la méthode globale. Cependant elle est beaucoup plus robuste aux faux appariements, nous regarderons donc seulement les résultats obtenus avec cette méthode.

L'expérimentation s'est déroulée sur le site expérimental du Cemagref situé à Montoldre dans l'Allier. La trajectoire parcourue par le véhicule est d'environ 100 mètres. Elle est composée d'un long demi tour à environ 40 mètres après le début de la trajectoire et le point d'arrivée se situe proche du point de départ. Sur cette trajectoire se situe une bute d'environ 2m de haut (un schéma représentant le parcours est disponible en figure 4.42).

La figure 4.38 montrent l'erreur totale de localisation comparée au GPS-RTK (le calcul est fait en utilisant les composantes en X et en Z du vecteur d'état seulement) en ayant initialisé le cap du véhicule de façon empirique (une vue de dessus du résultat de l'expérimentation est proposée en figure 4.39). Cette erreur montre des changements importants à certains moments (notamment vers 20m parcourus). Ces changements s'expliquent à cause de la présence des pentes situées sur la trajectoire du véhicule. Ces pentes n'apportent pas d'erreur puisque le système est capable de les prendre en compte. Cependant elles provoquent un dysfonctionnement du laser (points infinis lorsque le laser ne heurte pas le sol). Le système initialise alors des informations provenant uniquement de la caméra comme amer visuel ce qui est moins précis.

5. http://www.vision.caltech.edu/bouguetj/calib_doc/

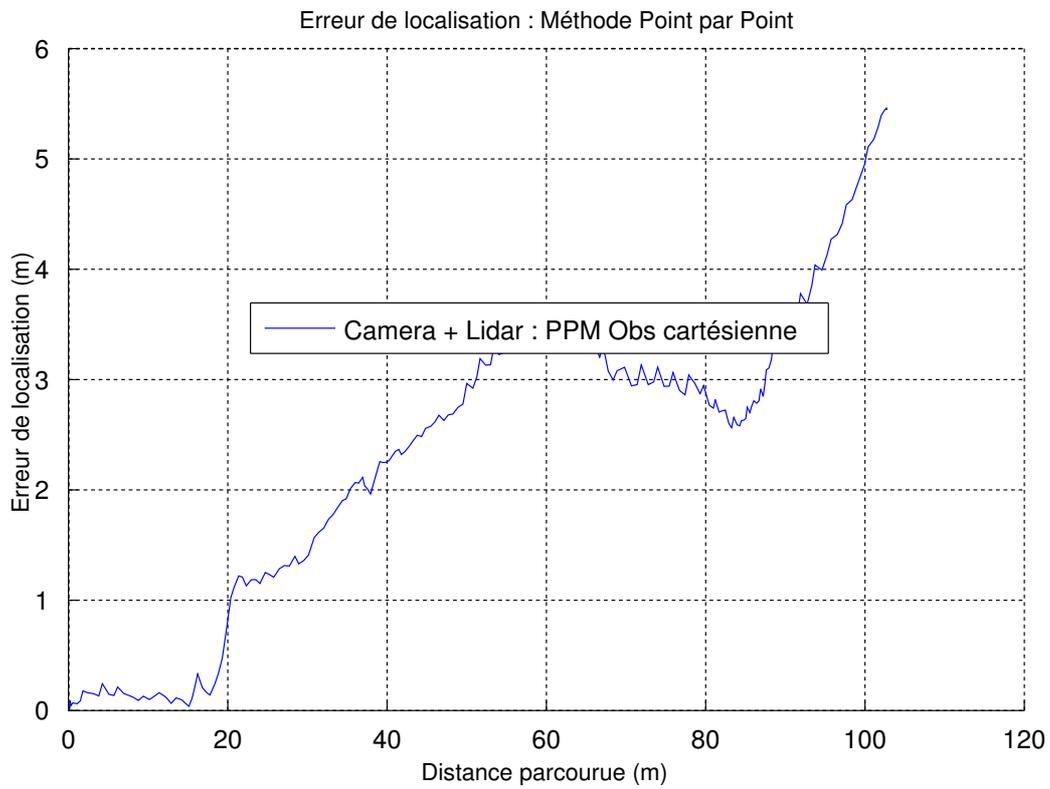


FIGURE 4.38 – Erreur totale de localisation en expérimentation réelle sans correction de la dérive.

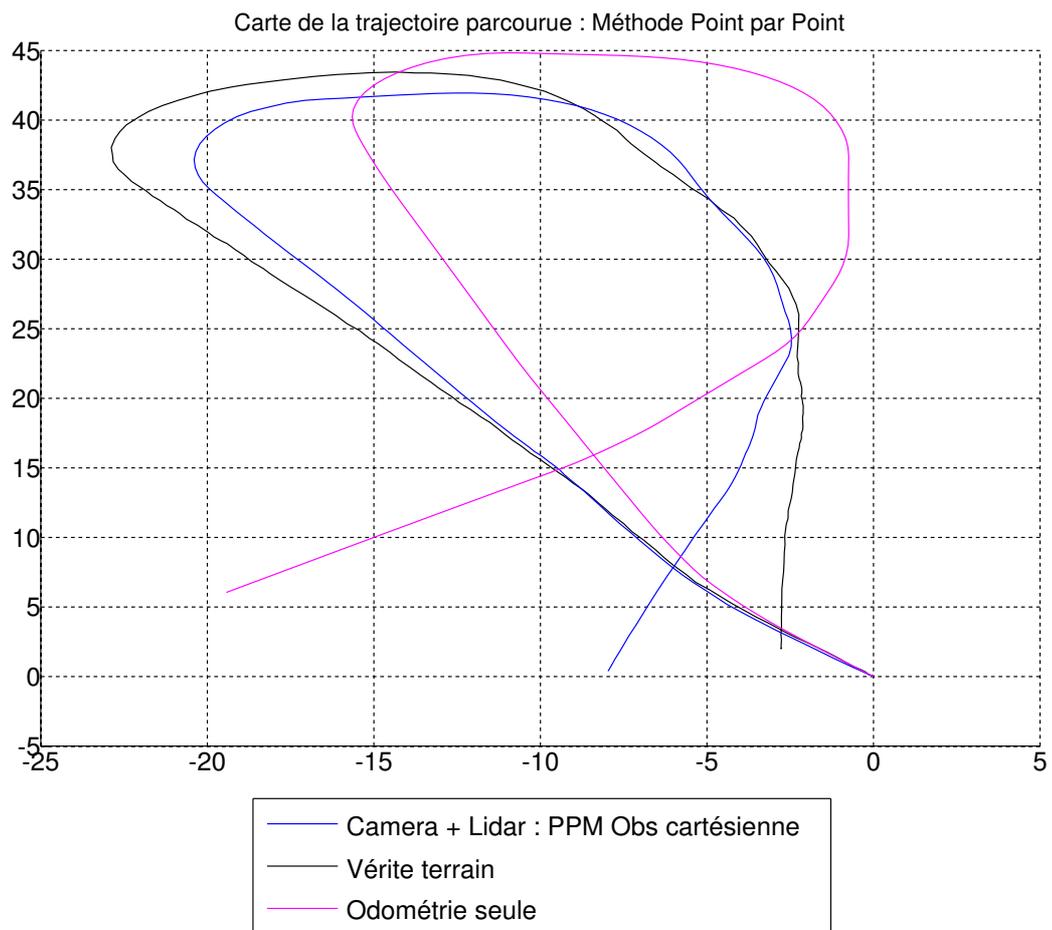


FIGURE 4.39 – Vue du dessus des trajectoires calculées en expérimentation réelle sans correction de la dérive.

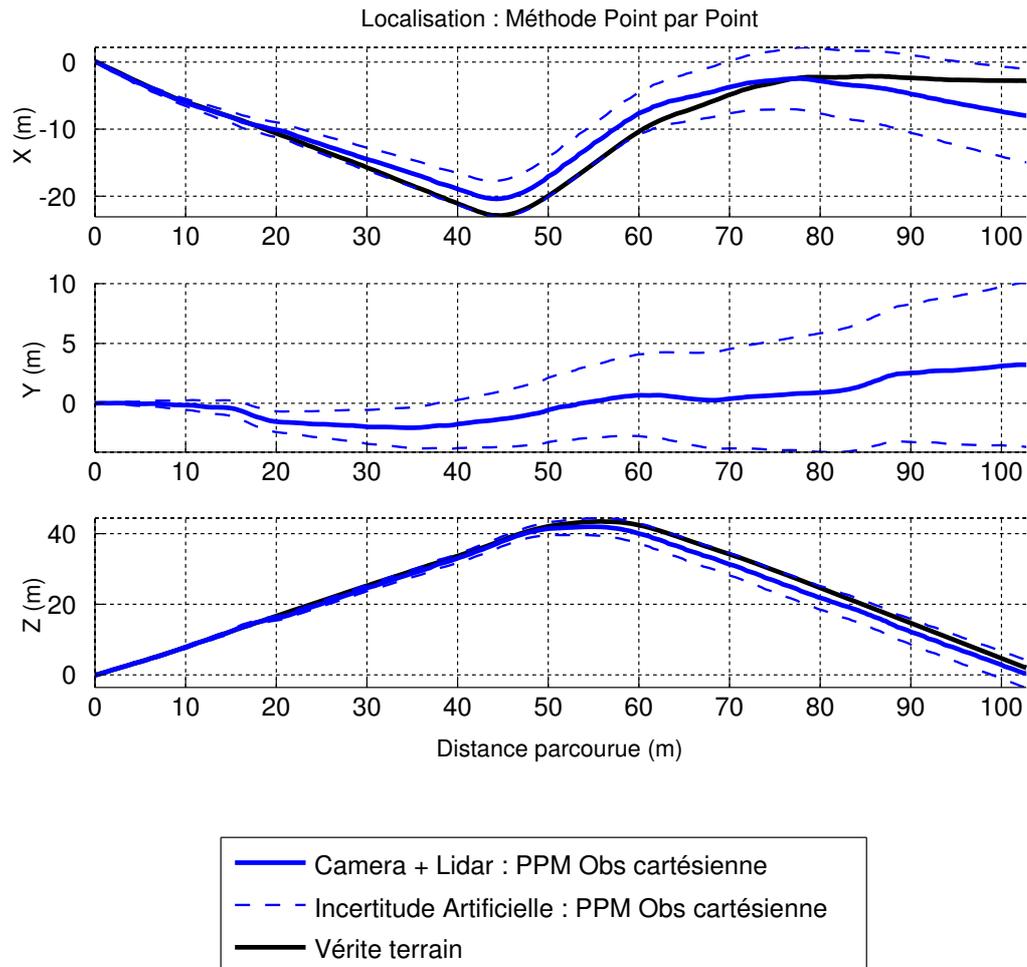


FIGURE 4.40 – Paramètres de localisation suivant chaque variable en expérimentation réelle sans correction de la dérive.

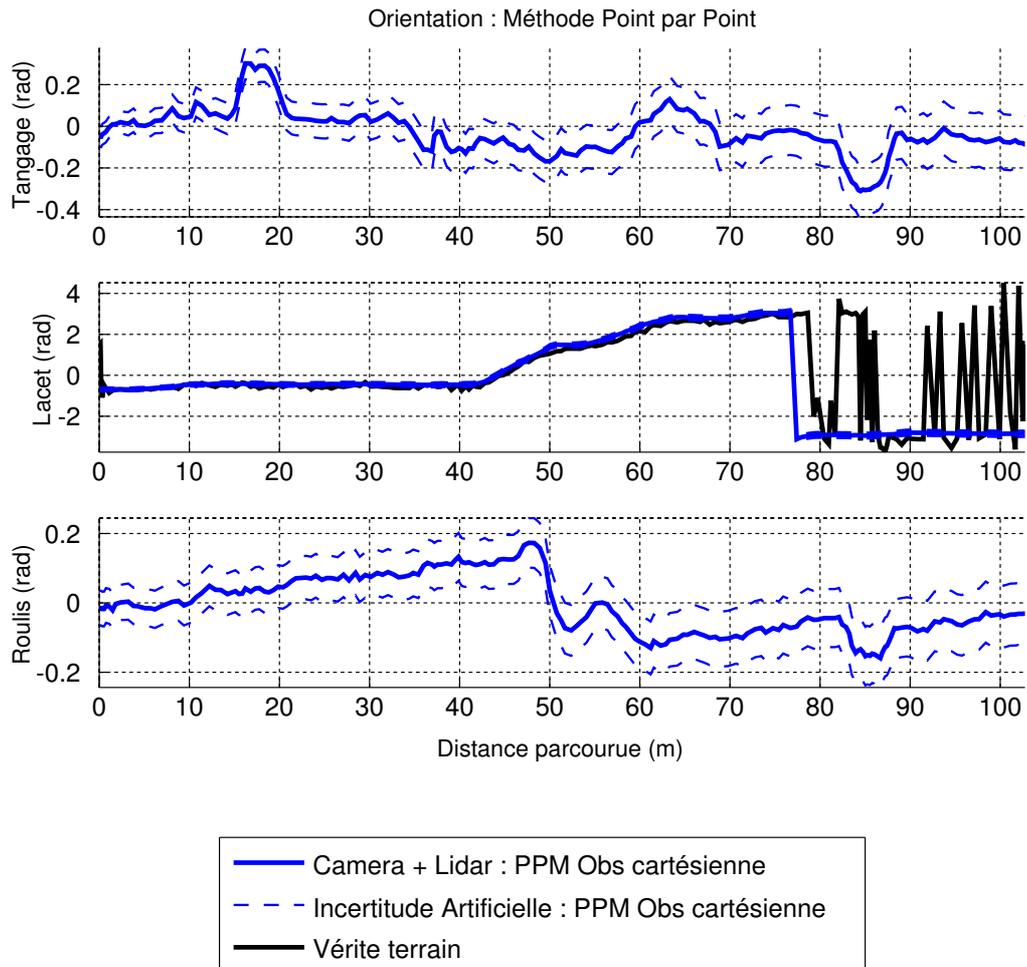


FIGURE 4.41 – Paramètres d'orientation suivant chaque variable en expérimentation réelle sans correction de la dérive.

Les figures 4.40 et 4.41 montrent respectivement l'estimation des paramètres de positionnement du véhicule en X, Y et Z et les paramètres d'orientation du robot en tangage, lacet et roulis faite par notre système de fusion de données (les grandes variations du lacet en fin de parcours sont dues au calcul du cap du véhicule par le GPS-RTK qui oscille entre $-\pi$ et π). Puisque le point d'arrivée se situe suffisamment proche du point de départ et sur une surface plane on peut dire que le tangage, le roulis et la coordonnée en Y du véhicule doivent être les mêmes au début et à la fin du parcours. A partir de cette supposition on peut voir que le système proposé permet bien d'obtenir des valeurs pour ces paramètres en fin de parcours quasiment identiques à celles au début du parcours. Pendant cette trajectoire, on peut voir que les influences de l'environnement sur l'état du véhicule sont qualitativement correctement évaluées (voir figure 4.42). En zone rouge et en zone bleue le tangage du véhicule augmente à cause de la bute. Il diminue ensuite à l'arrivée en zone vert et zone jaune lors de la descente de cette bute.

4.3.1.3 Résultats avec correction de la dérive

Dans cette section nous allons appliquer notre système de correction de dérive sur des données réelles. Le jeu de données est exactement le même que celui utilisé en section 4.3.1.2.

Ici, les informations intègres proviennent du GPS-RTK disponible sur notre véhicule AROCO. Une donnée est utilisée par le système lorsqu'un certain laps de temps s'est écoulé, ici ce temps est réglé à 15 secondes. Les bruits d'observation de la mesure sont les mêmes que ceux utilisés en section 4.2.1.3. La figure 4.43 montre une vue de dessus de la trajectoire obtenue après mise en place de la correction de dérive.

Les figures 4.44 et 4.45 montrent les paramètres de l'état du véhicule après correction de la dérive. On voit que les positions en X, Z et en lacet sont bien corrigées grâce à l'ajout des mesures intègres. D'un autre côté, les estimations des autres paramètres (Y, tangage, lacet) restent intègres.

Il faut noter que même si l'expérimentation est faite avec un GPS centimétrique, n'importe quelle information intègre peut convenir. Cette expérimentation a montré la correction du processus de localisation du système de fusion de données en utilisant les données intègres d'un GPS-RTK. Cette correction donne de bons résultats et maintient l'intégrité de la pose du véhicule.

4.3.2 Résultat de construction du MNT local

Le principe de reconstruction 3D de l'environnement ayant été validé sur simulateur, nous pouvons appliquer ce dispositif sur une plateforme réelle. Pour cela, le système va utiliser le jeu de données provenant du site expérimental de Montoldre

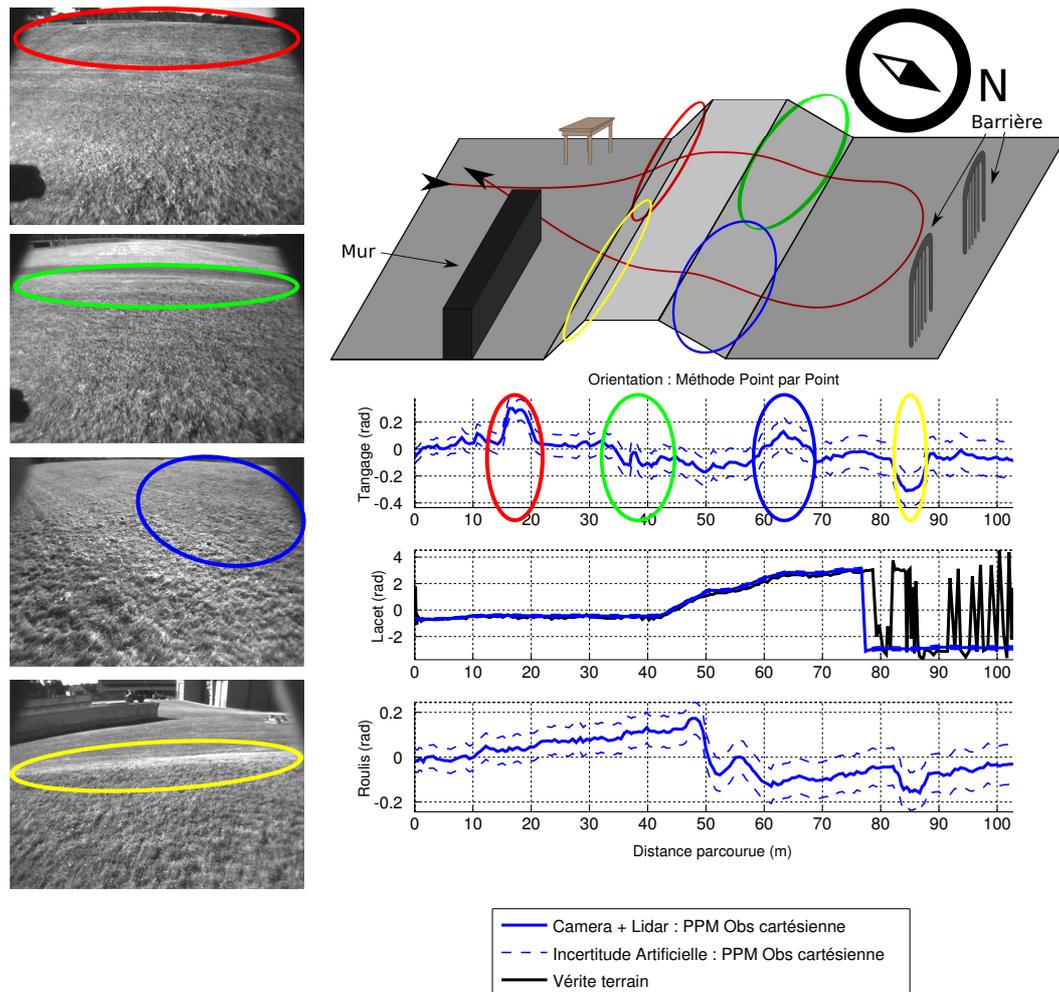


FIGURE 4.42 – La correspondance entre la variation de l'état et l'environnement pour des points clés du parcours.

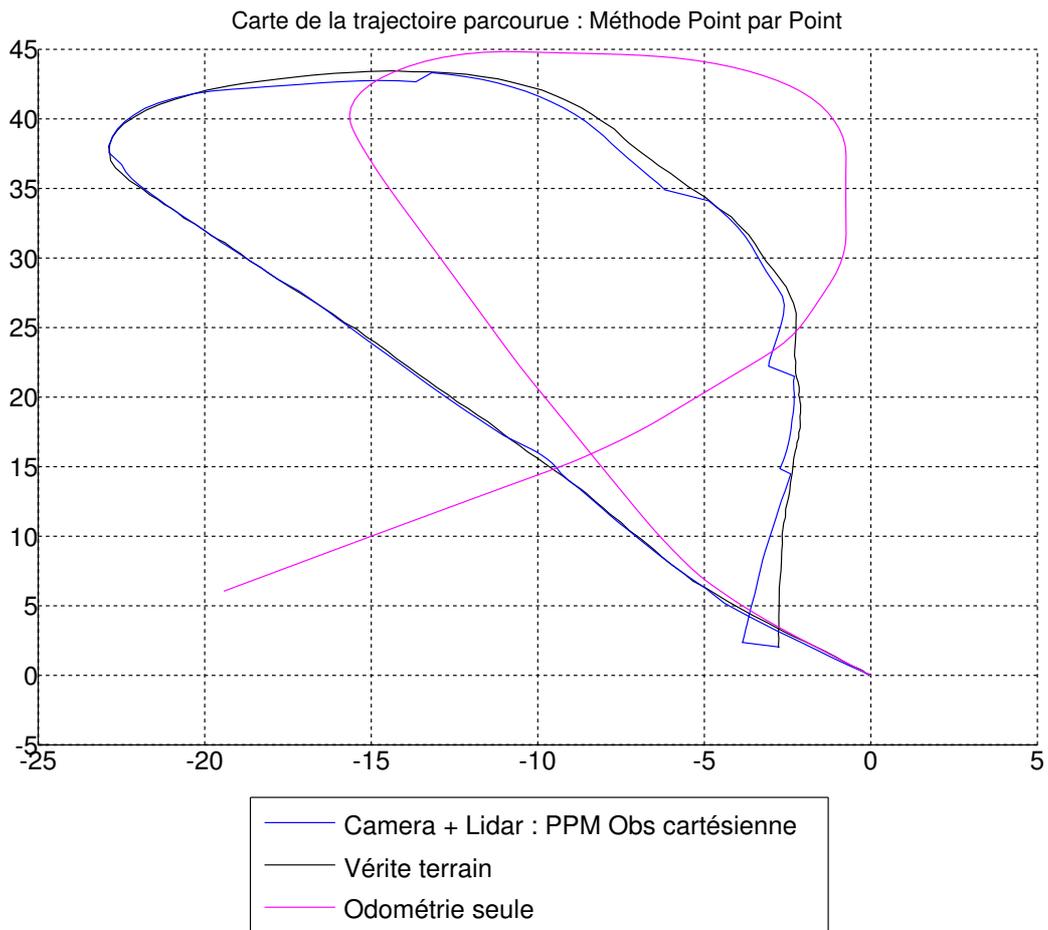


FIGURE 4.43 – Vue du dessus de la trajectoire parcourue en environnement réelle après correction de la dérive.

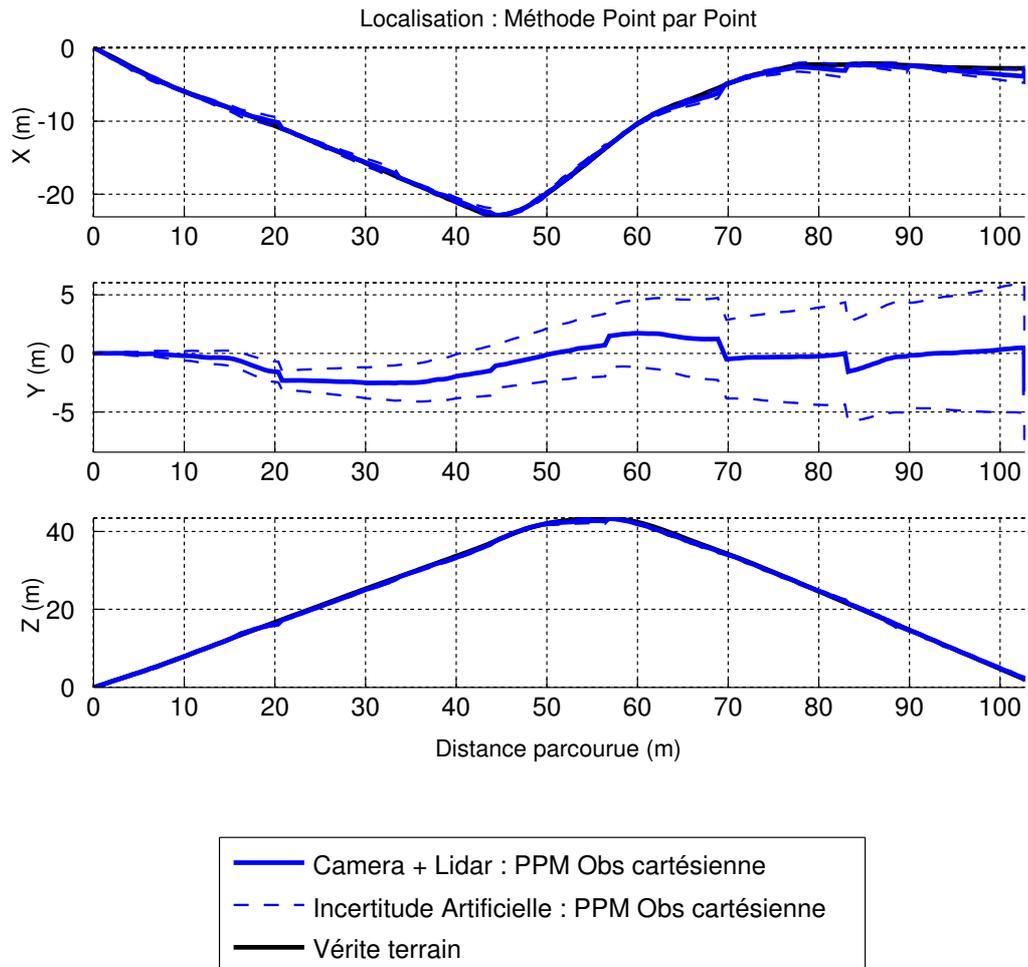


FIGURE 4.44 – Erreur de localisation suivant chaque variable après correction de la dérive.

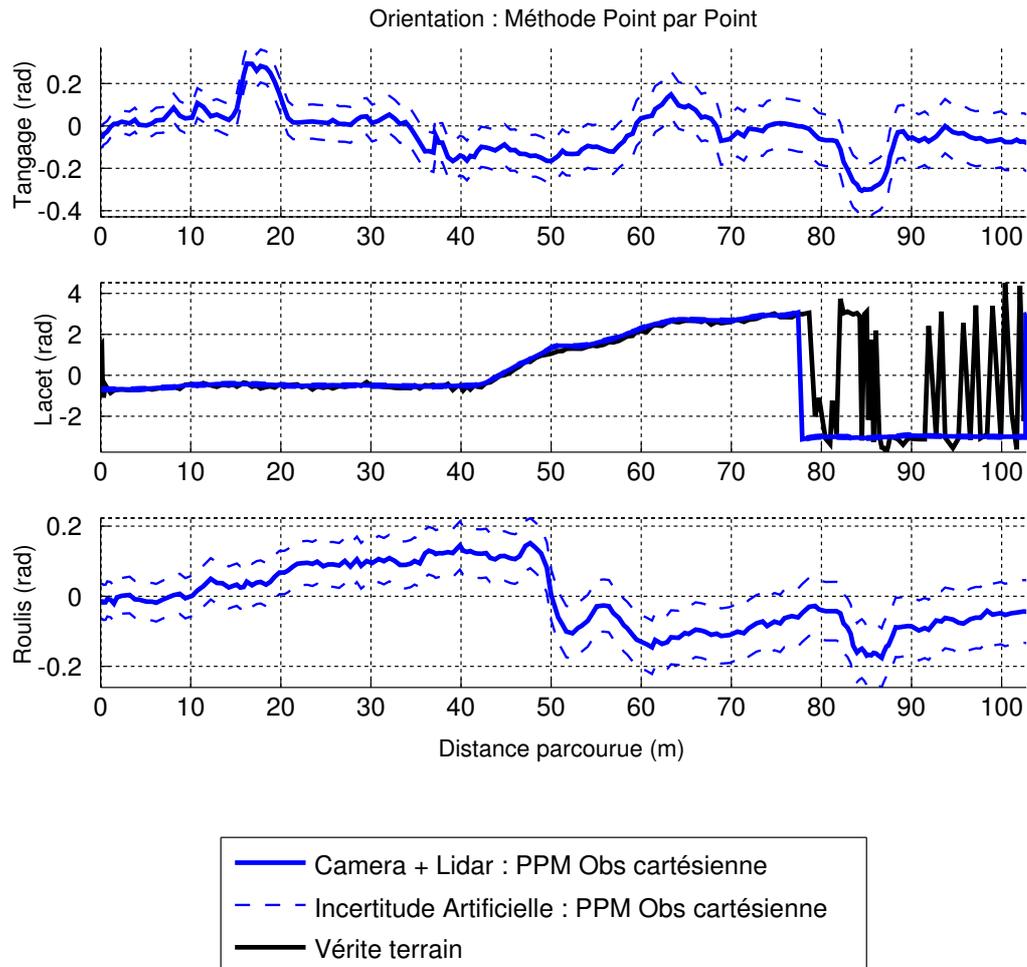


FIGURE 4.45 – Erreur d'orientation suivant chaque variable après correction de la dérive.

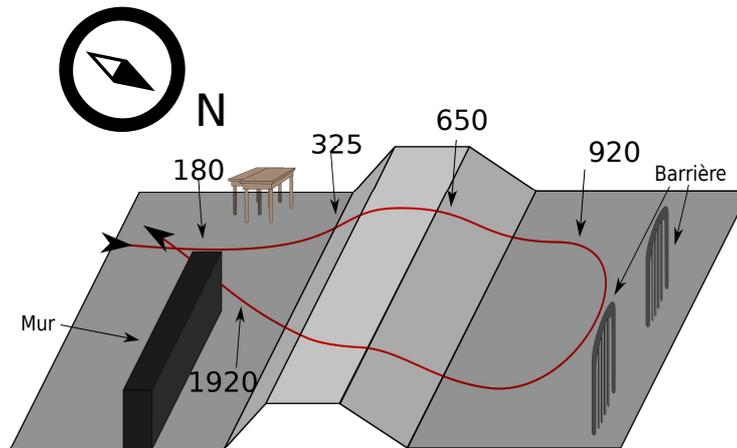


FIGURE 4.46 – Schéma des moments clés de l'expérimentation où seront montrés le MNT en local.

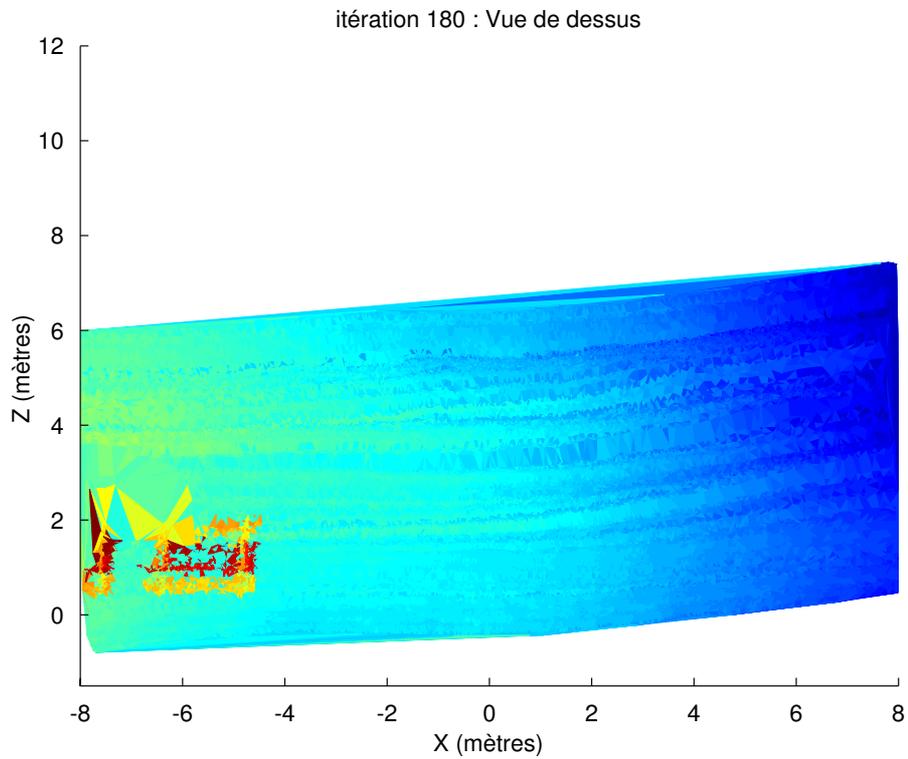
qui est utilisé pour quantifier les résultats de localisation en section 4.3.1.2. Dans cette expérimentation, les données télémétriques et les données caméra vont être fusionnées. En effet, à certains moments de la trajectoire, le laser du télémètre ne touche pas le sol (montée sur la bute), dans ces cas là le système utilise des points initialisés dans la caméra pour se localiser et pour reconstruire le MNT local.

La figure 4.46 présente les endroits où vont être montrés les résultats du MNT local. Les figures 4.47, 4.48, 4.49, 4.50 et 4.51 montrent les cartes d'élévations obtenues en local pour ces moments clés de la trajectoire. La figure 4.48 représentant l'itération 325 par exemple montre la première bute que franchit le véhicule. On y voit la pente reconstruite par le système. Sur la figure 4.51 à la fin du parcours on peut voir le mur sur la gauche ainsi que qu'une table en bois sur la droite.

4.3.3 Bilan

Dans cette section, nous avons montré la capacité de notre algorithme à fonctionner sur des données réelles avec une exécution temps réel. Le système a montré des améliorations notables du calcul de pose du véhicule en comparaison avec l'utilisation de l'odométrie seule. Les lacet, roulis, tangage du véhicule sont bien estimés grâce à l'utilisation des données précises fournies par le télémètre.

Nous avons présenté notre système de correction de dérive par une mesure intégrée fournie par exemple par un GPS-RTK. L'ajout d'une covariance artificielle nous a permis de garder une position estimée consistante. Cela nous permet de fusionner l'état de notre véhicule avec n'importe quelle source d'information intègre tout en conservant cette consistance. Avec les résultats précédent nous avons vu que cet objectif était bien réalisé. Enfin nous avons montré l'aptitude de notre système à construire une carte d'élévation locale précise et dense grâce à l'utilisation des



1.2

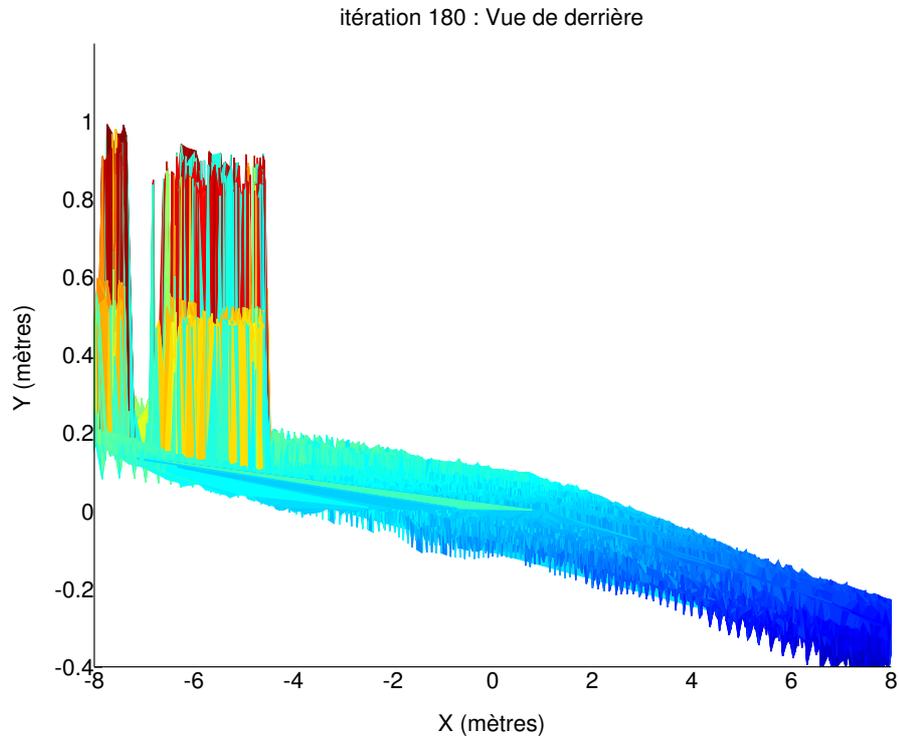
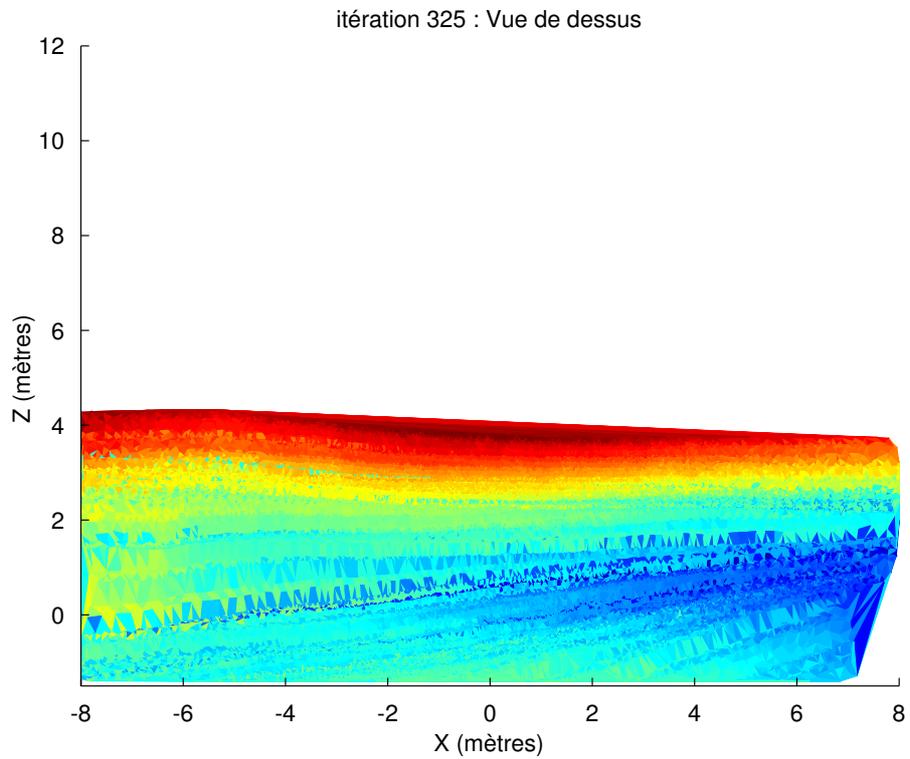


FIGURE 4.47 – Modèle numérique de terrain local obtenu en environnement réel pour l'itération 180. La reconstruction d'une table en bois apparaît sur la gauche.



1.2

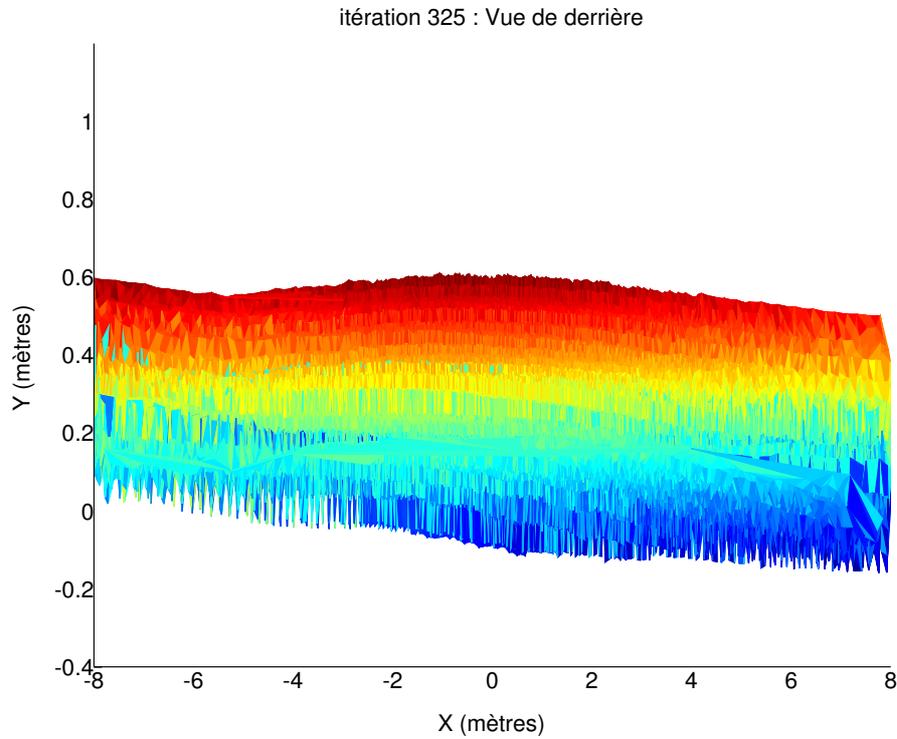
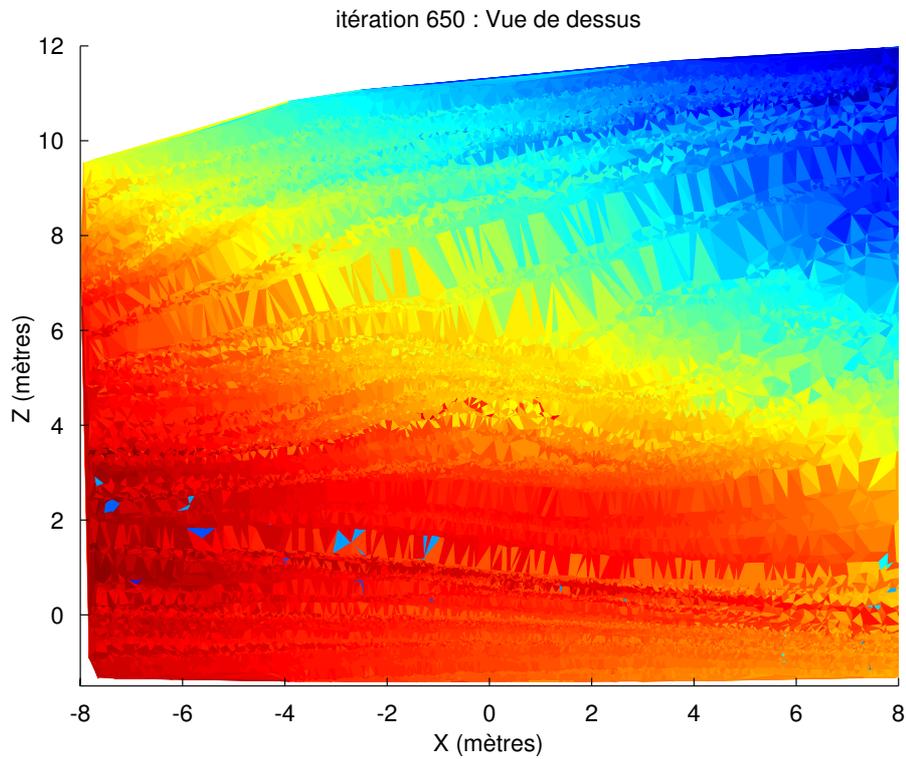


FIGURE 4.48 – Modèle numérique de terrain local obtenu en environnement réel pour l'itération 325. La bute apparaît à $Z = 3m$.



1.2

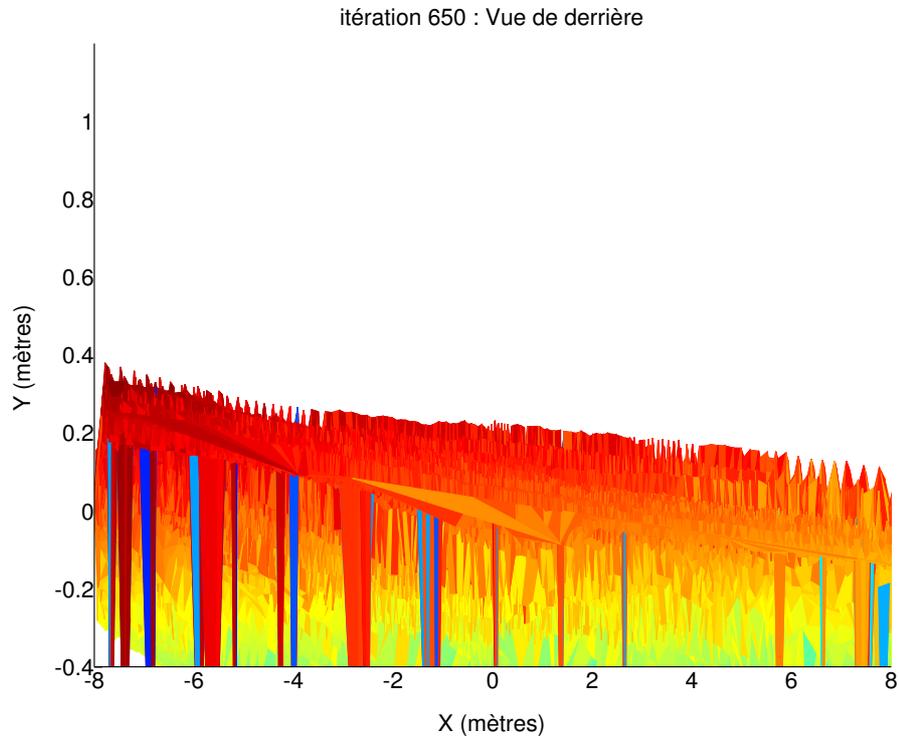
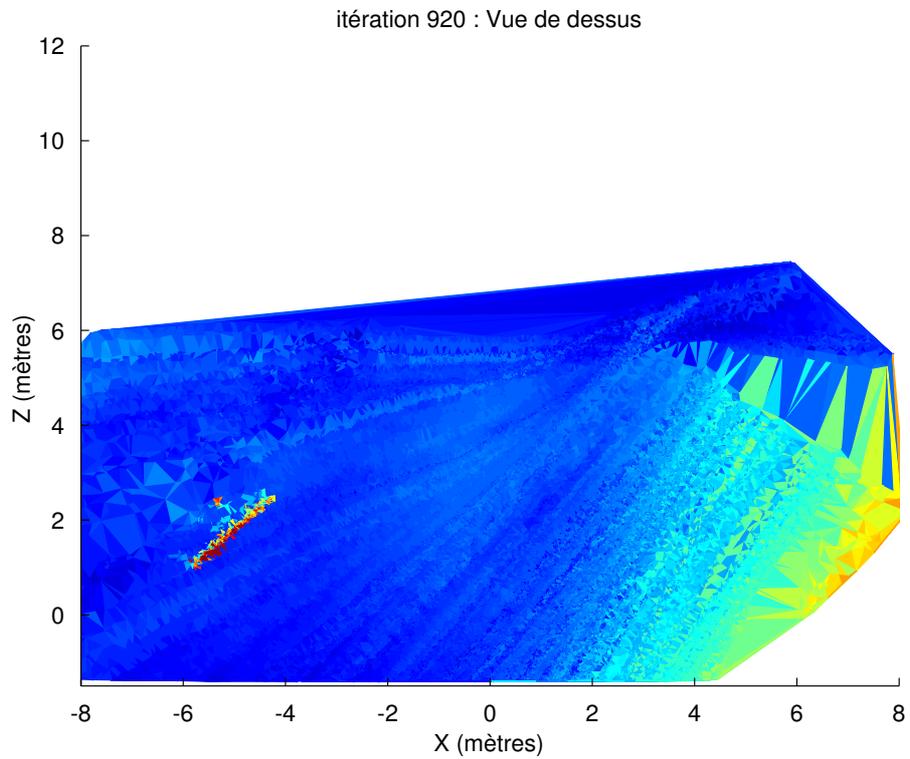


FIGURE 4.49 – Modèle numérique de terrain local obtenu en environnement réel pour l'itération 650. La descente et la fin de la bute sont représentées.



1.2

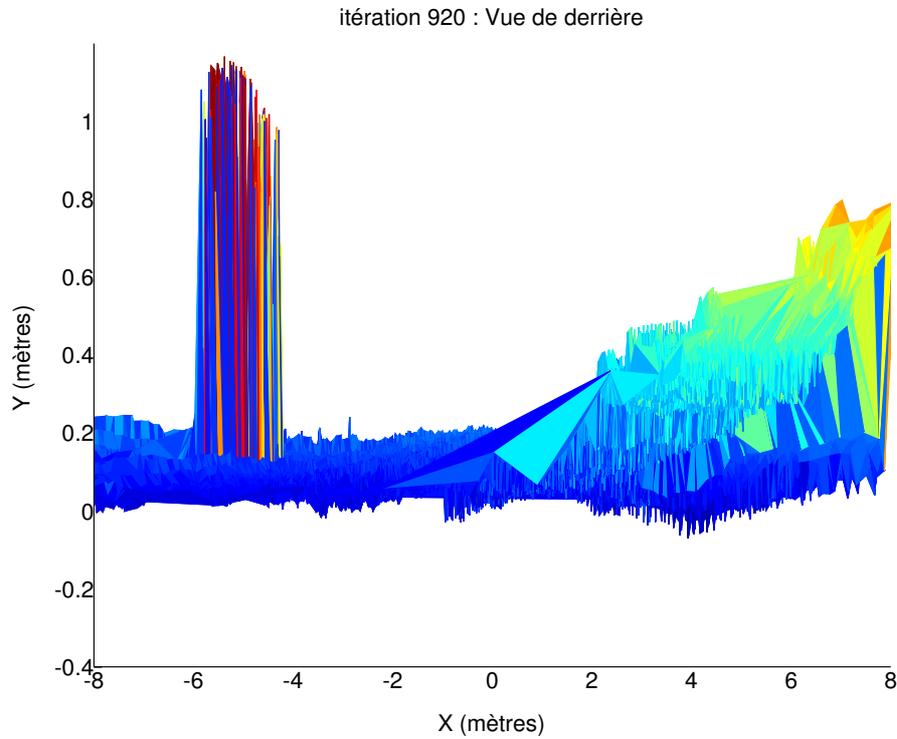
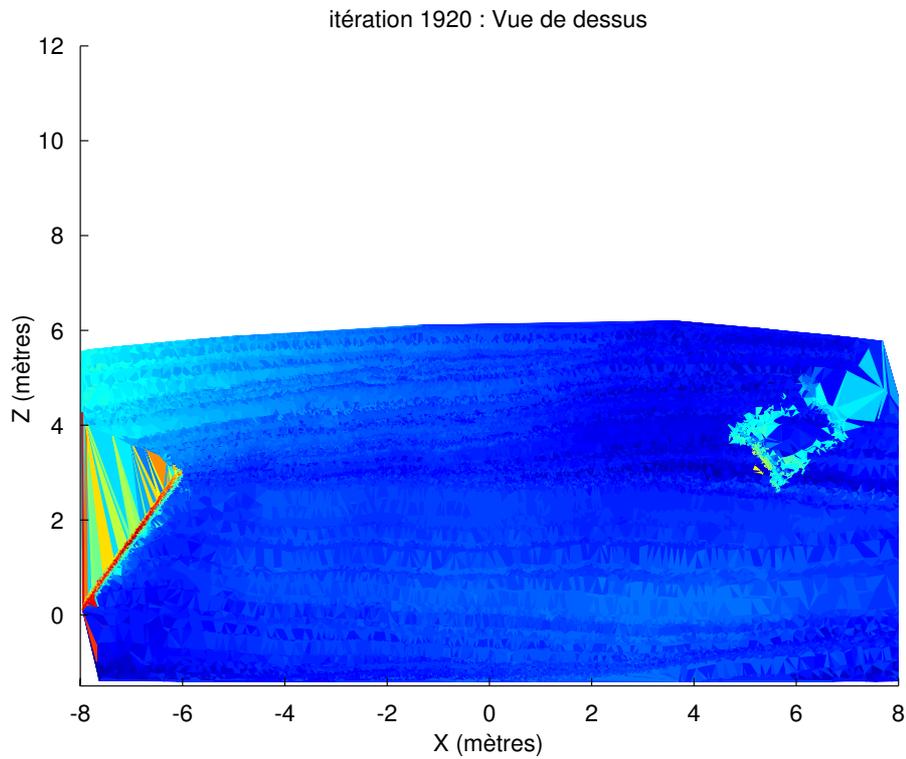


FIGURE 4.50 – Modèle numérique de terrain local obtenu en environnement réel pour l'itération 920. A gauche se trouve la reconstruction d'une barrière en métal et sur la droite apparaît le profil de la bute.



1.2

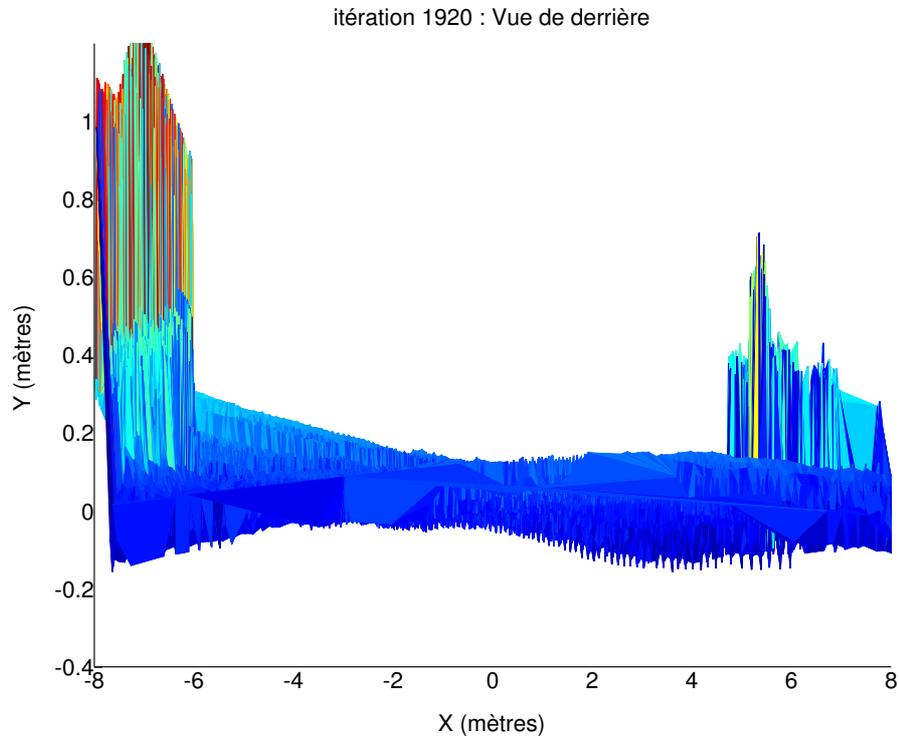


FIGURE 4.51 – Modèle numérique de terrain local obtenu en environnement réel pour l'itération 1920. La reconstruction d'un mur en béton se trouve sur la gauche de la carte tandis que l'on retrouve la table en bois sur la droite.

nappes télémétriques. Cette compétence a permis par la suite de mettre en œuvre un système complet de navigation en utilisant d'autres travaux sur la planification de trajectoires dans un milieu naturel ([Delmas, 2011]).

4.4 Résultat de temps de Calcul

4.4.1 Temps de calcul de la mise à jour de l'EKF

Pour notre application, il est important de maîtriser les ressources utilisées pour pouvoir les partager efficacement entre système de perception et système de guidage. Dans cette section nous allons donner les temps de calculs obtenus en fonction de la taille du vecteur d'état afin de comparer les approches en méthode globale et point par point. Dans un premier temps, les résultats montrés seront ceux obtenus avec une procédure de test où un vecteur d'état quelconque est utilisé. Pour ce test, la taille de ce vecteur d'état est générée de façon croissante.

La figure 4.52 montre les temps de calculs obtenus pour la méthode globale. La figure 4.53 montre les temps de calculs obtenus pour la méthode point par point.

Comme prévu, les étapes les plus coûteuses sont le calcul du gain optimal de Kalman ainsi que la mise à jour de la covariance. Si on compare les résultats de la figure 4.52 avec ceux de la figure 4.53 nous voyons que le temps passé au calcul du gain de Kalman a été significativement diminué alors que le celui de la mise à jour de la covariance est resté quasiment identique. Dans la section 2.6 du chapitre 2 nous avons montré les calculs et courbes théoriques du coût opératoire d'une mise à jour du filtre de Kalman. Même si le gain en temps de calcul n'est pas aussi important que prévu à cause de surcoût d'implémentation, notre approche point par point se trouve donc quand même plus rapide que la méthode globale d'un facteur d'environ 1.6. **Donc il est possible de stopper le processus lorsque le temps disponible est écoulé.**

4.4.2 Temps de calcul du processus entier en expérimentation

Dans cette section, nous allons donner les temps d'exécution du système entier en comparant les méthodologies de traitement global et point par point. Ces temps de traitement comprennent toutes les étapes de la mise à jour du filtre de Kalman étendu et les temps de mise en correspondance de patch entre les images. Les figures 4.54 et 4.55 montrent respectivement les temps d'exécution du processus pour la méthode globale et la méthode en point par point.

La figure 4.54 montre des temps de calculs en méthode globale bien plus importants que ceux donnés en figure 4.52. Ceci vient du fait que le temps pour calculer les mises en correspondance est très grand. Ce temps est beaucoup réduit dans la

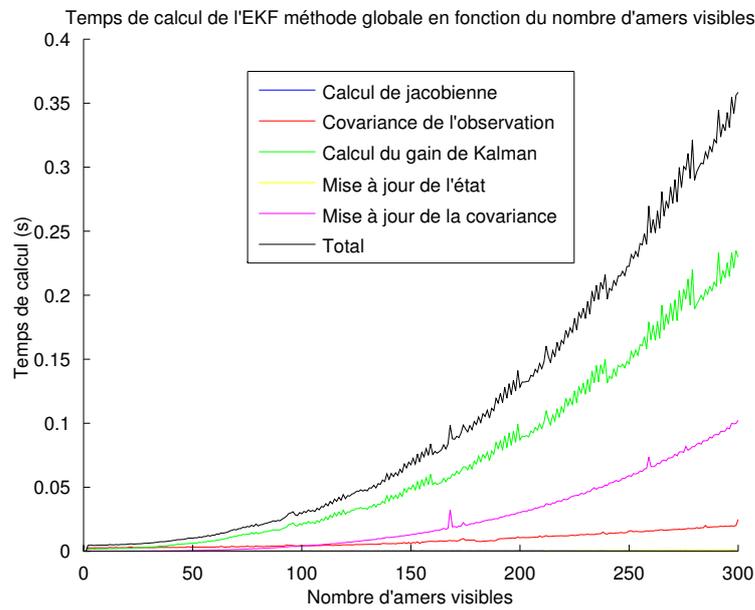


FIGURE 4.52 – Temps de calcul de la mise à jour de l'EKF pour la méthode globale en fonction du nombre d'amers dans le vecteur d'état.

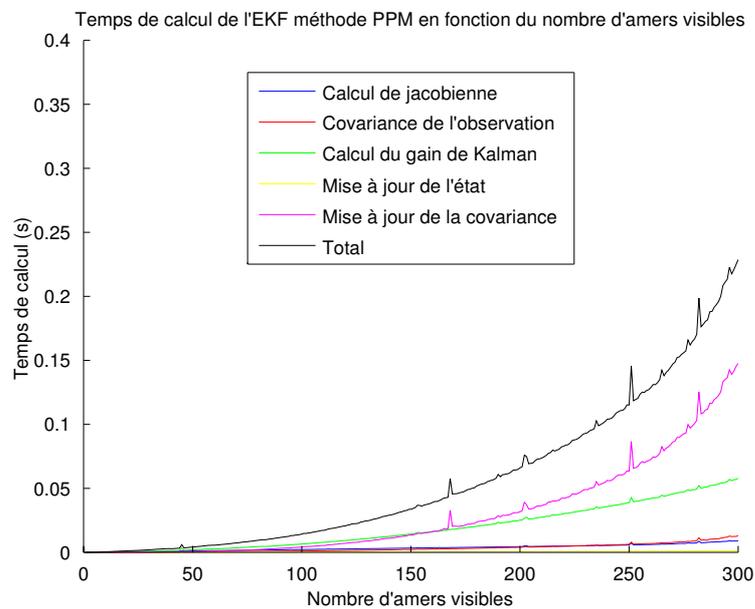


FIGURE 4.53 – Temps de calcul de la mise à jour de l'EKF pour la méthode point par point en fonction du nombre d'amers dans le vecteur d'état.

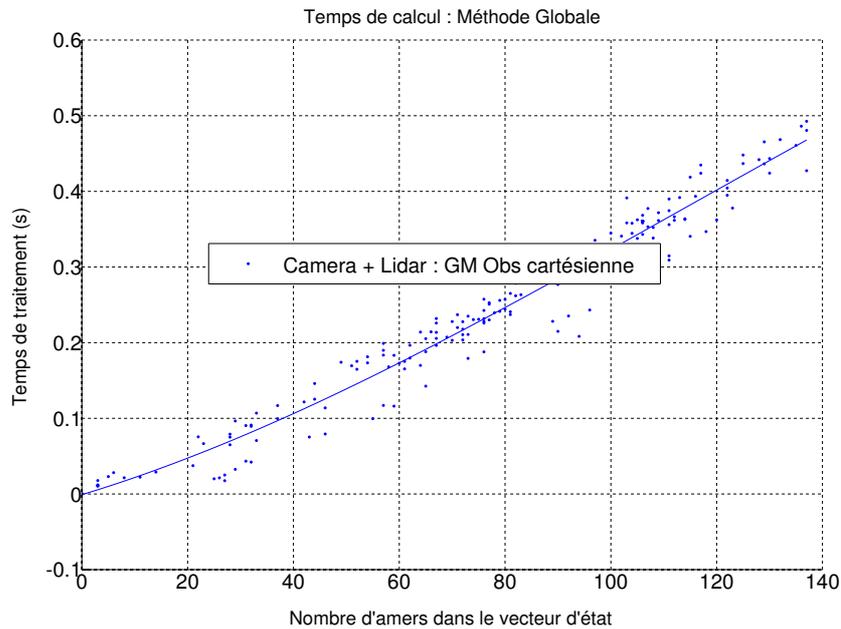


FIGURE 4.54 – Temps de calcul en méthode globale en fonction du nombre d’amers dans le vecteur d’état. La courbe bleue montre un lissage par un polynôme d’ordre 3.

version en point par point du fait de la réduction de l’espace de recherche des appariements des points. Le temps de calcul total de l’algorithme en méthode point par point (figure 4.55) est proche de celui donné par la figure 4.53. Cela vient du fait que la focalisation diminue grandement le temps passé à réaliser la recherche d’appariement. Les temps de calculs en méthode point par point sont donc plus compatibles avec un dispositif temps réel.

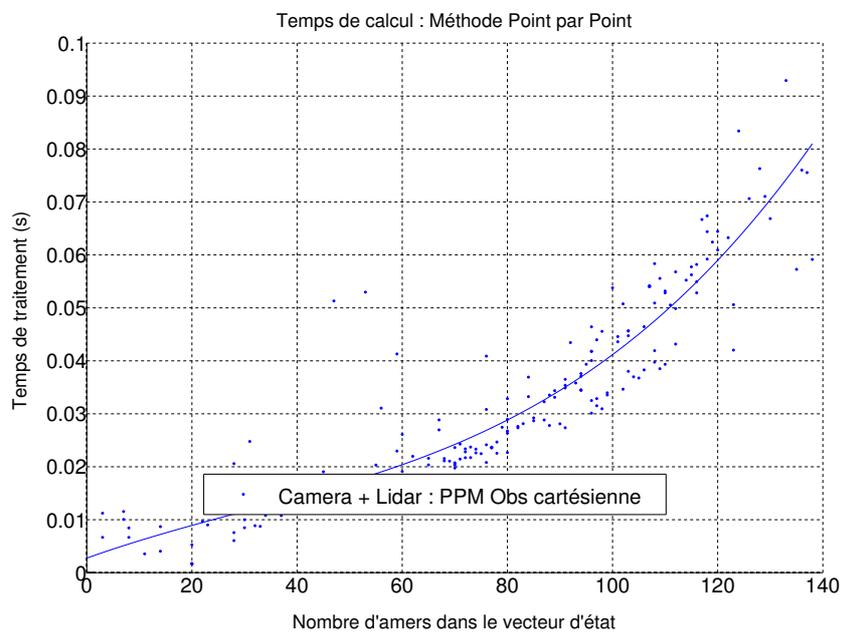


FIGURE 4.55 – Temps de calcul en méthode point par point en fonction du nombre d'amers dans le vecteur d'état. La courbe bleue montre un lissage par un polynôme d'ordre 3.

Conclusion - Perspective

Conclusion

Afin de mener à bien une tâche de navigation autonome, un système complet, efficace et sécurisé de perception et de guidage doit être mis en place. Les travaux présentés dans ce mémoire apportent une solution multi-fonctions de localisation 6D intègre et de perception active de l'environnement basée sur une combinaison de capteurs complémentaires : une caméra et un télémètre. Ce système perceptif a été implémenté avec succès sur un robot mobile afin de lui permettre d'accomplir une tâche de guidage autonome tout en maintenant son intégrité physique.

Le système de localisation est basé sur une odométrie visuelle augmentée des informations de profondeur du télémètre projetées dans l'espace de la caméra. Cette odométrie visuelle est capable d'évaluer les six paramètres de pose du véhicule dans son environnement. Pour palier l'inconsistance du filtre de Kalman étendu ainsi que le problème de pose initiale du véhicule dans son environnement une intégrité de localisation artificielle a été mise en place. Cette intégrité artificielle permet de garder une consistance de l'estimation de la pose du robot mobile dans son environnement afin de garantir un certain niveau de sécurité du système de navigation à la fois pour le robot et son entourage. Finalement l'utilisation de données absolues et précises permet de corriger la dérive du processus.

Afin de garantir l'intégrité physique d'un véhicule piloté automatiquement par un système de guidage un algorithme de construction de carte d'élévation a été proposé. Le couplage d'un télémètre laser à balayage et d'une caméra permet d'obtenir une reconstruction dense et précise de l'environnement pertinente pour l'évolution du véhicule. Afin de combler le manque d'informations potentiellement ignorées par le télémètre une interaction entre perception et décision a été mise en œuvre pour permettre l'observation des zones d'évolution probables du véhicule. Dans ce cas, l'utilisation de la caméra permet d'estimer l'élévation de ces zones.

Finalement, ce système de perception a permis de faire fonctionner en autonomie un véhicule d'expérimentation dans un contexte de navigation ayant pour but de rejoindre un objectif préalablement défini tout en maintenant l'intégrité physique du véhicule. Pour mettre en place cette expérimentation, ces travaux ont été

couplés à des travaux portant sur le guidage d'un véhicule autonome en environnement naturel. Ces derniers cherchent à évaluer la franchissabilité de l'environnement pour définir la meilleure trajectoire associée à son profil de vitesse. Cette collaboration a permis la mise en œuvre d'un système complet en terme d'autonomie, d'intégrité et de performance.

Perspectives

Les travaux présentés dans ce mémoire permettent de localiser un véhicule guidé automatiquement tout en construisant le MNT du sol sur lequel il se déplace. Pendant les tests du système en environnement simulé nous avons remarqué la présence de glissement dans la mise en correspondance des points entre deux images. Ces glissements ont des impacts importants dans notre algorithme de perception puisque s'il y a du glissement dans l'appariement la conséquence d'une mauvaise mise à jour est une mauvaise localisation du robot et donc une dérive de la localisation.

Ce phénomène apparaît plus significativement dans notre cas que dans d'autres applications d'odométrie visuelle puisque notre système est totalement détaché de l'environnement pour effectuer son calcul de pose. Les autres techniques d'odométrie visuelle reposent sur la recherche d'amers visuels remarquables dans une image qui correspondent toujours au même point 3D. Dans un premier temps on pourrait chercher à sélectionner les points télémétriques les plus remarquables (au sens du score de Harris par exemple) ce qui pourrait limiter le glissement des appariements.

Pour éviter ces phénomènes de glissement il faudrait être capable d'assurer un appariement correct de l'amer dans la caméra. Une technique utilisée en odométrie visuelle est de déformer le patch qui représente l'amer en fonction de son déplacement dans l'image. Cette transformation permet de réaliser un appariement à partir d'une information préalablement transformée qui est plus proche de la vraie projection du point 3D dans l'image que le patch enregistré précédemment. Pour cela une technique consiste à appliquer une homographie au patch qui représente l'amer en fonction du déplacement de la caméra. Le patch est alors considéré comme représentant une surface plane de normale connue. Cette technique a été réalisée dans plusieurs travaux comme dans [Charmette *et al.*, 2010] mais sont très coûteux en temps de calcul (environ 3 secondes pour 2000 patches transformés). Pour accélérer cette procédure, Charmette a déporté les calculs sur un GP-GPU (General Purpose - Graphics Processing Units). Cette technique pourrait être mise en place pour améliorer la précision des appariements afin d'éviter ces phénomènes de glissement et ainsi obtenir une estimation de la pose plus juste.

La bonne estimation du calcul de pose est primordiale dans ce genre d'application puisque c'est elle qui va influencer sur la reconstruction du MNT de l'environ-

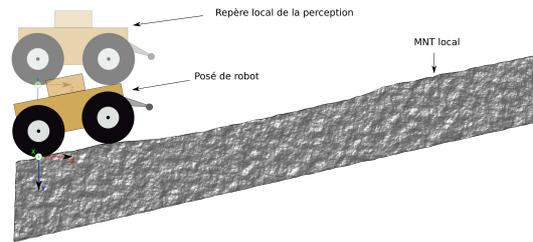


FIGURE 4.56 – L'utilisation du posé de robot pour prendre en compte le principe que le robot roule sur le sol.

nement et par conséquent joue sur la sécurité du véhicule et de ses occupants. Comme nous en avons parlé dans le chapitre 1 l'interaction entre perception et guidage doit être constante. Cette information n'est pas, à l'heure actuelle, utilisée au maximum dans notre système. Par exemple lorsque le système de navigation utilise les informations de la carte il calcule les effets de l'environnement sur les paramètres d'orientation du véhicule. Ces calculs sont ensuite utilisés pour évaluer les possibilités de renversement. Ces informations peuvent être utilisées par le système de perception pour orienter l'évolution du système afin de focaliser plus pertinemment les ressources. Aussi, ces informations sur le posé de robot peuvent corriger l'état du véhicule calculé par l'odométrie visuelle en regardant localement la différence entre la pose du robot théorique et le posé de robot calculé par le système de guidage. Cette information du posé de robot va agir comme une observation de l'odométrie visuelle pour lui indiquer qu'il roule sur le sol (un schéma explicatif montre ce principe en figure 4.56).

Le système de guidage a besoin de maintenir une certaine précision pour assurer son intégrité physique dans le couloir de navigation. Pour cela, les informations du GPS ont été intégrées dans le système de perception afin de corriger la dérive occasionnée par l'odométrie visuelle. Cependant ces corrections ne sont pas complètes puisque les informations du GPS sont limitées (pas de mesure du tangage ou du roulis). Pour cela l'utilisation d'un SIG par exemple permettrait d'allier à la fois des données intègres et la possibilité de corriger tous les paramètres de la pose du véhicule. Par exemple, si plusieurs amers référencés apparaissent dans une image caméra une technique de triangulation peut être utilisée. Cette estimation pourrait remplacer l'utilisation d'un GPS en fournissant des données plus complètes sur l'état du véhicule.

La compréhension de l'environnement passe aussi par la connaissance de la nature du sol. Aussi, à travers les informations de textures, l'environnement local du véhicule pourrait être labellisé comme correspondant à un revêtement particulier. Cette information supplémentaire pourrait donc être utilisée par le système de guidage pour évaluer de façon plus juste la navigabilité de l'environnement au vu

de son état. Par exemple un sol glissant pourra limiter les capacités d'évitement du robot et ainsi limiter sa vitesse alors qu'un sol adhérent autorisera des manœuvres et des changements de vitesse plus rapides. Des travaux existent sur la caractérisation du type de sol comme par exemple les travaux de Tessier ([Tessier, 2004]) où il cherche à classifier le sol en utilisant les informations d'une caméra. Plus récemment, Renner ([Renner *et al.*, 2009]) a recensé les différentes sortes de classifications comme la végétation ou des étendues d'eau. Ces classifications ont été expérimentées sur le terrain par leur robot d'exploration RAVON grâce à différents capteurs dont une caméra ou un télémètre. Cette perspective est indispensable si nous envisageons une évolution à vitesse élevée du véhicule autonome.

Annexe A

Annexe

A.1 Calcul des jacobiennes des observations

Dans le chapitre 2 nous avons présenté les méthodes pour calculer les jacobiennes des fonctions d'observation sans donner les résultats complets de tous les calculs : nous allons les présenter ici. Dans un premier temps, nous avons posé $\frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} = \left(\frac{\partial \mathbf{R}_{(v-m)}}{\partial \alpha_v} \right)^T = \mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v))^T$: vérifions cette formule.

Tout d'abord rappelons que $\mathbf{R}_{(v-m)} = \mathbf{R}_y(\beta_v) \cdot \mathbf{R}_x(\alpha_v) \cdot \mathbf{R}_z(\gamma_v)$. Nous avons donc $\mathbf{R}_{(v-m)}^T = \mathbf{R}_z(\gamma_v)^T \cdot \mathbf{R}_x(\alpha_v)^T \cdot \mathbf{R}_y(\beta_v)^T$.

La dérivé partielle de la matrice $\mathbf{R}_{(v-m)}^T$ par rapport au tangage vaut donc :

$$\frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} = \frac{\partial \mathbf{R}_z^T(\gamma_v) \cdot \mathbf{R}_x^T(\alpha_v) \cdot \mathbf{R}_y^T(\beta_v)}{\partial \alpha_v}$$

Les matrices $\mathbf{R}_z^T(\gamma_v)$ et $\mathbf{R}_y^T(\beta_v)$ peuvent être sorties de la dérivée puisqu'elles sont constantes par rapport à α_v .

$$\frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} = \mathbf{R}_z^T(\gamma_v) \cdot \frac{\partial \mathbf{R}_x^T(\alpha_v)}{\partial \alpha_v} \cdot \mathbf{R}_y^T(\beta_v)$$

Or, la dérivée d'une matrice transposée quelconque \mathbf{A} par rapport à une variable x est égale à la transposée de la dérivée de cette matrice par rapport à x :

$$\frac{\partial \mathbf{A}^T}{\partial x} = \left(\frac{\partial \mathbf{A}}{\partial x} \right)^T$$

Donc nous obtenons :

$$\frac{\partial \mathbf{R}_{(v-m)}^T}{\partial \alpha_v} = \mathbf{R}_z^T(\gamma_v) \cdot \left(\frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \right)^T \cdot \mathbf{R}_y^T(\beta_v)$$

Enfin nous trouvons :

$$\mathbf{R}_z^T(\gamma_v) \cdot \left(\frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \right)^T \cdot \mathbf{R}_y^T(\beta_v) = \left(\mathbf{R}_y(\beta_v) \cdot \left(\frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \right) \cdot \mathbf{R}_z(\gamma_v) \right)^T = \left(\frac{\partial \mathbf{R}_{(v-m)}}{\partial \alpha_v} \right)^T$$

Observation cartésienne

En observation cartésienne, tous les résultats des dérivées partielles n'ont pas été écrits. Pour rappel, la fonction d'observation est :

$$u = \frac{\vec{K}_1 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{\vec{K}_3 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} = \frac{N_u}{D_{uv}}$$

$$v = \frac{\vec{K}_2 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{\vec{K}_3 \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} = \frac{N_v}{D_{uv}}$$

Où K_i représente la $i^{\text{ème}}$ ligne de \mathbf{K} .

La jacobienne de la fonction d'observation est :

$$\mathbf{J}_{uv} = \begin{pmatrix} \frac{\partial u}{\partial X_v} & \frac{\partial u}{\partial Y_v} & \frac{\partial u}{\partial Z_v} & \frac{\partial u}{\partial \alpha_v} & \frac{\partial u}{\partial \beta_v} & \frac{\partial u}{\partial \gamma_v} & \cdots & \frac{\partial u}{\partial x_i} & \frac{\partial u}{\partial y_i} & \frac{\partial u}{\partial z_i} \\ \frac{\partial v}{\partial X_v} & \frac{\partial v}{\partial Y_v} & \frac{\partial v}{\partial Z_v} & \frac{\partial v}{\partial \alpha_v} & \frac{\partial v}{\partial \beta_v} & \frac{\partial v}{\partial \gamma_v} & \cdots & \frac{\partial v}{\partial x_i} & \frac{\partial v}{\partial y_i} & \frac{\partial v}{\partial z_i} \end{pmatrix}$$

Jacobienne de la position

Les résultats des calculs des dérivées partielles par rapport à la position du véhicule sont montrés ici.

$$\frac{\partial u}{\partial X_v} = K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial u}{\partial Y_v} = K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

$$\frac{\partial u}{\partial Z_v} = K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

$$\frac{\partial v}{\partial X_v} = K_v \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial v}{\partial Y_v} = K_v \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

$$\frac{\partial v}{\partial Z_v} = K_v \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

$$\text{avec } K_u = \frac{D_{uv}\vec{K}_1 - N_u\vec{K}_3}{D_{uv}^2} \text{ et } K_v = \frac{D_{uv}\vec{K}_2 - N_v\vec{K}_3}{D_{uv}^2}.$$

Jacobienne de l'orientation

Les résultats des calculs des dérivées partielles par rapport à l'orientation du véhicule sont montrés ici.

$$\begin{aligned} \frac{\partial u}{\partial \alpha_v} &= K_u \mathbf{R}_{(c-v)}^T \left[\mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v) \right]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial u}{\partial \beta_v} &= K_u \mathbf{R}_{(c-v)}^T \left[\mathbf{R}_z(\gamma_v) \mathbf{R}_x(\alpha_v) \frac{\partial \mathbf{R}_y(\beta_v)}{\partial \beta_v} \right]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial u}{\partial \gamma_v} &= K_u \mathbf{R}_{(c-v)}^T \left[\frac{\partial \mathbf{R}_z(\gamma_v)}{\partial \gamma_v} \mathbf{R}_x(\alpha_v) \mathbf{R}_y(\beta_v) \right]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial v}{\partial \alpha_v} &= K_v \mathbf{R}_{(c-v)}^T \left[\mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v) \right]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial v}{\partial \beta_v} &= K_v \mathbf{R}_{(c-v)}^T \left[\mathbf{R}_z(\gamma_v) \mathbf{R}_x(\alpha_v) \frac{\partial \mathbf{R}_y(\beta_v)}{\partial \beta_v} \right]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial v}{\partial \gamma_v} &= K_v \mathbf{R}_{(c-v)}^T \left[\frac{\partial \mathbf{R}_z(\gamma_v)}{\partial \gamma_v} \mathbf{R}_x(\alpha_v) \mathbf{R}_y(\beta_v) \right]^T (\vec{P}_n - \vec{T}_{(v-m)}) \end{aligned}$$

Jacobienne du point 3D

Les résultats des calculs des dérivées partielles par rapport à la position du point 3D sont montrés ici.

$$\begin{aligned}
\frac{\partial u}{\partial P_{xi}} &= K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
\frac{\partial u}{\partial P_{yi}} &= K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\
\frac{\partial u}{\partial P_{zi}} &= K_u \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
\frac{\partial v}{\partial P_{xi}} &= K_v \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
\frac{\partial v}{\partial P_{yi}} &= K_v \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\
\frac{\partial v}{\partial P_{zi}} &= K_v \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
\end{aligned}$$

Observation sphérique

En observation sphérique, tout les résultats des dérivés partielles n'ont pas été écrits. Pour rappel, la fonction d'observation est :

$$\begin{aligned}
\begin{pmatrix} \phi \\ \psi \end{pmatrix} &= \begin{pmatrix} \arctan \left(\frac{x_c}{z_c} \right) \\ \arctan \left(\frac{y_c}{z_c} \right) \end{pmatrix} = \begin{pmatrix} \arctan \left(\frac{(1 \ 0 \ 0) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{(0 \ 0 \ 1) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} \right) \\ \arctan \left(\frac{(0 \ 1 \ 0) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})}{(0 \ 0 \ 1) \mathbf{R}_{(c-v)}^T (\mathbf{R}_{(v-m)}^T (\vec{P}_n - \vec{T}_{(v-m)}) - \vec{T}_{(c-v)})} \right) \end{pmatrix} \\
&= \begin{pmatrix} \arctan \left(\frac{N_\phi}{D_{\phi\psi}} \right) \\ \arctan \left(\frac{N_\psi}{D_{\phi\psi}} \right) \end{pmatrix} \tag{A.1}
\end{aligned}$$

Jacobienne de la position

Les résultats des calculs des dérivés partielles par rapport à la position du véhicule sont montrés ici.

$$\frac{\partial \phi}{\partial X_v} = K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned}\frac{\partial \phi}{\partial Y_v} &= K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \\ \frac{\partial \phi}{\partial Z_v} &= K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \\ \frac{\partial \psi}{\partial X_v} &= K_\psi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \\ \frac{\partial \psi}{\partial Y_v} &= K_\psi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \\ \frac{\partial \psi}{\partial Z_v} &= K_\psi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}\end{aligned}$$

$$\text{avec } K_\phi = \frac{D_{\phi\psi} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} - N_\phi \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}}{D_{\phi\psi}^2 + N_\phi^2} \text{ et } K_\psi = \frac{D_{\phi\psi} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} - N_\psi \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}}{D_{\phi\psi}^2 + N_\psi^2}$$

Jacobienne de l'orientation

Les résultats des calculs des dérivées partielles par rapport à l'orientation du véhicule sont montrés ici.

$$\begin{aligned}\frac{\partial \phi}{\partial \alpha_v} &= K_\phi \mathbf{R}_{(c-v)}^T [\mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v)]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial \phi}{\partial \beta_v} &= K_\phi \mathbf{R}_{(c-v)}^T [\mathbf{R}_z(\gamma_v) \mathbf{R}_x(\alpha_v) \frac{\partial \mathbf{R}_y(\beta_v)}{\partial \beta_v}]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial \phi}{\partial \gamma_v} &= K_\phi \mathbf{R}_{(c-v)}^T [\frac{\partial \mathbf{R}_z(\gamma_v)}{\partial \gamma_v} \mathbf{R}_x(\alpha_v) \mathbf{R}_y(\beta_v)]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial \psi}{\partial \alpha_v} &= K_\psi \mathbf{R}_{(c-v)}^T [\mathbf{R}_z(\gamma_v) \frac{\partial \mathbf{R}_x(\alpha_v)}{\partial \alpha_v} \mathbf{R}_y(\beta_v)]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial \psi}{\partial \beta_v} &= K_\psi \mathbf{R}_{(c-v)}^T [\mathbf{R}_z(\gamma_v) \mathbf{R}_x(\alpha_v) \frac{\partial \mathbf{R}_y(\beta_v)}{\partial \beta_v}]^T (\vec{P}_n - \vec{T}_{(v-m)}) \\ \frac{\partial \psi}{\partial \gamma_v} &= K_\psi \mathbf{R}_{(c-v)}^T [\frac{\partial \mathbf{R}_z(\gamma_v)}{\partial \gamma_v} \mathbf{R}_x(\alpha_v) \mathbf{R}_y(\beta_v)]^T (\vec{P}_n - \vec{T}_{(v-m)})\end{aligned}$$

Jacobienne du point 3D

Les résultats des calculs des dérivées partielles par rapport à la position du point 3D sont montrés ici.

$$\frac{\partial \phi}{\partial P_{xi}} = K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned} \frac{\partial \phi}{\partial P_{yi}} &= K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ \frac{\partial \phi}{\partial P_{zi}} &= K_\phi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ \frac{\partial \psi}{\partial P_{xi}} &= K_\psi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ \frac{\partial \psi}{\partial P_{yi}} &= K_\psi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ \frac{\partial \psi}{\partial P_{zi}} &= K_\psi \mathbf{R}_{(c-v)}^T \mathbf{R}_{(v-m)}^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

A.2 Calibration du système Caméra-Télemètre

Pour pouvoir utiliser les données du télemètre dans la caméra elles doivent être observées dans cette dernière. Étant donné que le laser du télemètre n'est pas visible dans l'espace de couleur de la caméra on ne peut pas observer les données directement. Cependant si les coordonnées des impacts laser étaient connues directement dans le repère de la caméra il suffirait de les projeter pour connaître les coordonnées (u_l, v_l) des impacts dans l'image comme si c'était la caméra elle-même qui le faisait. Le lien qui va du télemètre à la caméra étant considéré comme indéformable, il nous suffit de calculer la transformation rigide entre le télemètre et la caméra ; c'est la calibration.

Dans notre cas de calibration entre une caméra et un télemètre, le principe est d'observer à travers les deux capteurs une information commune dont les paramètres sont connus. La description de la calibration d'un tel système a été proposée par [Trassoudaine, 1993]. Ce principe a ensuite été utilisé par [Pujas et Aldon, 1996] et [Chateau, 1999] pour effectuer cette calibration à l'aide d'une mire de forme cubique. D'autres méthodes ont été proposées dans la littérature (par exemple [Zhang et Pless, 2004] propose une calibration basée sur plusieurs jeux de données d'une mire planaire) mais nous avons choisi de mettre en œuvre la solution présentée dans les travaux de [Chateau, 1999]. La méthode consiste à utiliser un repère accessible aux deux capteurs et visible sous forme d'une mire cubique (la figure A.1 montre la configuration de la calibration). La calibration se fait en plusieurs étapes :

- La première étape consiste à calculer la matrice de passage $\mathbf{M}_{(m-c)}$ de la

mire vers la caméra. Pendant cette étape nous calculons aussi le déplacement de la mire durant toute la séquence. Ces informations sont nécessaires pour la deuxième étape. Pour cela un filtre de Kalman étendu est utilisé. Le vecteur d'état contient les 6 paramètres de pose de la caméra ainsi que les positions des points de la mire utilisés comme amers. Les coordonnées de ces points sont supposées connues parfaitement (les incertitudes associées aux amers sont nulles). L'observation de ces points de contrôle dans la séquence d'images nous donne le déplacement de la mire par rapport à la caméra (qui est l'opposé du déplacement de la caméra par rapport à la mire).

- La deuxième étape consiste à reconstruire la mire en 3D avec l'ensemble des données télemètre et du déplacement de la mire connu grâce à la première étape. Ensuite, il faut déterminer les équation des 3 plans qui composent la mire. Nous obtenons des équations de la forme $Plan_i = a_i.x + b_i.y + c_i.z + d_i$ (avec $i \in \{1..3\}$) en s'assurant que les vecteurs $(a_i \ b_i \ c_i)$ soit de norme unitaire. En mettant bout à bout les équations des 3 plans qui forment la mire nous obtenons l'équation A.2.

$$\begin{aligned}
 Plan_1 &= a_1.X + b_1.Y + c_1.Z + d_1 \\
 Plan_2 &= a_2.X + b_2.Y + c_2.Z + d_2 \\
 Plan_3 &= a_3.X + b_3.Y + c_3.Z + d_3 \\
 Ajout &= 0.X + 0.Y + 0.Z + 1
 \end{aligned} \tag{A.2}$$

De plus, comme la forme de la mire est semi-cubique, les vecteurs $(a_1 \ a_2 \ a_3)$, $(b_1 \ b_2 \ b_3)$ et $(c_1 \ c_2 \ c_3)$ sont aussi de norme unitaire. La sous-matrice $(3, 3)$ formée par les coefficients a_i , b_i et c_i (avec $i \in 1..3$) a donc un déterminant égal à 1.

$$\begin{vmatrix}
 a_1 & b_1 & c_1 \\
 a_2 & b_2 & c_2 \\
 a_3 & b_3 & c_3
 \end{vmatrix} = 1$$

La matrice $(4, 4)$ formée par ce système est la matrice de passage de la mire vers le télemètre $M_{(t-m)}$.

- Finalement il suffit d'effectuer le produit matriciel des deux matrices $M_{(m-c)}$ et $M_{(t-m)}$ pour obtenir la matrice de passage du télemètre vers la caméra $M_{(t-c)}$ (équation A.3).

$$M_{(t-c)} = M_{(m-c)}.M_{(t-m)} \tag{A.3}$$

Nous utilisons aussi cette fonctionnalité pour la reconstruction du MNT.

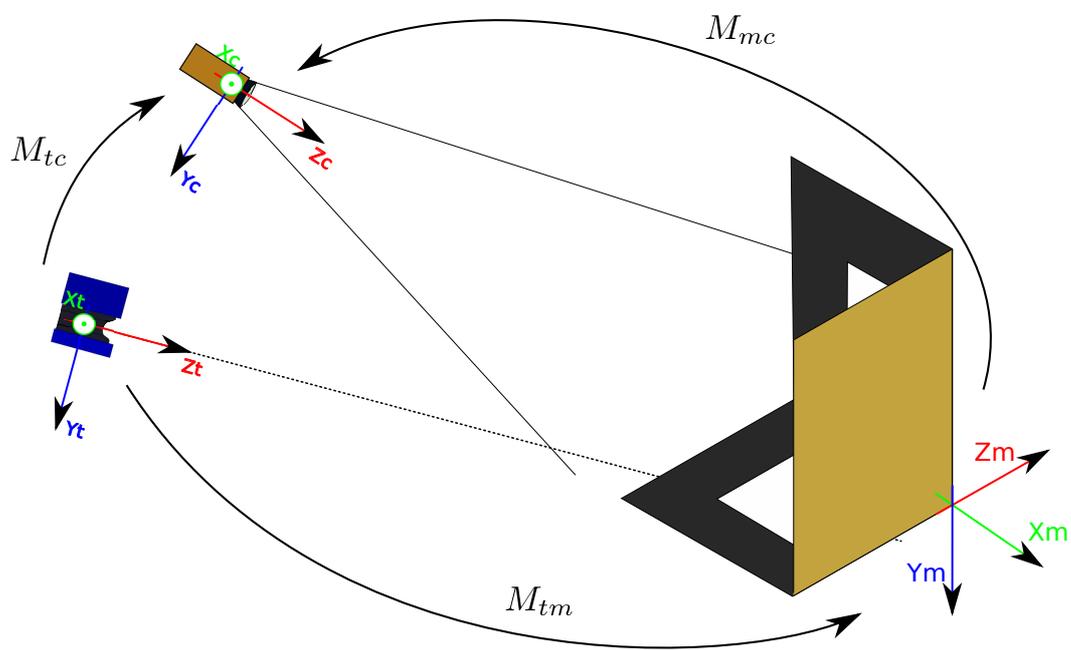


FIGURE A.1 – La calibration : passer par un repère tiers

Bibliographie

- [Araujo *et al.*, 1998] ARAUJO, H., L.CARCERONI, R. et CHRISTOPHER.M.BROWN (1998). A fully projective formulation to improve the accuracy of lowe'spose estimation algorithm. *Computer Vision and Image Understanding*, 70(2):227–238.
- [Arulampalam *et al.*, 2001] ARULAMPALAM, S., MASKELL, S., GORDON, N. et CLAPP, T. (2001). A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188.
- [Bailey et Durrant-Whyte, 2006] BAILEY, T. et DURRANT-WHYTE, H. (2006). Simultaneous localization and mapping (slam) : part ii. *Robotics Automation Magazine, IEEE*, 13(3):108 –117.
- [Bailey *et al.*, 2006a] BAILEY, T., NIETO, J., GUIVANT, J., STEVENS, M. et NEBOT, E. (2006a). Consistency of the ekf-slam algorithm. *In International Conference on Intelligent Robots and Systems*.
- [Bailey *et al.*, 2006b] BAILEY, T., NIETO, J. et NEBOT, E. (2006b). Consistency of the fastslam algorithm. *In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference*, pages 424–429.
- [Barbour et Schmidt, 2001] BARBOUR, N. et SCHMIDT, G. (2001). Inertial sensor technology trends. *Sensors Journal, IEEE*, 1(4):332–339.
- [Bétaille *et al.*, 2007] BÉTAILLE, D., CHAPELON, A., LUSETTI, B., KAIS, M. et MILLESCAMPS, D. (2007). High integrity reference trajectory for benchmarking land navigation data fusion methods. *In IEEE Intelligent Vehicles Symposium*, pages 346–351, Istanbul, Turkey.
- [Borenstein et Feng, 1996] BORENSTEIN, J. et FENG, L. (1996). Gyrodometry : A new method for combining data from gyros and odom etry in mobile robots. *In Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 423–428.
- [Broggi *et al.*, 2005] BROGGI, A., CARAFFI, C., FEDRIGA, R. I. et GRISLERI, P. (2005). Obstacle detection with stereo vision for off-road vehicle navigation. *In Procs. Intl. IEEE Wks. on Machine Vision for Intelligent Vehicles*, page 65.

- [Bujnak *et al.*, 2008] BUJNAK, M., KUKELOVA, Z. et PAJDLA, T. (2008). A general solution to the p4p problem for camera with unknown focal length.
- [Campion *et al.*, 1996] CAMPION, G., BASTIN, G. et D'ANDRÉA-NOVEL, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62.
- [Carlson, 2010] CARLSON, J. (2010). *Mapping Large, Urban Environments with GPS-Aided SLAM*. Thèse de doctorat, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- [Charmette *et al.*, 2010] CHARMETTE, B., ROYER, E. et CHAUSSE, F. (2010). Efficient planar features matching for robot localization using gpu. *In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference*, pages 16–23.
- [Chateau, 1999] CHATEAU, T. (1999). *Détection robuste d'interface par fusion d'informations incertaines : application à l'aide au guidage d'engines agricoles*. Thèse de doctorat, Université Blaise Pascal - Clermont Ferrand II.
- [Chli et Davison, 2008] CHLI, M. et DAVISON, A. (2008). Active matching. *In FORSYTH, D., TORR, P. et ZISSERMAN, A., éditeurs : Computer Vision - ECCV 2008*, volume 5302 de *Lecture Notes in Computer Science*, pages 72–85. Springer Berlin / Heidelberg.
- [Davison, 2003] DAVISON, A. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. *In IEEE International Conference on Computer Vision*, pages 1403–1410.
- [Delmas, 2011] DELMAS, P. (2011). *Génération active des déplacements d'un véhicule agricole dans son environnement*. Thèse de doctorat, Université Blaise Pascal - Clermont II.
- [Dickmanns *et al.*, 1993] DICKMANNNS, E., BEHRINGER, R., BRUDIGAM, C., DICKMANNNS, D., THOMANEK, F. et van HOLT, V. (1993). An all-transputer visual autobahn-autopilot/copilot. *In Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 608–615.
- [Dissanayake *et al.*, 2001] DISSANAYAKE, M., NEWMAN, P., CLARK, S., DURRANT-WHYTE, H. et CSORBA, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions*, 17(3):229–241.
- [Dubbelman *et al.*, 2007] DUBBELMAN, G., van der MARK, W., van den HEUVEL, J. C. et GROEN, F. C. A. (2007). Obstacle detection during day and night conditions using stereo vision. *In IROS'07*, pages 109–116.

- [Durrant-Whyte et Bailey, 2006] DURRANT-WHYTE, H. et BAILEY, T. (2006). Simultaneous localization and mapping (slam) : part i. *Robotics Automation Magazine, IEEE*, 13(2):99–110.
- [Fischler et Bolles, 1981] FISCHLER, M. A. et BOLLES, R. C. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.
- [Forsyth et Ponce, 2002] FORSYTH, D. A. et PONCE, J. (2002). *Computer Vision : A Modern Approach*. Prentice Hall.
- [Fremont, 2009] FREMONT, V. (2009). Odométrie 3d vision/lidar pour les véhicules intelligents. *In Journées Nationales de la Recherche en Robotique, JNRR'09*, Neuvy-sur-Barangeon France.
- [Féraud *et al.*, 2010] FÉRAUD, T., CHECCHIN, P. et CHAPUIS., R. (2010). Véhicules communicants en convoi et localisation par vision monoculaire. Caen - FRANCE. 17ème congrès francophone AFRIF-AFIA, Reconnaissance des Formes et Intelligence Artificielle. RFIA 2010.
- [Gibbens *et al.*, 2000] GIBBENS, P., DISSANAYAKE, G. et DURRANT-WHYTE, H. (2000). A closed form solution to the single degree of freedom simultaneous localisation and map building (slam) problem. *In Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, pages 191–196 vol.1.
- [Haralick *et al.*, 1994] HARALICK, R. M., LEE, C.-N., OTTENBERG, K. et NÖLLE, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. Comput. Vision*, 13(3):331–356.
- [Harris et Stephens, 1988] HARRIS, C. et STEPHENS, M. (1988). A combined corner and edge detection. *In Proceedings of The Fourth Alvey Vision Conference*, pages 147–151.
- [Hartley, 1995] HARTLEY, R. (1995). In defence of the 8-point algorithm. *In Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 1064–1070.
- [Huang *et al.*, 2008] HUANG, G., MOURIKIS, A. et ROUMELIOTIS, S. (2008). Analysis and improvement of the consistency of extended kalman filter based slam. *In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 473–479.
- [Jensfelt et Kristensen, 2001] JENSFELT, P. et KRISTENSEN, S. (2001). Active global localisation for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5):748–760.
- [Joly, 2010] JOLY, C. (2010). *Contributions aux méthodes de localisation et cartographie simultanées par vision omnidirectionnelle*. Thèse de doctorat, École nationale supérieure des mines de Paris.

- [Julier et Uhlmann, 2001] JULIER, S. J. et UHLMANN, J. K. (2001). A counter example to the theory of simultaneous localization and map building. *In IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4238–4243.
- [Jung et Lacroix, 2003] JUNG, I.-K. et LACROIX, S. (2003). High resolution terrain mapping using low altitude aerial stereo imagery. *In International Conference on Computer Vision, Nice (France)*.
- [Kelly et Stentz, 1997a] KELLY, A. et STENTZ, A. (1997a). Analysis of requirements for high speed rough terrain autonomous mobility. i. throughput and response. *In Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference*, volume 4, pages 3318 –3325 vol.4.
- [Kelly et Stentz, 1997b] KELLY, A. et STENTZ, A. (1997b). Analysis of requirements for high speed rough terrain autonomous mobility. ii. resolution and accuracy. *In Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference*, volume 4, pages 3326 –3333 vol.4.
- [Konolige et Agrawal, 2008] KONOLIGE, K. et AGRAWAL, M. (2008). Frameslam : From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions*, 24(5):1066 –1077.
- [Kwok et al., 2005] KWOK, N. M., DISSANAYAKE, G. et HA, Q. P. (2005). Bearing-only slam using a sprt based gaussian sum filter. *In Proceedings of the 2005 IEEE International Conference on Robotics and Automation*.
- [Lacroix et al., 2002] LACROIX, S., JUNG, I.-K. et MALLET, A. (2002). Digital elevation map building with low altitude stereo imagery. *Robotics and Autonomous Systems*, 41(2):119–127.
- [Laneurit, 2006] LANEURIT, J. (2006). *Perception multisensorielle pour la localisation de robot mobile en environnement extérieur, application aux véhicules routiers*. Thèse de doctorat, Université Blaise Pascal - Clermont Ferrand II.
- [Lemaire et al., 2007] LEMAIRE, T., BERGER, C., JUNG, I.-K. et LACROIX, S. (2007). Vision-based slam : stereo and monocular approaches. *International Journal on Computer Vision*, 74(3):343–364.
- [Lemaire et al., 2005] LEMAIRE, T., LACROIX, S. et SOLÀ, J. (2005). A practical 3d bearing only slam algorithm. *IEEE International Conference on Intelligent Robots and Systems*, pages 2575–2562.
- [Lenain et al., 2006] LENAIN, R., THUILOT, B., CARIOU, C. et MARTINET, P. (2006). High accuracy path tracking for vehicles in presence of sliding : Application to farm vehicle automatic guidance for agricultural tasks. *Autonomous Robots*, 21(1):79–97.
- [Li et Hartley, 2006] LI, H. et HARTLEY, R. (2006). Five-point motion estimation made easy. *In ICPR '06 : Proceedings of the 18th International Conference on*

- Pattern Recognition*, pages 630–633, Washington, DC, USA. IEEE Computer Society.
- [Longuet-Higgins, 1981] LONGUET-HIGGINS, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. In FISCHLER, M. A. et FIRSCHEIN, O., éditeurs : *Readings in Computer Vision : Issues, Problems, Principles, and Paradigms*, pages 61–62. Kaufmann, Los Altos, CA.
- [Lowe, 1999] LOWE, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision - Volume 2 - Volume 2, ICCV '99*, pages 1150–, Washington, DC, USA. IEEE Computer Society.
- [Malartre *et al.*, 2010] MALARTRE, F., DELMAS, P., CHAPUIS, R. et DEBAIN, C. (2010). Real-time dense digital elevation map estimation using laserscanner and camera slam process. *ICARCV2010, 11th International Conference on Control, Automation, Robotics and Vision*.
- [Malartre *et al.*, 2009a] MALARTRE, F., FERAUD, T., CHAPUIS, R. et DEBAIN, C. (2009a). Geometric reconstruction of a natural environment by fusing camera and rangefinder. In *International Symposium « Innovation Technologiques et Systèmes de Transport ITT'09 »*, ENSTA ParisTech - PARIS 15 ème - FRANCE.
- [Malartre *et al.*, 2009b] MALARTRE, F., FERAUD, T., DEBAIN, C. et CHAPUIS, R. (2009b). Digital elevation map estimation by vision-lidar fusion. *IEEE International Conference on Robotics and Biomimetics (ROBIO09)*.
- [Montemerlo *et al.*, 2002] MONTEMERLO, M., THRUN, S., KOLLER, D. et WEGBREIT, B. (2002). Fastslam : A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada. AAAI.
- [Montemerlo *et al.*, 2003] MONTEMERLO, M., THRUN, S., KOLLER, D. et WEGBREIT, B. (2003). Fastslam 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1151–1156.
- [Mouragnon *et al.*, 2006] MOURAGNON, E., LHUILLIER, M., M., D., DEKEYSER, F. et SAYD, P. (2006). Real time localization and 3d reconstruction. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 363–370, Washington, DC, USA. IEEE Computer Society.
- [Mourllion *et al.*, 2006] MOURLLION, B., GRUYER, D. et LAMBERT, A. (2006). Variance behavior and signification in probabilistic framework applied to vehicle localization. In *IEEE/ITSS International Conference on Intelligent Vehicle*, pages 294–299, Tokyo, Japan.

- [Nistér, 2004] NISTÉR, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777.
- [Nüchter *et al.*, 2005] NÜCHTER, A., LINGEMANN, K., HERTZBERG, J. et SURMANN, H. (2005). Heuristic-based laser scan matching for outdoor 6d slam. *In Advances in artificial intelligence. 28th annual German Conf. on AI*, pages 304–319.
- [Ohta et Kanade, 1985] OHTA, Y. et KANADE, T. (1985). Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-7(1):139–154.
- [Press *et al.*, 1992] PRESS, W., FLANNERY, B., TEUKOLSKY, S. et VETTERLING, W. (1992). *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press.
- [Pujas et Aldon, 1996] PUJAS, P. et ALDON, M. (1996). Etalonnage d'un système de fusion caméra-télemètre pour la fusion multi-sensorielle. *In AFCET RFIA'96*.
- [Quan et Lan, 1999] QUAN, L. et LAN, Z. (1999). Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:774–780.
- [Renner *et al.*, 2009] RENNER, A., FÖHST, T. et BERNS, K. (2009). Perception of environment properties relevant for off-road navigation. *In DILLMANN, R., BEYERER, J., STILLER, C., ZÖLLNER, J. M. et GINDELE, T., éditeurs : Autonome Mobile Systeme 2009*, pages 201–208, Karlsruhe, Germany. Springer Berlin / Heidelberg. ISBN 978-3-642-10284-4.
- [Royer *et al.*, 2006] ROYER, E., LHUILLIER, M., DHOME, M. et LAVEST, J.-M. (2006). Localisation par vision monoculaire pour la navigation autonome : précision et stabilité de la méthode. *RFIA, 15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*.
- [Rusinkiewicz et Levoy, 2001] RUSINKIEWICZ, S. et LEVOY, M. (2001). Efficient variants of the icp algorithm. *3D Digital Imaging and Modeling, International Conference*, 0:145.
- [Schleicher *et al.*, 2009] SCHLEICHER, D., BERGASA, L. M., OCANA, M., BAREA, R. et LOPEZ, M. E. (2009). Real-time hierarchical outdoor slam based on stereovision and gps fusion. *Intelligent Transportation Systems, IEEE Transactions*, 10(3):440–452.
- [Stewénius *et al.*, 2006] STEWÉNIUS, H., ENGELS, C. et NISTÉR, D. (2006). Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294.
- [Tessier, 2004] TESSIER, C. (2004). Détection de type de sol par analyse de texture pour le guidage de véhicules. Rapport de DEA, Composants et systèmes pour le traitement de l'information.

- [Tessier, 2007] TESSIER, C. (2007). *Système de localisation basé sur une stratégie de perception cognitive appliqué à la navigation autonome d'un robot mobile*. Thèse de doctorat, Université Blaise Pascal - Clermont Ferrand II.
- [Tessier et al., 2006] TESSIER, C., CARIOU, C., DEBAIN, C., CHAPUIS, R., CHAUSSE, F. et ROUSSET, C. (2006). A real-time, multi-sensor architecture for fusion of delayed observations : Application to vehicle localisation. In *9th International IEEE Conference on Intelligent Transportation Systems*, pages 1316–1321, Toronto, Canada.
- [Thrun et al., 2006] THRUN, S., MONTEMERLO, M., DAHLKAMP, H., STAVENS, D., ARON, A., DIEBEL, J., FONG, P., GALE, J., HALPENNY, M., LAU, K., OAKLEY, C., PALATUCCI, M., PRATT, V., STANG, P., STROHB, S., DUPONT, C., erik JENDROSSEK, L., KOELEN, C., MARKEY, C., RUMMEL, C., NIEKERK, J. V., JENSEN, E., BRADSKI, G., DAVIES, B., ETTINGER, S., KAEHLER, A., NEFIAN, A. et MAHONEY, P. (2006). The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23:661–692.
- [Thuilot et al., 2004] THUILOT, B., BOM, J., MARMOITON, F. et MARTINET, P. (2004). Accurate automatic guidance of an urban vehicle relying on a kinematic gps sensor. In *Symposium on Intelligent Autonomous Vehicles IAV04*.
- [Trassoudaine, 1993] TRASSOUDAINE, L. (1993). *Solution multisensorielle temps réel pour la détection d'obstacles sur la route*. Thèse de doctorat, Université Blaise Pascal - Clermont Ferrand.
- [Urmson et al., 2008] URMSON, C., ANHALT, J., BAE, H., BAGNELL, J. A. D., BAKER, C., BITTNER, R. E., BROWN, T., CLARK, M. N., DARMS, M., DEMITRISH, D., DOLAN, J. M., DUGGINS, D., FERGUSON, D., GALATALI, T., GEYER, C. M., GITTLEMAN, M., HARBAUGH, S., HEBERT, M., HOWARD, T., KOLSKI, S., LIKHACHEV, M., LITKOUHI, B., KELLY, A., MCNAUGHTON, M., MILLER, N., NICKOLAOU, J., PETERSON, K., PILNICK, B., RAJKUMAR, R., RYBSKI, P., SADEKAR, V., SALESKY, B., SEO, Y.-W., SINGH, S., SNIDER, J. M., STRUBLE, J. C., STENTZ, A. T., TAYLOR, M., WHITTAKER, W. R. L., WOLKOWICKI, Z., ZHANG, W. et ZIGLAR, J. (2008). Autonomous driving in urban environments : Boss and the urban challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(1):425–466.
- [Velodyne, 2007] VELODYNE (2007). High definition lidar. Datasheet for Velodyne HDL-64E S2. <http://www.velodyne.com/lidar/hdlproducts/hdl64e.aspx>.
- [Zhang et Pless, 2004] ZHANG, Q. et PLESS, R. (2004). Extrinsic calibration of a camera and laser range finder (improves camera calibration). volume 3, pages 2301 – 2306 vol.3.