



HAL
open science

Localisation d'un robot mobile autonome en environnements naturels

Anthony Mallet

► **To cite this version:**

Anthony Mallet. Localisation d'un robot mobile autonome en environnements naturels. Automatique / Robotique. Institut National Polytechnique de Toulouse - INPT, 2001. Français. NNT: . tel-00131779

HAL Id: tel-00131779

<https://theses.hal.science/tel-00131779>

Submitted on 19 Feb 2007

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.

Thèse

préparée au
Laboratoire d'Analyse et d'Architecture des Systèmes

en vue de l'obtention du titre de
Docteur de l'Institut National Polytechnique de Toulouse
Systèmes Automatiques

LOCALISATION D'UN ROBOT MOBILE AUTONOME EN ENVIRONNEMENTS NATURELS MÉTHODES ET INTÉGRATION

Anthony Mallet

Soutenue le 2 juillet 2001 devant le jury

Président	A. AYACHE
Rapporteurs	M. HEBERT M. RICHTIN P. RIVES
Examineurs	N. BURGAUD R. CHATILA
Directeur de thèse	S. LACROIX

Avant propos

Ce manuscrit présente une photographie de l'état des travaux que j'ai mené pendant trois années et quelques mois au LAAS ; les quelques lignes de cet avant-propos sont, elles, rédigées dans le calme qui suit la période dense de la soutenance. Je crois, avec le recul, qu'aucun des chapitres ne me satisfait vraiment. De fait, les thèses ont malheureusement une durée relativement inextensible, et quasiment la totalité des points abordés nécessiteraient à eux seuls une quantité de travail non négligeable pour être considérés comme pleinement achevés.

Au moment de la rédaction, j'ai choisi de ne pas censurer les voies de recherche que j'avais en tête. Je suis conscient que pour le lecteur, la présentation de ces pistes sans conclusions tranchées provoquera un sentiment de travail inachevé. Néanmoins, plusieurs résultats tangibles sont présentés et, même s'ils sont parfois résumés en quelques lignes ou paragraphes, leur obtention a, dans l'ensemble, nécessité les trois années de ma thèse.

J'ai attaché une grande importance à ce que ce travail ne soit pas mené dans l'isolement. D'une part, j'espère qu'il a profité et qu'il profitera à l'équipe de roboticiens du laboratoire. D'autre part, cette même équipe a également participé, et fortement contribué, à sa réalisation. Je pense en particulier à Simon, directeur de ma thèse ; le travail qu'il a effectué pour, et avec moi, est inestimable. L'attention qu'il a manifesté, l'implication qu'il a eue, la volonté dont il a fait preuve pour me tirer vers l'échéance de la soutenance sont, je crois, d'une rare intensité. Le travail décrit ici reflète donc, à chaque phrase, ligne, ou mot, cette collaboration particulièrement enrichissante pour moi. Je ne résiste pas à la tentation de citer ici une anecdote, un clin d'oeil amical du hasard : les initiales de nos noms respectifs forment, dans le bon ordre, le mot clef *SLAM*. Comment, alors, travailler sur un autre sujet de thèse que celui-ci ?

Je voudrais également remercier Raja, « directeur en chef » de ma thèse. À mon grand regret, notre collaboration a été plus tardive dans le déroulement de ma thèse. Néanmoins, j'espère et je crois que l'année à venir nous permettra de nous rencontrer davantage.

J'adresse également une pensée amicale à toutes les personnalités fortes et actives de RIA. En particulier à Matthieu et Sara, ou Sara et Matthieu, duo d'ingénieurs dont les compétences sont elles aussi exceptionnelles, ainsi qu'à l'incontournable Bauz et tous les gens qui « font bouger des robots ».

Et bien sûr, il m'est impossible de ne pas évoquer l'omniprésence subtile et discrète de Betty, dont l'esprit gravite certainement encore autour de ces quelques lignes, ainsi que la présence — certes moins discrète et plus conséquente — de Ben, dont les conversations auront sans aucun doute marqué ces quelques années. :-)

Hop, bonne lecture !

Table des matières

Introduction	13
1 Navigation Autonome et Localisation	17
1.1 Navigation autonome en environnements naturels	17
1.1.1 Spécificités des environnements naturels	18
1.1.2 Quelques projets et applications	19
1.1.3 Le projet EDEN au LAAS	23
1.1.4 Conclusions	25
1.2 De l'importance de la localisation	25
1.3 Localisation en environnements naturels : état de l'art	27
1.3.1 Localisation odométrique	27
1.3.2 Localisation inertielle	28
1.3.3 Localisation à partir d'images de profondeur	29
1.3.4 Localisation à partir de la vision	30
1.3.5 Localisation topologique	31
1.4 Typologie des méthodes de localisation	32
1.4.1 Trois classes de méthodes	33
1.4.2 Critères de classement	34
1.5 Outils nécessaires à la localisation	36
1.6 Conclusions	37
2 L'odométrie	39
2.1 Commande des châssis de type « Marsokhod »	40
2.2 Calcul de la position odométrique	41
2.3 Contrôle du mouvement	43
2.3.1 Contrôle de la vitesse de rotation des roues	43
2.3.2 Contrôle de la vitesse de rotation du châssis	44
2.4 Utilisation d'un compas et d'un gyromètre	45
2.4.1 Estimation de la position	45

2.4.2	Asservissement du mouvement	46
2.5	Utilisation d'inclinomètres pour une position $3-D$	48
2.6	Comportement de l'estimation de position	50
2.7	Conclusions	53
3	L'odométrie optique	55
3.1	Principe	56
3.1.1	Reconstruction tridimensionnelle	57
3.1.2	Suivi de pixels dans une séquence d'images	57
3.1.3	Prédiction de la position des pixels	59
3.1.4	Sélection des pixels	59
3.1.5	Estimation du déplacement entre deux ensembles de points $3-D$	60
3.2	Premiers résultats	61
3.2.1	Estimation de position sur <i>Lama</i>	61
3.2.2	Estimation de position pour un ballon dirigeable	64
3.3	Analyse et qualification de la méthode	65
3.3.1	Sensibilité au bruit	66
3.3.2	Précision intrinsèque	67
3.3.3	Influence de la quantité d'appariements	68
3.3.4	Influence de la vitesse du robot	69
3.3.5	Autres paramètres	69
3.4	Conclusions	70
4	Les cartes d'élévation	71
4.1	Modélisation de l'imprécision et grilles d'occupation	72
4.1.1	Un modèle de capteur pour la stéréo-vision	74
4.1.2	Approche bayésienne	75
4.1.3	Approche de Dempster-Shafer	78
4.1.4	Des grilles d'occupation aux cartes d'élévation	80
4.1.5	Une implémentation sur le robot <i>Lama</i>	81
4.2	Gestion du modèle à long terme	84
4.2.1	Carte locale	84
4.2.2	Modèle global	85
4.2.3	Déformation du modèle	86
4.3	Localisation sur des cartes d'élévation	88
4.3.1	Corrections locales de la position	88
4.3.2	Estimation de la position sur la base d'amers	90
4.4	Conclusions	92
5	Intégration logicielle	93
5.1	Une architecture pour l'autonomie	94
5.1.1	Hierarchie et modularité	94
5.1.2	Flux de contrôle, flux de données	96
5.1.3	Cas particulier du robot <i>Lama</i>	98
5.2	Intégration d'algorithmes de localisation	99
5.2.1	Graphe géométrique du robot	99
5.2.2	Étiquetage et contexte d'acquisition des données	100
5.2.3	Gestion centralisée des positions	101
5.2.4	Vers une fusion et une détection de fautes	102
5.3	Une démonstration de navigation autonome	104
5.3.1	Couche fonctionnelle	104

5.3.2	Contrôle d'exécution	106
5.4	Conclusions	106
	Conclusions	109
	Annexes	111
A	Le robot <i>Lama</i>	113
A.1	Géométrie du châssis	114
A.2	Capteurs proprioceptifs	114
A.2.1	Les capteurs de configuration	114
A.2.2	Les codeurs odométriques	114
A.2.3	Le compas magnétique	115
A.2.4	Le gyromètre	116
A.2.5	Le GPS	117
A.3	Capteurs extéroceptifs	117
A.4	Architecture informatique	118
B	Geométrie sur le robot <i>Lama</i>	119
B.1	Conventions de notations	119
B.2	Définition des différents repères	119
B.2.1	Les repères liés au banc stéréoscopique	119
B.2.2	Les repères liés au robot	120
B.2.3	Le repère global	120
B.3	Les paramètres perçus permettant de déterminer les repères	120
B.3.1	Les paramètres directement lus sur les instruments du robot	120
B.3.2	Les paramètres calculés par différents modules	121
B.4	Détermination des différentes transformations	122
B.4.1	Les transformations fixes	122
B.4.2	Platine vers Mat	123
B.4.3	Robot vers Robot_Red	123
B.5	Graphe géométrique	123
C	Modélisation des erreurs de la stéréo-vision	127
C.1	Principe de l'algorithme de stéréo-vision	127
C.2	Analyse empirique de l'algorithme	130
C.2.1	Méthode	130
C.2.2	Vers un modèle d'erreur	130
C.2.3	Problèmes non résolus	131

Table des figures

1	Le robot <i>Lama</i> du LAAS.	15
1.1	Exemples de scènes naturelles.	18
1.2	Le robot Adam.	23
1.3	Le robot Lama.	24
1.4	Typologie des méthodes de localisation.	34
2.1	Commande d'un châssis dépourvu de roues directionnelles.	40
2.2	Calcul de la position odométrique.	42
2.3	Conception des roues du châssis de <i>Lama</i>	43
2.4	Asservissement de type PID sur les vitesses des roues.	44
2.5	Exécution d'un cercle sans asservissement global.	45
2.6	Comparaison des positions odométriques, compas et GPS.	46
2.7	Calcul de la vitesse angulaire à partir du cap donné par le compas.	47
2.8	Vitesse angulaire globale mesurée par gyromètre.	47
2.9	Comparaison des positions odométriques et GPS avec asservissement global.	48
2.10	Calcul de la position odométrique $3-D$	49
2.11	Comportement de l'odométrie sur trois types de terrain.	50
2.12	Vitesses des roues en fonction du terrain.	52
2.13	Moyennage des vitesses des six roues.	52
2.14	Courants consommés dans les moteurs des roues.	53
3.1	Principe général de l'odométrie optique.	56
3.2	Principe de la corrélation de pixel	57
3.3	Image non structurée de points $3-D$ acquise sur le robot <i>Lama</i>	58
3.4	Exemple de suivi de pixels pour trois types de mouvement	58
3.5	Mesure de ressemblance locale calculée sur une image complète	59
3.6	Trajectoire réelle mesurée par odométrie optique, odométrie et GPS.	62

3.7	Comparaison des erreurs en translation entre odométrie optique et odométrie.	62
3.8	Erreurs de l'odométrie optique sur l'estimation des angles.	62
3.9	Histogramme des erreurs de l'odométrie et de l'odométrie optique	63
3.10	Localisation pour la construction d'un modèle $\mathcal{3-D}$ aérien.	65
3.11	Analyse du niveau de bruit pour une estimation de déplacement.	66
3.12	Comparaison entre le bruit sur les points $\mathcal{3-D}$ et la précision de la solution.	67
3.13	Influence du nombre d'appariements.	68
3.14	Quantité d'appariements effectivement utilisés pour le calcul du déplacement.	68
3.15	Influence de la vitesse du robot.	69
4.1	Projection d'un point $\mathcal{3-D}$ imprécis dans une carte d'élévation	73
4.2	Profil de terrain $\mathcal{2-D}$ simulé pour la construction d'une grille d'occupation	74
4.3	Modèle de l'incertitude sur les points issus de stéréo-vision	75
4.4	Construction d'une grille d'occupation avec une approche bayésienne. . .	77
4.5	Exemple de profil reconstruit avec le formalisme de Bayes.	77
4.6	Construction d'une grille d'occupation avec une approche de Dempster-Shafer.	79
4.7	Exemple de profil reconstruit avec le formalisme de Dempster-Shafer. . .	79
4.8	Reconstruction d'un profil de terrain à partir d'une grille d'occupation. .	80
4.9	Reconstruction d'un profil de terrain à partir d'un moyennage.	81
4.10	Reconstruction d'un profil de terrain à partir d'une interpolation.	82
4.11	Influence de l'interpolation pour la construction d'une carte d'élévation	82
4.12	Densité des images de points $\mathcal{3-D}$	82
4.13	Filtrage des données les plus incertaines	83
4.14	Carte d'élévation incluant les informations de luminance	84
4.15	Position du modèle local par rapport au robot.	85
4.16	Mise à jour d'une carte d'élévation locale.	86
4.17	Représentation schématique du modèle de cartes d'élévations	87
4.18	Déformation du modèle global.	87
4.19	Trajectoire réelle mesurée par recalage sur cartes d'élévation, odométrie et GPS.	89
4.20	Comparaison des erreurs en translation entre recalage sur cartes d'élévation et odométrie.	90
4.21	Extraction d'amers sur la base de la densité des points $\mathcal{3-D}$	91
4.22	Extraction d'amers sur la base de cartes d'élévation.	91
4.23	Processus de modélisation d'amers par maillages déformables.	92
5.1	Architecture logicielle LAAS.	94
5.2	Description des modules.	96
5.3	Robots ayant des configurations géométriques particulièrement différentes.	99
5.4	Exemple de données exportées par le module « Géométrie ».	100
5.5	Données utilisées pour l'étiquetage.	101
5.6	Gestion temporelle de la production des positions.	102
5.7	Repère virtuel permettant d'éviter des « sauts » de position.	103
5.8	Génération de déplacement élémentaires.	104
5.9	Les angles de configuration permettant le placement sur les cartes d'élévations.	105
5.10	Planification et sélection d'un sous-but.	105

A.1	Évolution du robot <i>Lama</i>	114
A.2	Angles de configuration interne du châssis de <i>Lama</i>	114
A.3	Calibrage du compas.	115
A.4	Dérive du gyromètre.	116
A.5	Densité de probabilité des erreurs commises par le GPS à l'arrêt.	117
A.6	Une image brute du banc stéréoscopique du bas.	118
A.7	Vue de la même scène	118
B.1	Les repères liés à <i>Lama</i>	121
B.2	Les angles du châssis de <i>Lama</i>	122
B.3	Les angles donnés par les inclinomètres.	123
B.4	Grammaire du fichier de description géométrique d'un robot.	124
B.5	Exemple de graphe géométrique et description textuelle associée.	124
B.6	Le graphe géométrique du robot <i>Lama</i>	126
C.1	Les différentes étapes de l'algorithme de stéréo-vision par corrélation de pixels	128
C.2	Principe de l'étape de mise en correspondance pour la stéréo-vision.	128
C.3	Quelques courbes de scores de corrélation	129
C.4	Un résultat de stéréo-vision à partir des images acquises par le banc stéréoscopique supérieur de <i>Lama</i>	130
C.5	Les deux scènes sur lesquelles une analyse statistique de la stéréo-vision a été effectuée	130
C.6	Quelques répartitions des disparités calculées sur une centaine d'images	131
C.7	Écarts types mesurés sur l'estimation de la disparité.	132
C.8	Erreurs de l'algorithme de stéréo-vision le long des lignes de gradient de profondeur.	132

Introduction

La robotique mobile autonome est un axe de recherche qui vise à donner à une *machine* la capacité de se mouvoir dans un environnement *sans assistance* ni intervention humaine. Le mot *machine* désignant ici à la fois la machine traditionnelle, qui agit de façon *tangible* dans son environnement, et l'ordinateur, qui apporte un support pour la *décision*.

Le concept fondamental de ces recherches est l'*autonomie*. Si l'on sait en effet aujourd'hui concevoir des robots mobiles et si l'on est également capable de donner à ceux-ci une structure de locomotion suffisamment puissante pour les faire évoluer dans des environnements variés et parfois difficiles, ils sont malgré tout encore peu autonomes : ils requièrent de la part de l'homme un contrôle, une attention et une expertise poussés pour effectuer des tâches utiles.

Contexte général

Quantité d'applications sont envisagées pour les robots autonomes. Ce document se place cependant dans un contexte restreint aux *environnements naturels*. Les environnements de ce type sont particuliers en ceci qu'aucune structure n'est prévue, ni envisagée, pour aider la machine. Les applications considérées par ce domaine sont par exemple l'exploration planétaire, le déminage, la surveillance ou l'intervention en milieu hostile à l'homme.

La réalisation d'un robot autonome requiert la mise en œuvre d'une vaste gamme de fonctionnalités mais la capacité de *percevoir* et *d'analyser* l'environnement est une des principales clefs de leur autonomie : elle leur permettra de *décider* des actions à entreprendre et de mener à bien des *missions* qui leur seront confiées.

En premier lieu, c'est bien sûr grâce à différents capteurs que la machine peut appréhender son environnement. Ceux-ci se répartissent en deux catégories principales : les capteurs *proprioceptifs*, qui donnent à la machine les moyens de connaître son état interne et les capteurs *extéroceptifs*, qui permettent de percevoir le monde extérieur.

Si l'utilisation des premiers est aujourd'hui relativement bien maîtrisée (accéléromètres, inclinomètres, gyroscopes, compas, centrales inertielles, odomètres, ...), l'exploitation des données fournies par les seconds (caméras vidéo, paires de caméras stéréoscopiques, télémètres laser, radars, ...) est encore récente [Lacroix 2002]. L'état de l'art en ce domaine est déjà vaste, mais se limite à une grande variété d'algorithmes dédiés, capables certes d'extraire *quelques informations* de la masse des données produites, mais en général de façon *peu fiable et limitée* à certains types de scènes.

L'analyse de l'environnement s'effectue grâce à la *modélisation* des données issues des capteurs : des représentations synthétiques, construites à partir des perceptions brutes, seront fournies aux systèmes de *raisonnement*, de *planification* et de *décision* que l'on trouvera à bord du robot. Cette étape permet au robot de manipuler ce qu'il *perçoit* et d'en extraire un *sens* pour les algorithmes qui utiliseront ces modèles.

L'absence de structure géométrique, caractéristique des environnements naturels, rend les algorithmes de modélisation difficiles. Ainsi le problème de la segmentation, c'est à dire la subdivision d'une scène en objets ou en éléments simples, reste un problème très ouvert dans ce contexte.

Quand le robot est à même d'interpréter son environnement, il peut s'y déplacer, afin d'exécuter les missions qui lui sont confiées. Dès lors, des fonctions de *localisation* deviennent indispensables : elles permettent d'estimer, à divers ordres de grandeur, la position du robot et sa situation relativement à l'environnement ou à des objectifs géographiques.

Ainsi, les modèles construits localement peuvent être agrégés en un modèle global, continu et cohérent. Cette cohérence permet d'obtenir un niveau de *raisonnement* à grande échelle et autorise la planification de trajectoires ou l'élaboration de stratégies d'exploration.

Enfin l'étape finale vers l'autonomie consiste à intégrer les diverses fonctionnalités en un système. La présence de nombreuses unités qui traitent, interagissent et communiquent de manière cohérente et coordonnée permet la réalisation d'un robot.

Motivations et structure du document

Cette thèse aborde le problème particulier de la localisation en environnements naturels et, dans une moindre mesure, celui de la modélisation.

Afin de présenter des résultats concrets et *applicables*, nous avons souhaité valider nos travaux grâce à une plateforme réelle, disponible au LAAS, et tous les développements ont été effectués sur le robot *Lama*. Grâce au travail coordonné des chercheurs et doctorants travaillant dans des domaines connexes, un ensemble de logiciels a ainsi été conçu ; ils permettent au robot de se déplacer sur plusieurs centaines de mètres, en construisant une représentation de l'environnement sous forme de cartes d'élévation. De *nombreuses* expériences ont permis de prouver la *faisabilité* des méthodes proposées.

Ce document est divisé en cinq chapitres mettant chacun en avant une contribution particulière de cette thèse.

Le premier chapitre présente le contexte dans lequel s'inscrivent les présents travaux. Les ambitions ultimes de la navigation autonome ainsi que les applications envisagées sont précisées et permettent de situer ce document par rapport aux recherches en cours.



FIG. 1 – Le robot Lama du LAAS. Démonstration de navigation autonome à la Cité de l’Espace. Toulouse, septembre 2000.

Un aperçu des nombreux projets de robotique abordant le problème de la navigation des robots, classés par type d’application, permet ensuite de démontrer l’aspect fondamental de la tâche de localisation.

Enfin un état de l’art des méthodes d’estimation de position fournit la matière à notre première contribution : une typologie des méthodes de localisation. Celle-ci nous a permis de définir l’ensemble des fonctions à réaliser pour obtenir un système de localisation et a guidé les développements présentés dans la suite du document.

Le second chapitre présente une méthode jouant un double rôle d’estimation de position et de contrôle des déplacements : l’odométrie.

Son utilisation sur un châssis particulier, destiné à évoluer sur des terrains accidentés, est expliquée et la dualité des aspects « contrôle » et « localisation » est analysée.

Plusieurs façons d’estimer la position sont ensuite présentées. Notamment, l’ajout de capteurs proprioceptifs montre comment améliorer la qualité, d’une part, de la position estimée et, d’autre part, des trajectoires réalisées.

En conclusion, des pistes pour la qualification des positions calculées par cette méthode sont proposées. Elles pourraient permettre de détecter les cas où l’environnement fait que cette méthode n’est pas utilisable.

Le troisième chapitre propose une méthode de localisation basée sur la vision. Celle-ci permettant d’estimer la position du robot par cumul de déplacements élémentaires, elle est analogue à l’odométrie. Elle est donc qualifiée d’« odométrie optique ».

Le principe de la méthode est tout d’abord détaillé, puis les premiers résultats obtenus sont présentés et comparés avec ceux de l’odométrie, en utilisant un système GPS comme référence.

Une étude statistique de la précision de la position calculée en fonction de différents paramètres permet d’exhiber un ensemble d’améliorations qui pourraient rendre cette méthode extrêmement performante.

Le quatrième chapitre aborde le problème de la modélisation et de la localisation simultanées, par l'intermédiaire de cartes d'élévation.

Plusieurs techniques de modélisation sophistiquées sont évaluées et comparées en simulation afin de valider les techniques plus simples couramment employées dans des contextes réels.

La structure du modèle construit est ensuite présentée. Celle-ci permet de gérer les erreurs de localisation et supporte des corrections *a posteriori* sur l'estimation de la trajectoire effectuée par le robot.

Enfin, des pistes permettant de réaliser des fonctions de localisation à partir de ce modèle sont proposées et une méthode préliminaire est comparée aux méthodes précédentes.

Le cinquième et dernier chapitre est plus prospectif : une démonstration de navigation présentant le savoir-faire actuel du LAAS en matière de navigation en environnements naturels y est détaillée.

Cette démonstration a été conçue et réalisée tout au long de la durée de cette thèse et intègre plus de dix fonctionnalités différentes — dont les trois méthodes présentées dans les chapitres précédents. Elle nous permet, dans le cadre de ce document, de mettre en avant les difficultés rencontrées lors de la conception des systèmes complexes que sont les robots.

Des problèmes généraux ayant trait au transfert et au partage des données à bord sont exhibés et une solution pour gérer une partie de ces problèmes est proposée.

I

Navigation Autonome et Localisation

Le problème de la localisation traité dans ce document a été envisagé dans le contexte de l'autonomie des robots mobiles. L'estimation de la position n'est donc pas ici un but, mais plutôt un moyen permettant d'envisager la réalisation de machines capables de naviguer seules. Ce chapitre se propose de montrer comment, et pourquoi, la localisation est une étape fondamentale pour cet objectif.

Pour cela, la navigation autonome est tout d'abord présentée à travers un aperçu des principaux projets qui ont abordé le problème, dans différents contextes. En particulier, les travaux menés au LAAS depuis près d'une dizaine d'années sont situés dans le contexte international.

L'importance que revêt la localisation est ensuite exhibée au travers d'une présentation de différentes contributions dans le contexte des environnements naturels.

Le chapitre se clôt par une synthèse qui permet d'établir une typologie des différentes méthodes de localisation. Cette classification servira, dans la suite du manuscrit, de fil conducteur pour la présentation de nos contributions, qui s'intègrent dans les classes ainsi exhibées.

1.1 Navigation autonome en environnements naturels

Il existe de nombreuses applications pour lesquelles le développement de robots autonomes, capables d'évoluer en terrains naturels, présente un intérêt. La plupart du temps, celles-ci prennent place dans des contextes où l'homme ne peut intervenir directement et où apparaissent parfois de fortes contraintes sur les délais et les débits des communications : exploration planétaire, déminage, reconnaissance militaire ou, plus généralement, intervention en milieu hostile et distant.

Le point commun entre ces applications est qu'elle requièrent l'exécution de missions de haut niveau comme « rallier tel objectif », « explorer telle région » ou « surveiller telle zone ». Ces missions se déclinent en différentes tâches qui, en particulier, forment la capacité de *naviguer*. La *navigation autonome*, dans le contexte de ce document, doit se

comprendre comme la *capacité d'enchaîner des actions de perception et de déplacement dans un environnement initialement peu ou pas connu*¹. Il s'agit, sans conteste, d'une composante fondamentale pour les applications visées par la robotique mobile.

La réalisation de telles tâches de navigation autonome requiert la mise en œuvre, l'intégration et la coopération d'un très large spectre de fonctionnalités : capacités d'exécution d'actions sensori-motrices « bas-niveau », construction de représentations de l'environnement, planification et de contrôle de l'exécution de trajectoires, élaboration de stratégies de déplacements et de perception... Pour ces raisons, la navigation autonome est une illustration exemplaire de l'intelligence des machines.

Elle reste à ce jour l'ambition d'un grand nombre de recherches en robotique mobile.

1.1.1 Spécificités des environnements naturels

Ce document se place dans le contexte des environnements dits naturels, pour lesquels aucune infrastructure humaine n'est présente *ni* envisagée (ou envisageable). Ces environnements sont pour cela souvent qualifiés de *non-structurés*, par opposition aux environnements intérieurs qui portent la marque des aménagements humains.

L'un comme l'autre de ces types d'environnements font l'objet de recherches en matière de navigation autonome et de nombreux points communs les relient. Ainsi, l'ensemble des aspects décisionnels et architecturaux sont développés sans se focaliser sur un contexte particulier. Cependant, des spécificités peuvent être distinguées assez rapidement.

La figure 1.1 présente quelques scènes typiques issues d'environnements naturels non structurés, pour lesquels des projets de robotique sont considérés. Il s'agit ici de trois cas extrêmement différents, qui exhibent une grande variabilité dans les environnements que l'on peut être amené à considérer. Cette diversité a des conséquences à la fois sur les aspects liés à la mobilité et sur ceux liés à la perception.



FIG. 1.1 – Exemples de scènes naturelles. De gauche à droite : une vue du site d'expérimentation de robotique planétaire Geroms au CNES, une vue du site d'expérimentation du LAAS, et une vue acquise en Antarctique.

Aspects liés à la mobilité

Le robot peut avoir à traverser de simples zones quasiment planes, gravir des pentes plus ou moins raides sur différents types de sols, ou franchir des zones très accidentées. Une simple structure de locomotion de type « chariot » à roues de faible diamètre — que l'on retrouve dans l'ensemble des robots mobiles d'intérieur — devient inopérante. De nombreuses structures plus ou moins complexes et originales ont donc été proposées.

¹L'acception du terme « navigation » est donc ici bien plus générale qu'elle ne l'est dans la littérature anglo-saxonne, où le même terme désigne plus précisément la capacité de se localiser.

De ce fait, les algorithmes de génération de trajectoires doivent non seulement considérer la géométrie particulière de l'environnement — que l'on ne peut se contenter de représenter à l'aide primitives géométriques simples (polygones dans un plan, par exemple) — mais aussi la géométrie particulière des structures à déplacer.

Par ailleurs, la commande des déplacements est bien plus difficile qu'à l'intérieur des bâtiments parce que, d'une part, le robot évolue dans les trois dimensions de l'espace et que, d'autre part, les interactions entre les roues et le sol sont variables, mal connues et difficilement mesurables².

Aspects liés à la perception

La première constatation évidente lorsqu'on observe les scènes de la figure 1.1 est l'absence d'éléments qui peuvent être bien représentés par des primitives géométriques du premier ordre (droites et plans), ou mêmes d'ordres supérieurs, exception faite de certaines zones du sol bien planes. Les répercussions sur l'ensemble des fonctionnalités de perception sont importantes.

En premier lieu, le problème de la segmentation des données est rendu particulièrement difficile : les séparations entre différentes régions ou objets ne sont pas très nettes et même l'oeil humain a parfois des difficultés à dessiner des frontières ou isoler des objets.

Ensuite, le choix des représentations de l'environnement devient délicat. En particulier, toute représentation qui décrit la géométrie des scènes à partir de formes géométriques *3-D analytiques*, même sophistiquées, est peu réaliste. Il semble que ce soient les modèles numériques (présentés dans le chapitre 4) qui permettent de représenter le plus fidèlement cette géométrie et que seuls les maillages déformables parviennent à décrire assez fidèlement la géométrie des objets qui s'y trouvent.

Enfin, le problème général de la *mise en correspondance* prend une forme particulière car l'appariement entre données et modèles est rendu délicat par la complexité, l'inadéquation ou même l'inexistence de modèles. Nous verrons dans le reste du manuscrit que des algorithmes de mise en correspondance données \leftrightarrow données sont souvent très efficaces : mis à part quelques cas très particuliers et rares (comme les scènes polaires), les surfaces perçues sont extrêmement texturées.

Notons par ailleurs que le cas des environnements terrestres « communs » exhibe de nombreuses couleurs et est souvent caractérisé par la présence de végétation. L'aspect changeant de celle-ci au cours des saisons et le fait que le moindre courant d'air les fasse bouger peut poser de nombreux problèmes aux différents algorithmes de perception. Nous nous restreignons cependant au cas des environnements statiques dans l'ensemble du manuscrit.

1.1.2 Quelques projets et applications

Cette section présente brièvement quelques projets de robotique en environnements naturels. Après un aperçu des travaux pionniers dans le domaine, les principaux projets actuels sont déclinés selon leur contexte d'application.

Les travaux pionniers

C'est aux États-Unis que l'on trouve les premiers projets ambitieux de navigation autonome en environnements naturels. Dès le milieu des années 80, la DARPA finançait

²Ce dernier point est évoqué dans les conclusions des chapitres 2 et 3.

le projet ALV³ [McTamney 1987, Daily 1988, Olin 1991]. Les premiers objectifs étaient le suivi autonome de routes et de chemins et, à la fin des années 80, des expérimentations de suivi de trajectoire initialement planifiées sur un modèle numérique de terrain ont été réalisées.

Dans un contexte un peu plus orienté vers les terrains non structurés (« off-road »), la DARPA a défini le programme UGV⁴ dont le Robotics Institute de CMU a été un des principaux acteurs [Thorpe 1988, Thorpe 1991a, Thorpe 1991b]. Les travaux menés dans ce contexte ont notamment débouché sur le développement et l'expérimentation de véhicules automatiques sur autoroutes, comme les plateformes Navlab à CMU [Thorpe 1997].

Enfin, mentionnons les premiers travaux dans le contexte de l'exploration planétaire menés au JPL autour du robot Robby [Miller 1989], et à CMU avec le robot Ambler [Bares 1989, Simmons 1991, Krotkov 1992].

Au cours de ces différents travaux, la plupart des aspects relatifs à la navigation autonome ont été abordés et défrichés : perception (acquisition de données, détection d'obstacles [Matthies 1995] et modélisation de l'environnement [Hebert 1989]), génération de déplacements (le plus souvent élémentaires), suivi de routes et de chemins et, bien sûr, conception d'architecture décisionnelle nécessaire à l'intégration de ces fonctionnalités [Linden 1986]. Les approches étaient encore relativement simples, le passage de la perception à l'action se faisant avec un minimum — voire pas — de planification.

Exploration planétaire

L'exploration planétaire est une ambition formidable, pour laquelle la robotique s'est rapidement imposée. En effet, les contraintes technologiques et financières rendent pour l'instant impossible toute intervention humaine sur une autre planète et, surtout, les contraintes sur les communications entre la Terre et les autres planètes, en termes de fenêtres, de débits et de délais imposent le développement de machines capables de réaliser des missions en très faible interaction avec un opérateur.

C'est donc tout naturellement dans ce contexte que l'on trouve, pour la réalisation de missions d'exploration scientifiques, les travaux de robotique en environnements non structurés les plus ambitieux en termes d'autonomie.

C'est aux États-Unis que les contributions les plus importantes ont été réalisées. Au début des années 90, la NASA a délaissé les gros systèmes développés dans les années 80 (Robby, Ambler), pour favoriser le développement de robots plus petits et plus simples — suivant en cela la philosophie de développement « smaller, faster, cheaper » [Weisbin 1992]. Les approches plutôt réactives de la navigation étaient alors nettement favorisées [Gat 1990, Gat 1994] et c'est à cette époque que les travaux autour de la famille des robots Rocky ont été commencés et inscrits dans le cadre du programme d'exploration de Mars.

Ces travaux ont notamment abouti à la conception du robot Sojourner, qui a pu accomplir près d'une centaine de mètres sur Mars lors de la mission PathFinder durant les mois de juillet et août 1997, sur le terrain cartographié par l'atterrisseur. Ses capacités d'autonomie se limitaient à l'exécution d'une petite trajectoire d'un ou deux mètres, spécifiée par les opérateurs sur Terre. En fin de mission ses capacités d'évitement d'obstacles imprévus, détectés sur la base de points 3D perçus par un système de triangulation laser/caméra, ont pu être testées [Morrison 1996, Stone 1996].

³Autonomous Land Vehicle.

⁴Unmanned Ground Vehicle.

Les futures missions du programme, dont les échéances ont été révisées suite aux échecs des missions Mars Polar Lander et Mars Climate Observer fin 1999, impliquent cependant des robots bien plus gros dont les capacités d'autonomie devraient leur permettre de générer et d'exécuter des trajectoires de plusieurs dizaines de mètres (comme les deux robots Athena, qui seront lancés en 2003) : le temps des simples approches réactives est révolu.

Actuellement, de nombreux travaux portant essentiellement sur les capacités de navigation autonome sont menés au JPL, tandis que le centre de recherche NASA Ames étudie les aspects plus amonts relatifs à l'autonomie décisionnelle [Washington 1999]. La NASA supporte aussi des projets dans des Universités : le robot Nomad a été testé dans le désert d'Atacama et à plusieurs reprises dans l'Antarctique par les chercheurs et ingénieurs de CMU [Whittaker 1997, Moorehead 1999].

En Europe, le projet d'exploration planétaire le plus ambitieux est le projet I-ARES⁵, lancé à l'initiative du CNES en 1993 à la suite du programme d'étude VAP⁶ engagé en 1989 [Giralt 1992]. C'est dans ce contexte que s'est constitué le groupement de partenaires académiques RISP⁷ auquel le LAAS a beaucoup participé.

Le projet I-ARES visait à la réalisation d'un démonstrateur terrestre, capable d'effectuer de manière autonome des missions d'exploration spécifiées par un opérateur [Boissier 1995]. De nombreux partenaires industriels et académiques (dont le LAAS) y ont participé, et les travaux ont couvert un large spectre de problèmes relatifs à la navigation autonome : perception et modélisation du terrain, localisation, génération et exécution de trajectoires. D'autres aspects relatifs à la planification de missions et au développement d'une structure de locomotion ont aussi été considérés.

Le projet, qui n'était pas défini dans un contexte programmatique, a quasiment été interrompu en 1996. Heureusement, une activité de développement pour la robotique d'exploration planétaire a pu être maintenue [Maurette 1998, Rastel 1998] et les ingénieurs du CNES ont désormais un savoir-faire qui leur permet de s'impliquer dans le programme de la NASA.

Ce n'est que depuis peu que l'Agence Spatiale Européenne considère la robotique d'exploration planétaire. Les récents projets financés sont encore restés peu ambitieux car ils concernent l'étude de micro-rovers de moins de 10 kilos, qui doivent effectuer des trajectoires de quelques mètres en visibilité de l'atterrisseur, de manière analogue au robot Sojourner [Fontaine 2000]. Néanmoins, l'ESA affiche de plus en plus la robotique comme un moyen indispensable à l'exploration planétaire. Des études très amont sur l'exploration et l'exploitation de ressources planétaires sont réalisées en interne [Novara 1998] et des investissements plus conséquents dans des projets plus ambitieux sont annoncés pour les années à venir [Putz 2000].

Enfin, d'autres travaux relatifs à la robotique d'exploration planétaire sont menés au Japon [Kubota 1998]. Signalons aussi l'existence d'une initiative privée américaine (Lunar Rover Initiative), dans le cadre de laquelle CMU a mené des travaux sur la navigation au milieu des années 90 [Simmons 1995], mais qui semble avoir été abandonnée.

L'avenir de la robotique d'exploration planétaire est sans aucun doute assuré : d'une part, les prochaines missions planétaires habitées n'auront pas lieu avant de nombreuses années et, d'autre part, tant à la NASA qu'à l'ESA ces missions habitées sont d'ores et déjà envisagées avec un support robotique lourd, déployé préalablement et utilisé pendant l'exploration par des hommes.

⁵Illustrateur Autonome de Robotique d'Exploration Spatiale.

⁶Véhicule Autonome Planétaire.

⁷Robotique d'Intervention sur Site Planétaire.

Applications terrestres

Outre l'exploration scientifique dans les zones polaires — problème très analogue à l'exploration planétaire — il existe plusieurs contextes d'application terrestres de la navigation autonome en environnements naturels qui sont considérés par les roboticiens : applications militaires, déminage, et plus généralement intervention en milieu hostile.

D'autres applications, telles que celles de la robotique agricole, visent à automatiser des tâches longues et répétitives. Elles ne sont pas présentées ici car la navigation autonome — telle que nous l'entendons — n'est pas une priorité pour elles. Il est néanmoins évident que les travaux sur l'autonomie pourraient leur être profitables.

Applications militaires L'intérêt de disposer de véhicules terrestres sans pilotes est évident dans un contexte d'opérations militaires, et nous avons vu que dès les années 80, la DARPA finançait des recherches sur ces aspects.

Les premières tâches considérées concernaient essentiellement la reconnaissance et le suivi de routes ou de chemins, en se reposant néanmoins sur une interaction assez forte avec un opérateur distant. Plus récemment, la DARPA a lancé le projet extrêmement ambitieux TMR⁸ [Krotkov 1999] pour lequel les besoins exprimés en terme d'autonomie sont très grands.

En France, la DGA a financé le projet DARDS au début des années 90 mais ne commence à considérer des véhicules dotés d'autonomie réelle que depuis peu (dans le Plan d'études Amont Robotique par exemple). DARDS consistait initialement à développer un système de télé-conduite où les capacités d'autonomie du véhicule étaient quasiment inexistantes, mais a maintenant évolué vers un système autonome basant sa conduite sur des informations GPS.

Déminage Le problème posé par les innombrables zones minées non cartographiées est bien connu, particulièrement pour les mines anti-personnels. Il s'agit là d'un domaine d'application bien plus motivant que le précédent, pour lequel des techniques robotiques d'exploration autonome pourraient apporter beaucoup : différentes rencontres dédiées à ce problème ont déjà eu lieu (notamment au LAAS en septembre 1998 [Demining 1998]).

Quelques projets sont menés dans ce contexte [DeBolt 1997], mais les défis sont immenses. D'une part, il reste beaucoup de progrès à effectuer dans le domaine de la détection des mines (la « charge utile » des robots). D'autre part, les types d'environnement à traiter sont le plus souvent extrêmement complexes : zones urbaines dévastées ou naturelles très accidentées, présence de végétation...

Beaucoup de travail reste donc à effectuer dans les différents domaines considérés par les roboticiens pour fournir des systèmes capables de cartographier les mines (exhaustivement et sans oubli) dans de tels environnements.

Intervention en milieu hostile Il s'agit d'un des premiers contextes d'application envisagés pour la robotique en environnements non structurés (comme, par exemple, le projet AMR⁹). L'exemple de la catastrophe de Tchernobyl nous a clairement montré l'intérêt de disposer de machines capables de réaliser différentes tâches de reconnaissance, de manipulation, d'excavation ou d'enfouissement, en limitant l'intervention d'opérateurs sur place.

Pour ces projets d'intervention après des catastrophes industrielles, de lutte contre les pollutions dangereuses ou les incendies, des approches téléopérées sont étudiées.

⁸Tactical Mobile Robots.

⁹Advanced Mobile Robot.

La difficulté d'opérer des robots à distance et les moyens de communications que cela nécessite font que des approches plus autonomes pourront apporter une efficacité supplémentaire.

Dans un contexte analogue, on peut remarquer l'apparition récente de projets relatifs à l'intervention pour le sauvetage de victimes (« rescue robotics ») : deux sessions entières étaient consacrées à ces applications au cours de la conférence [IROS 2000].

1.1.3 Le projet EDEN au LAAS

Aperçu historique

L'origine des travaux du LAAS dans le domaine de la robotique en environnements naturels remonte à la toute fin des années 80, au sein du projet AMR qui consistait à concevoir et développer un robot d'intervention sur site hostile [Laurette 1991]. Les études se sont poursuivies dans le cadre du projet VAP, mais c'est à la fin de l'année 1992, lors de l'arrivée au laboratoire du robot Adam, instrumenté dans le contexte d'AMR (figure 1.2), que les travaux ont pris de l'ampleur. À cette occasion, le projet EDEN¹⁰ a été défini : il s'agit d'un projet interne au laboratoire, dont l'objectif est de concevoir, développer et intégrer les fonctionnalités nécessaires à la réalisation de tâches de navigation autonome en environnements naturels.



FIG. 1.2 – Le robot Adam. Utilisé au LAAS de 1992 à 1996, sa couche de locomotion a été développée par Matra Marconi Space. Le seul moyen d'acquérir des données sur son environnement était un télémètre laser associé à un système de déflexion suivant deux axes.

Un large spectre de problèmes a été abordé à l'aide du robot Adam : perception et modélisation de l'environnement [Devy 1994], planification de trajectoires sur des terrains accidentés [Simeon 1993], stratégies de navigation [Lacroix 1995b], localisation sur amers [Fillatreau 1993, Betge-Brezetz 1996] ou aspects décisionnels liés à la planification et l'exécution de missions [Chatila 1993]. Différentes démonstrations ont été intégrées à bord du robot (génération de chemins pour rallier un but, planification de trajectoires et localisation sur amers). Elles étaient excessivement lentes, le robot s'arrêtant près de 3 minutes à chaque acquisition d'une image de profondeur, et n'ont pas été poussées très loin.

Au début de l'année 1996, à la suite de l'arrêt du projet I-ARES, la société Al-

¹⁰Expérimentations de Déplacement en Environnements Naturels.

catel Space Industries a prêté au LAAS¹¹ le robot *Lama*, qui a été instrumenté par le laboratoire (figure 1.3). L'arrivée de ce robot a donné un second souffle aux travaux, même si l'absence de grand projet collaboratif a provoqué une baisse du volume des activités. Le groupe a néanmoins continué un travail de fond sur la plupart des problèmes relatifs à la navigation autonome, et dès fin de l'année 1997, des démonstrations de navigation réactive en mouvement continu, basées sur la stéréo-vision, ont été réalisées [Haddad 1998b]. Différents progrès ont été effectués, essentiellement dans le domaine de la perception et la modélisation de l'environnement (algorithmes de vision de base [Murrieta-Cid 1997, Vandapel 2000, Jung 2001], de sélection et de suivi d'éléments particuliers [Murrieta-Cid 1998], de localisation [Ayala-Ramirez 1998, Devy 1998], de modélisation [Devy 1999]), mais aussi dans celui de la génération des déplacements [Hait 1997, Bonnafous 2001] et de leur contrôle [Andrade-Barosso 1998, Chatila 1999]. L'ensemble des travaux présentés dans ce manuscrit s'inscrivent bien entendu dans ce contexte.



FIG. 1.3 – Le robot *Lama*, sur le site d'expérimentation du LAAS pendant l'été 2000. Il perçoit son environnement grâce à deux bancs stéréoscopiques montés sur des tourelles orientables.

Récemment, les aspects relatifs à l'opération coordonnée de robots terrestres et aériens ont commencé à être considérés dans le projet [Lacroix 2001b]. Dans ce but, le groupe a acquis un ballon dirigeable en 2001, en cours d'instrumentation [Lacroix 2000a] : des travaux sur la commande de ce type d'engins [Hygounenc 2001] et sur la modélisation coordonnée aérienne/terrestre [Lacroix 2001a] ont débuté (nous évoquerons ces derniers dans le chapitre 3).

Approche générale

L'approche générale de résolution du problème de la navigation autonome a été définie très tôt dans le projet : elle consiste à développer un robot qui *adapte* son mode de navigation en fonction du type de terrain à traverser [Lacroix 1994, Chatila 1995]. Différents modes ont donc été définis et démontrés à bord des robots *Adam* et *Lama* :

- Des modes de déplacement réactifs, appliqués aux terrains simples (essentiellement plans, avec quelques obstacles épars), dans lesquels des mouvements élémentaires sont générés sur la base d'informations locales (détection d'obstacles) ;
- Des modes de déplacements où les trajectoires à exécuter sont planifiées sur la base de représentations globales de l'environnement, applicables lorsque le terrain est plus encombré, accidenté, ou exhibe des impasses ;

¹¹On trouvera une description précise de *Lama* et de ses instruments en annexe A.

- Et à bord du robot *Lama*, un mode de locomotion *péristaltique*, appliqué lorsque la faible cohérence du sol traversé empêche ou rend dangereux les déplacements en mode de roulement (voir annexe A).

Les travaux les plus récents dans le cadre du projet se situent toujours dans une telle approche [Lacroix 2000b], que nous jugeons nécessaire pour la réalisation de robots capables d'effectuer de manière autonome des tâches de navigation à *long terme* (sur plusieurs kilomètres et durant plusieurs heures). Notons que la définition des différents modes de navigation reste bien entendu ouverte, et à adapter aux types d'environnements et de structures de locomotion considérés.

La volonté de doter le robot d'une telle « versatilité » a plusieurs implications : elle nécessite le développement de différentes fonctions de modélisation de l'environnement — et donc le maintien de la cohérence entre les représentations construites — et le développement et l'intégration de différents moyens de générer les déplacements — et donc le contrôle *actif* des différentes activités du robot.

1.1.4 Conclusions

Ce rapide aperçu montre que, d'une part, le champ potentiel d'applications considérés par les roboticiens en environnements non structurés (et pas uniquement naturels) est très vaste et, d'autre part, qu'un nombre de plus en plus grand de contributions y sont consacrées. Les plus ambitieuses considèrent même désormais l'opération autonome d'un ensemble de robots [Brumitt 1998] et la coopération entre robots terrestres et aériens.

Le travail à effectuer pour passer d'une démonstration de laboratoire à la réalisation effective d'engins opérationnels reste cependant énorme : nous avons vu les faibles capacités d'autonomie décisionnelle du robot Sojourner et les difficultés d'opération rencontrées lors de l'intervention des robots sous-marins Abyssub et Triton sur l'épave du pétrolier Erika¹².

De nombreux progrès ont néanmoins été réalisés pour la résolution de l'ensemble des problèmes posés par la navigation autonome en environnements naturels : des moyens fiables et rapides de percevoir l'environnement ont été développés, des techniques de modélisation de l'environnement et de génération de déplacements ont fait leurs preuves, des avancées dans les travaux de planification permettent d'aborder des problèmes d'une grande complexité en considérant des contraintes sur le temps et les ressources, des progrès en automatique permettent de mieux en mieux de contrôler l'exécution des déplacements, différentes architectures capables d'intégrer l'ensemble des fonctionnalités ont été définies... Les principales difficultés qui subsistent sont essentiellement liées au manque de robustesse et de versatilité de ces différentes fonctions et à la complexité posée par leur intégration. Nous verrons dans le paragraphe suivant que le problème de la localisation est un élément clef de la robustesse de nombreuses fonctions et de leur intégration.

1.2 De l'importance de la localisation

Pour les différentes applications considérées par la robotique en environnements naturels — et pour la robotique mobile d'une manière plus générale — la capacité pour le robot de se *localiser*, c'est-à-dire de connaître à tout instant sa position et la précision de celle-ci, est *essentielle* pour plusieurs raisons :

¹²Nous insistons ici sur la difficulté opérationnelle liée au contexte, ces derniers robots étant entièrement téléopérés : le robot Triton a été mis à l'eau pour libérer le robot Abyssub...

- La plupart des missions que doivent exécuter les robots sont souvent définies en termes de lieux géographiques (« rallier tel objectif », « explorer telle région », ...). C'est donc la connaissance de la position courante du robot, mais aussi celle des positions par lesquelles il est passé qui permettent d'assurer et de contrôler l'exécution de ces missions.
- Une approche réaliste de la navigation autonome à long terme requiert la mise en œuvre de différents moyens de construction de représentations de l'environnement, à partir des données acquises par le robot lors de ses déplacements. Il y a bien sûr tout intérêt à maintenir la cohérence entre ces différentes représentations, tant pour améliorer la qualité de chacune d'entre-elles que pour augmenter la « compréhension » de l'environnement par la machine. Cette *cohérence spatiale* ne peut être garantie que si les positions du robot sont connues lors de l'acquisition des données.
- Les planificateurs de trajectoires et les fonctions décisionnelles qui contrôlent le déroulement de la mission se basent sur des représentations *globales* de l'environnement : la *cohérence spatiale* est donc encore une fois requise. De plus, c'est la mesure précise des déplacements du robot qui permet d'assurer l'exécution correcte des trajectoires *géométriques* déterminées par les planificateurs. Seuls des algorithmes de localisation permettent cette mesure.

Le problème de la localisation est donc présent à différents niveaux de la navigation autonome : sa résolution est nécessaire pour assurer aussi bien la bonne exécution d'un déplacement élémentaire que la réalisation correcte d'une mission définie à haut niveau d'abstraction. En particulier, il est intimement lié à la modélisation de l'environnement, comme cela a été rapidement compris par les roboticiens [Crowley 1985, Chatila 1985], au point que l'acronyme « SLAM », introduit au début des années 90 pour « Simultaneous Localization And Map building » [Leonard 1991], est largement répandu et compris dans la communauté¹³. Pour ces différentes raisons, nous considérons que la capacité de se localiser est une composante essentielle au développement de la navigation autonome, et est notamment la clef de la *robustesse* des approches.

Nous verrons dans la section suivante que la résolution du problème de la localisation en robotique implique l'intégration de nombreuses fonctionnalités de natures très diverses, qui vont de l'estimation de mouvement à la reconnaissance d'objets naturels et de lieux. Mais avant de présenter les différents moyens développés par les roboticiens pour aborder ce problème, il est intéressant de considérer la place que la nature et les hommes ont faite à la localisation.

Parmi les nombreux travaux scientifiques qui cherchent à expliquer les différentes formes d'intelligence présentes chez les être vivants, beaucoup ont trait à la *compréhension spatiale* qui semble être une composante nécessaire au développement de capacités intelligentes. Il paraît clair que les organismes les plus évolués utilisent un large spectre de fonctions qui vont, pour cette composante, de l'anticipation de la réalisation de mouvements et de déplacements à la reconnaissance d'objets ou de lieux. Mais il est également clair que la localisation est au cœur de la compréhension spatiale : de nombreuses études sont consacrées à la description et à l'explication des moyens qui permettent aux êtres vivants et mobiles de se localiser, qu'il s'agisse d'abeilles, de fourmis, de mammifères, d'oiseaux migrateurs... Le caractère essentiel de la localisation est aussi évidemment indéniable chez les hommes : notons simplement que l'impression de « se sentir perdu », dans un bâtiment, une ville ou dans la nature, est souvent anxiogène.

¹³Certains auteurs préfèrent l'acronyme CML, pour « Concurrent Mapping and Localization ».

La variété des solutions développées est très grande, et plusieurs solutions sont parfois présentes au sein d'une seule espèce : utilisation de véritables centrales inertielles, d'informations visuelles sur le mouvement (flot optique), mémorisation des chemins et des lieux par exemple. De nombreuses espèces ont aussi développé des techniques *ad hoc*, comme les marqueurs chimiques utilisés par certains insectes et mammifères.

Ces différentes solutions, que les éthologues et cognitiens cherchent à expliciter plus précisément, inspirent parfois des développements en robotique (voir par exemple un numéro spécial de la revue « Adaptive Behavior » [AB 1998] et le recueil de contributions dans [Aloimonos 1997]).

Il est aussi remarquable de voir la grande place qu'a occupée le problème de la localisation au cours des évolutions technologiques, et même les développements qu'il a induit. Le repérage à l'aide des étoiles a suscité de nombreuses avancées en géométrie et c'est en grande partie pour des besoins de localisation qu'ont été développés des instruments de mesure (sextants, loch, chronomètres, ...). Ce n'est d'ailleurs pas un hasard si ces derniers instruments évoquent les grands navigateurs : la réalisation de missions d'exploration d'environnements inconnus, mais aussi de tâches de navigation de ligne ne peuvent être conçues sans la capacité de se localiser. Plus proche de nous, la maîtrise de l'émission et la réception des ondes radio a vite conduit au développement du système de localisation Loran¹⁴ dès les années 40 et l'une des applications les plus répandues des technologies spatiales est la localisation par satellite (balises Argos, systèmes GPS, Glonass et bientôt GlobalStar), ces dernières applications connaissant de nos jours un développement très important.

1.3 Localisation en environnements naturels : état de l'art

Étant donné le rôle central joué par la localisation dans la navigation autonome, il n'est pas surprenant de trouver de très nombreuses contributions relatives à ce problème dans la littérature. Elles se répartissent dans un très large spectre, les capteurs, outils et formalismes utilisés étant très variés. [Borenstein 1996] est un ouvrage qui présente un bon aperçu des différentes approches, surtout en environnements intérieurs, avec des aspects pratiques liés au capteurs et aux outils utilisés très intéressants.

Cette section présente quelques contributions à ce problème dans le contexte d'environnements non-structurés, en les classant en fonction des données qui sont utilisées pour la résolution du problème. Seules les techniques de localisation *autonomes* sont présentées, c'est à dire celles qui n'utilisent que des informations fournies par les capteurs embarqués et non celles qui font appel à des informations externes, comme par exemple des balises radio disposées dans l'environnement.

1.3.1 Localisation odométrique

A priori utilisable dès que la structure de locomotion est à roues, la localisation odométrique est extrêmement répandue dans le contexte des environnements intérieurs, où les robots évoluent sur un plan. Elle est par contre moins utilisée en environnements naturels, d'une part parce que le robot évolue dans les trois dimensions de l'espace, ce qui nécessite l'utilisation d'au moins un inclinomètre pour mesurer l'angle de site¹⁵ du robot, et d'autre part parce que les principaux problèmes liés à l'odométrie (impossibilité de mesurer les glissements et dérapages) sont bien plus fréquents et importants qu'en environnements intérieurs.

¹⁴LONg RANge Navigation.

¹⁵On dit aussi « angle de tangage », voir chapitre 2.

Dans le contexte de la robotique d'exploration planétaire, [Fuke 1996] présente un système qui intègre les données fournies par les codeurs odométriques d'une plateforme à 4 roues avec les données fournies par trois gyromètres et trois accéléromètres. L'article est essentiellement consacré aux aspects liés à la fusion d'informations et des résultats quantitatifs sont présentés sur des portions de lignes droites de quelques mètres effectuées sur différents types de sols.

Toujours dans le contexte de l'exploration planétaire, on trouve dans [Volpe 1999] un système qui combine les informations odométriques avec un inclinomètre 2-axes et un capteur qui mesure l'angle sous lequel est perçu le soleil (analogue aux capteurs stellaires montés sur les satellites). Les auteurs présentent des résultats quantitatifs mesurés lors d'essais avec le robot *Rocky 7*, sur des trajectoires de plusieurs centaines de mètres — le robot n'effectuant toutefois que des rotations sur place ou des portions de droites. L'erreur relative sur la position n'excède par 6% et le cap du robot est toujours très bien estimé grâce au capteur solaire. Curieusement, aucune mention n'est faite de l'estimation de l'altitude du robot.

Dans [Bonnifait 1997], où sont considérées les applications de « robotique de chantier », les équations de l'odométrie 3D avec pour seul capteur supplémentaire un inclinomètre 2-axes sont présentées; la fusion de cet estimateur de position avec un système permettant de mesurer les angles sous lesquels sont vues des cibles de positions connues est introduite. Des résultats mesurés le long de trajectoires de plusieurs dizaines de mètres sont analysés : l'estimation de l'altitude est faite avec une précision de l'ordre du centimètre.

Enfin, on trouvera dans [Green 1994, Roumeliotis 1999] d'autres contributions qui intègrent les informations odométriques avec des données inertielles et dans [Miyake 1997, Sukkarieh 1999a] des travaux qui intègrent les données odométriques avec celles fournies par des récepteurs GPS.

Remarquons que ces différentes contributions à l'odométrie en environnements naturels considèrent toujours l'ajout de capteurs supplémentaires pour avoir une estimée réaliste des mouvements du robot : le chapitre 2 montrera pourquoi cela est indispensable.

1.3.2 Localisation inertielle

Les capteurs inertiels sont bien sûr incontournables pour la localisation : différentes technologies de perception de moments et de forces permettent d'avoir une mesure des accélérations transversales et des vitesses de rotations, mesures dont l'intégration dans le temps permet de déduire les 6 paramètres de position d'un mobile dans l'espace.

Ces capteurs sont particulièrement utilisés en navigation aérienne. Il s'agit d'une technologie que évolue très rapidement et si les plus chers d'entre eux ont d'excellentes performances en précision, on trouve désormais sur le marché des accéléromètres et gyromètres compacts et relativement précis pour quelques milliers de francs [Barshan 1995, Everett 1995, Pleska 2000]. La performance de ces systèmes est cependant bien moindre que celle d'une centrale gyroscopique de haut de gamme¹⁶.

Ces types de capteurs sont de plus en plus présents à bord de robots. Mais s'ils peuvent permettre d'asservir précisément l'exécution d'une courte trajectoire, leur utilisation exclusive comme estimateur de position n'est pas possible en grande partie à cause de la faible vitesse des robots considérés. D'une part, les estimations de position

¹⁶Parmi ces nouveaux capteurs, les gyromètres à fibre optique, qui ne sont pas à proprement parler des capteurs inertiels, sont particulièrement performants et de plus en plus utilisés — voir l'annexe A.

étant effectuées par intégration des mesures dans le temps, celles-ci dérivent d'autant plus (relativement à la distance parcourue) que le mobile est lent. D'autre part, les vibrations du robot induisent de forts bruits sur les mesures qui deviennent prépondérants devant le signal (particulièrement à faibles vitesses et en terrains accidentés). Ainsi, après essais d'une centrale inertielle construite spécialement, un bilan système global (masse, consommation et précision) a conduit les ingénieurs du CNES à préférer développer pour leur prototype expérimental un estimateur de mouvement basé sur la stéréovision à une centrale inertielle.

Les applications où les robots vont à plusieurs mètres par seconde permettent l'utilisation de capteurs inertiels. Néanmoins, les données de position de ces derniers sont souvent fusionnées à celles produites par d'autres moyens (par exemple grâce à des techniques d'hybridation de données inertielles et GPS) : l'Australian Center for Field Robotics de l'université de Sydney est laboratoire très actif dans ce domaine [Sukkarieh 1997, Nebot 1999].

1.3.3 Localisation à partir d'images de profondeur

Les images de profondeur, produites par télémétrie laser ou radar et par stéréovision, sont sans aucun doute les données extéroceptives les plus utilisées pour déterminer la position d'un robot lors de ses déplacements. On distingue deux types de contributions : les méthodes qui utilisent des représentations de l'environnement construites à partir des données et les méthodes qui utilisent directement les données, sans les structurer ni les modéliser.

Parmi les méthodes qui utilisent un modèle de l'environnement construit au cours de la navigation, certaines sont basées sur la détection *d'amers* dans l'environnement. Le modèle, composé des différents amers perçus, est donc dans ce cas exclusivement dédié à la fonctionnalité de localisation. La difficulté réside dans la conception des algorithmes d'extraction et de modélisation des amers, rendus difficiles par le caractère non structuré des environnements.

Dans [Hebert 1992], les amers sont détectés à partir de lignes de gradient extraites des images de profondeur et de réflectance acquises par télémétrie laser, et modélisés par des maillages déformables (nous avons développé des algorithmes similaires dans [Mallet 1997]).

Au LAAS, des méthodes utilisant le sommet d'objets, isolés d'un sol plat par une procédure de segmentation très simple [Betge-Brezetz 1995, Parra 1997], ont été proposées. Couplées à un filtre de Kalman étendu, elles sont une contribution au problème « SLAM » [Betge-Brezetz 1996].

D'autres travaux effectués dans le laboratoire se basent sur l'extraction de contours correspondant à la frontière entre différentes zones du terrain [Devy 1998], et nous avons proposé une méthode de ce type dans [Mallet 1998]. Notons que des techniques basées sur des amers ont aussi pu être développées dans le contexte de robots sous-marins, où la télémétrie ultra-sonore permet de détecter des amers sur le fond [Majumder 2000].

Beaucoup d'autres méthodes estiment la position du robot, ou corrigent une position estimée par ailleurs, sur la base d'un modèle numérique de terrain construit lors de la navigation. Le modèle n'est donc cette fois pas exclusivement dédié à la localisation, puisqu'il peut aussi permettre, par exemple, la détermination de trajectoires.

Certaines techniques se basent sur la mise en correspondance d'un modèle agrégé à partir des perceptions précédentes avec un modèle local construit à partir de la der-

nière perception : on parle de mise en correspondance « iconique »¹⁷. Les techniques de recherche d'appariement sont analogues à des techniques développées pour des images vidéo [Asada 1988, Kweon 1992, Olson 1997, Olson 1998] (on trouve des contributions analogues dans le contexte de la robotique sous-marine [Lucido 1998]).

D'autres techniques partent des modèles numériques pour extraire des amers, ceux-ci correspondant à des objets clairement identifiés dans l'environnement [Kweon 1991, Fillatreau 1993], où des zones du modèle présentant des caractéristiques (signatures) particulières [Olson 2000b, Huber 1999].

Enfin, quelques travaux préliminaires qui traitent de la mise en correspondance d'un modèle numérique de terrain avec des informations d'altimétrie initialement disponibles ont été présentées dans [Gennery 1989] et [Matthies 1997]. Ce dernier problème nous apparaît cependant encore largement ouvert.

Pour l'ensemble de ces algorithmes qui se basent sur une représentation de l'environnement, toute la difficulté réside bien sûr dans la construction et l'analyse de ces représentations. Pour le cas particulier des méthodes qui extraient directement des amers à partir des données de profondeur, ce sont notamment les algorithmes de segmentation qui sont délicats (nous reviendrons sur ces considérations dans le chapitre 4).

Malgré ces difficultés, les méthodes qui utilisent directement les données de profondeur, sans étape de segmentation, structuration ou modélisation, sont plus rares. Dans [Zhang 1991], on trouve une méthode de détermination du déplacement entre deux images des points $3-D$, acquises par stéréo-vision, qui se base sur la minimisation d'une énergie entre les deux nuages de points, sans que des correspondances entre ces points aient besoin d'être établies. Des techniques d'estimation du déplacement une fois que les points sont mis en correspondance existent bien [Arun 1987, Haralick 1989], mais la difficulté est justement d'établir ces correspondances. L'extraction et l'appariement de points particuliers dans les images de points 3D est rendue très difficile par le bruit sur les données : c'est donc le plus souvent à partir des informations vidéo qu'est effectuée cette mise en correspondance [Carre 1998, Olson 2000a]. Le chapitre 3 est dédié à la présentation d'une méthode de ce type que nous avons développée.

1.3.4 Localisation à partir de la vision

Les images acquises par vision contenant des informations extrêmement riches, beaucoup de contributions au problème de la localisation sont basées sur ces données, qu'elles proviennent d'une caméra munie d'un objectif « classique » ou panoramique (nous excluons dans ce paragraphe la stéréo-vision, qui produit des images de profondeur).

Il y a d'abord les différentes techniques développées dans la communauté des spécialistes en vision qui abordent la résolution du problème « motion and structure from motion ». Elles consistent à reconstruire à partir d'une séquence d'images la géométrie de l'environnement et la trajectoire suivie par la caméra. Les progrès en vitesse de calcul des processeurs, qui permettent par exemple de calculer rapidement le flot optique, font que ces méthodes se développent. À notre connaissance, aucune n'a cependant encore été appliquée à bord de robots.

Les techniques utilisées sont variées et on trouvera une bonne revue des contributions à ce problème dans [Huang 1994]. Au laboratoire, une méthode d'estimation de déplacement d'une caméra, basée sur une analyse fréquentielle des images, a été proposée [Ayala-Ramirez 1998].

¹⁷ « iconic matching » en anglais, à opposer aux techniques dites « feature matching ».

Plus intéressantes sont les contributions qui se basent sur la détection et le suivi d'amers par vision : comme elles mémorisent les amers de l'environnement, l'erreur sur leurs estimées de position croît bien moins vite qu'avec les méthodes précédentes.

Parmi ces techniques, beaucoup considèrent des amers définis par la ligne d'horizon (sommets, points de rupture sur cette ligne, ou même forme globale du profil) et les mettent en correspondance avec ceux détectés sur les profils de l'horizon *prédits* sur la base d'un modèle numérique initial de l'environnement [Suh 1993, Cozman 1997]. La précision de l'estimation finale dépend bien entendu de la finesse du modèle numérique de terrain initial [Cozman 1998].

Ces méthodes sont confrontées à deux problèmes : la complexité algorithmique liée la détermination des profils prédits et l'extraction des amers sur l'horizon des images perçues. De plus, il n'est pas garanti que de tels amers existent : par exemple, aucun n'est visible dans les images acquises par Viking et les deux sommets des « twin peaks » perçus par l'atterrisseur de la mission PathFinder ne sont pas suffisants. De plus, la probabilité d'occultation de ceux-ci par des éléments plus proches est grande (ainsi les sommets de « twin peaks » ne sont pas visibles dans les images acquises par le robot Sojourner).

D'autres techniques sont basées sur la détection, la mémorisation et la mise en correspondance d'amers situés dans l'environnement proche du robot. La mesure des angles sous lesquels ils sont vus à partir de différentes positions (éventuellement estimées par ailleurs) est une instance en 2 dimensions du problème « SLAM » [Maeyama 1997, Deans 2000].

Enfin, d'autres méthodes utilisent des « amers implicites », grâce à des techniques d'indexation d'images sur des vues panoramiques. Le terme « implicite » permet d'insister sur le fait que les index définis sur les images ne correspondent pas toujours à des éléments particuliers de l'environnement ; il peut même s'agir d'attributs globaux calculés sur l'image. Ces contributions commencent à se développer en environnements intérieurs [Matsumoto 1999, Matsui 2000] et semi-structurés (urbains par exemple), mais absolument rien ne s'oppose à leur utilisation en environnements naturels. Une thèse a débutée au LAAS sur ce problème [Gonzales 2000], et nous sommes convaincus que ces techniques ont beaucoup d'avenir.

1.3.5 Localisation topologique

Il existe de nombreuses contributions qui se réclament de la « localisation topologique », que nous mentionnons dans un paragraphe distinct des précédents car elles se basent le plus souvent sur des modèles de l'environnement abstraits, qui résultent de l'intégration de données hétérogènes.

Si l'on se tient à la définition du dictionnaire, la topologie est la partie de la géométrie qui traite « *des propriétés qualitatives et des positions relatives des êtres géométriques, indépendamment de leur forme et de leur grandeur* ». En ce sens, elles sont bien distinctes de l'ensemble des contributions précédentes car elles ne fournissent pas un résultat sous la forme d'une position métrique : le principe est de déterminer la position du robot relativement à un modèle topologique. Le terme de « localisation qualitative » est souvent utilisé pour désigner ces approches.

Parmi l'ensemble de ces contributions, certaines utilisent néanmoins des modèles topologiques construits sur la base de représentations métriques : nous pourrions les qualifier de « topologiques-métriques » [Thrun 1998]. Les modèles utilisés sont essentiellement des graphes de lieux et de chemins ; représentations qui se sont rapide-

ment avérées nécessaires pour la planification de tâches et de stratégies de déplacement (ainsi [Chatila 1995, Lacroix 1995b, Bulata 1996] au LAAS). Quelques contributions ont traité du problème de la localisation à partir de telles représentations, en utilisant des techniques d'appariement de graphes, pour lesquelles des informations métriques valant les arcs sont utilisées [Dudek 1993, Simhon 1998]. Parmi ces approches, il faut aussi mentionner celles qui utilisent un « graphe de Voronoi » [Nagatani 1998, Fox 1999, Zwynsvoorde 2000], construit sur la base d'informations télé-métriques et qui permettent de situer le robot par rapport à des noeuds particuliers du graphe.

Les approches topologiques « pures » excluent toute utilisation d'informations métriques. Initialement proposées par [Kuipers 1988] et [Levitt 1990], elles considèrent donc uniquement la construction de modèles de l'environnement à partir d'un graphe de lieux distincts. Il est possible de localiser le robot dans un tel graphe sur la seule base d'informations sémantiques (reconnaissance de noeuds du graphe ou de sous-graphes), mais il est évident que l'ajout d'informations métriques permet de rendre plus robuste ces approches.

Les approches basées sur des techniques d'indexation d'images, déjà mentionnées dans le paragraphe précédent et, plus généralement, celles qui se basent sur des modèles implicites de l'environnement [Herrmann 1999], peuvent aussi être considérées comme « qualitatives » : ce terme s'appliquant d'autant mieux qu'elles modélisent et reconnaissent des objets et des lieux de manière non analytique.

Ces approches séduisantes du problème de la localisation ont donné lieu à des développements dans différents laboratoires en environnements structurés. Elles s'inscrivent parfois dans une approche plus générale de la navigation autonome, où les déplacements sont générés grâce à des associations données \leftrightarrow actions mémorisées lors d'une phase d'exploration préliminaire.

À l'heure actuelle, elles sont encore peu considérées dans les différents projets applicatifs en environnements naturels, où les approches « analytiques » convainquent plus facilement les ingénieurs.

1.4 Typologie des méthodes de localisation

Nous venons de voir que les contributions au problème de la localisation sont extrêmement variées, tant du point de vue des données utilisées que des approches retenues (ce qui d'ailleurs confirme le caractère essentiel du problème).

Nous proposons ici une typologie de ces contributions, en les regroupant en trois grandes catégories : estimation de mouvement, recalage et localisation absolue. Des critères de classements différents sont ensuite présentés. L'intérêt de cette typologie est de mieux cerner le problème de la localisation dans son ensemble et d'exhiber ses différentes implications dans la conception d'un système de navigation autonome.

Il serait intéressant, à titre anecdotique, de vérifier si une classification similaire peut s'appliquer aux différentes solutions au problème de la localisation présentes dans le règne animal. Nous avons vu que la nature a développé différentes techniques d'estimation du mouvement (on parle alors de « vection ») sur la base de capteurs inertiels ou de vision. Par ailleurs, il ne fait aucun doute que de nombreuses techniques de recalage, basées sur une mémorisation de l'environnement, sont présentes chez beaucoup d'êtres vivants : les différents phénomènes impliqués dans cette capacité sont complexes, et encore expliqués de manière très préliminaire par les spécialistes (parmi l'ensemble des contributions des roboticiens, ce sont vraisemblablement les techniques de localisation qualitatives qui s'en inspirent le plus). Enfin, nous pouvons affirmer sans risquer de

nous tromper que seuls les hommes sont capables de se localiser sur la base de données qui leurs sont initialement fournies.

1.4.1 Trois classes de méthodes

Estimation de mouvement

Nous regroupons dans cette classe l'ensemble des techniques qui intègrent des données à très haute fréquence, *sans les mémoriser*, et qui estiment des vitesses ou des déplacements élémentaires : c'est par intégration dans le temps ou cumul des déplacements estimés que la position du robot est connue.

On trouve dans cette catégorie les techniques de localisation inertielle et odométrique, mais aussi les techniques d'estimation visuelle du mouvement ainsi que celles qui utilisent les données de profondeur sans construire de modèles de l'environnement.

Par essence, comme elles ne mémorisent aucune information sur l'environnement, ces techniques cumulent les erreurs qu'elles commettent et produisent donc des estimations de position dont l'erreur n'est pas bornée.

Recalage

Dans la classe des méthodes de recalage, nous regroupons les approches qui estiment la position du robot (ou corrigent son estimation) sur la base de représentations de l'environnement *construites et mémorisées à partir des données acquises*.

Le recalage est incontestablement la catégorie d'algorithmes de localisation où le plus de contributions ont été produites : on trouve dans cette classe les techniques basées sur la détection d'amers (par vision ou dans des images de profondeur), celles qui se basent sur un modèle numérique de l'environnement et celles qui représentent les amers ou l'environnement de manière moins explicite, telles les techniques qualitatives ou d'indexation d'images panoramiques.

Localisation absolue

Nous désignons par ce terme les méthodes qui sont capables de déterminer la position du robot dans un modèle de l'environnement *initialement connu* et fourni au robot. Il peut s'agir d'un modèle numérique de terrain construit par exemple à partir d'imagerie aérienne, ou même d'un véritable système d'information géographique de la zone d'évolution du robot.

On trouve dans cette catégorie les méthodes qui appariant le profil de l'horizon avec les profils prédits à partir d'un modèle numérique de terrain initial, ou celles qui mettent en correspondance un modèle numérique construit à bord du robot avec ce modèle initial. Des techniques topologiques pourraient aussi être adaptées à ce problème : la mise en correspondance de graphes de lieux, de régions ou de chemins étant tout à fait envisageable si l'on est capable de les construire à partir des données initiales et des données acquises par le robot. On peut également signaler ici les méthodes qui prennent place dans le cadre de la coopération multi-robots, comme par exemple l'utilisation d'un ballon dirigeable et d'un robot terrestre [Lacroix 2001b].

Cette catégorie a, jusqu'alors, connu le moins de contributions. Le problème est malgré tout de plus en plus considéré : il ne faut pas oublier que dans la grande majorité des contextes d'application que nous avons mentionnés, des informations initiales sont disponibles, mêmes si elles sont partielles, de faible résolution ou anciennes.

Notons que même si ces classes d'algorithmes nous semblent bien spécifiées, elles ne sont pas totalement exclusives. Certaines contributions au problème de la localisation peuvent être en effet difficilement cataloguées dans une seule des classes : les techniques basées sur la mise en correspondance de modèles numériques de terrain, peuvent par exemple permettre aussi bien d'effectuer des tâches de recalage que de localisation absolue.

Par ailleurs, il existe une dépendance entre les trois types de méthodes dans la plupart des approches : les techniques d'estimation du mouvement sont souvent nécessaires à l'exécution de tâches de recalage, en fournissant une estimée initiale de la position, et la localisation absolue nécessite souvent la construction de représentations à grande échelle par le robot, ce qui implique de disposer de tâches de recalage.

1.4.2 Critères de classement

Nous avons présenté trois catégories de méthodes de localisation, en les distinguant selon qu'elles mémorisent ou non des données sur l'environnement, ou qu'elles utilisent ou non un modèle initial de l'environnement.

Il existe en fait d'autres critères, présentés dans la suite de cette section, dont la considération renforce cette typologie : les propriétés sur les erreurs commises par les algorithmes, la fréquence de leur activation lors de la navigation, le niveau d'abstraction des données utilisées, en enfin la nécessité de contrôler l'acquisition des données pour que les algorithmes soient efficaces. Le tableau de la figure 1.4 résume les relations entre ces différents critères et les trois classes de méthodes considérées dans notre typologie.

Méthode Critère	Estimation de mouvement	Recalage	Localisation absolue
Évolution des erreurs	Croissance non bornée	Croissance non bornée mais peuvent décroître	Bornée
Fréquence d'activation	Très rapide	Lente	Très lente
Abstraction des données	Données brutes	Modèles d'amers ou de l'environnement	Modèles d'amers ou de l'environnement
Nécessité de contrôle	Non	Acquisition des données, modification des déplacements	Génération de déplacements dédiés

FIG. 1.4 – Typologie des méthodes de localisation.

Erreurs sur l'estimée de la position

Du point de vue de l'évolution des erreurs sur l'estimée de la position produite, les trois classes d'algorithmes se comportent très différemment.

Les méthodes d'estimation de mouvement produisent des positions dont l'erreur n'est jamais bornée et qui croissent avec le temps et la distance parcourue, quelque

soient les déplacements effectués. Pour les méthodes de recalage, ce comportement dépend des déplacements effectués. Ainsi, les erreurs croissent tant que le robot ne reperçoit pas d'éléments ou des zones précédemment modélisées : lors de l'exécution de lignes droites, les éléments perçus depuis plusieurs positions ne permettent que de limiter l'accroissement des erreurs ; par contre, la perception d'éléments modélisés depuis longtemps, lors de la réalisation de trajectoires en boucle par exemple, permet de réduire l'erreur sur l'estimée de la position. Enfin, les techniques de localisation absolue produisent une erreur toujours bornée, qui dépend essentiellement de la résolution du modèle initial.

Fréquence d'activation sur l'estimée de la position

Un autre critère qui permet de distinguer les trois classes d'algorithmes est la fréquence de leur activation. Les techniques d'estimation du mouvement doivent être exécutées de manière permanente, à des fréquences qui peuvent atteindre plusieurs dizaines de Hz . Pour permettre d'estimer ou de corriger une estimée de position, les techniques de recalage nécessitent que le robot ait effectué un déplacement significatif (de l'ordre du mètre environ) entre le moment où des éléments sont modélisés et celui où ils sont perçus à nouveau. Enfin, en ce qui concerne les techniques de localisation absolue, le modèle initial étant souvent de faible résolution il faut que le robot ait perçu une surface de l'environnement suffisamment grande pour pouvoir produire ou confirmer une estimée de position.

Notons que ce critère de fréquence temporelle d'activation des algorithmes peut bien sûr être traduit en fréquence spatiale et que, rapproché avec les erreurs sur les estimées de position, il permet de distinguer les classes d'algorithmes suivant différentes échelles. Très schématiquement, retenons que les techniques d'estimation du mouvement fournissent des estimées à une précision de l'ordre du centimètre, les techniques de recalage à une précision de l'ordre du mètre, et les techniques de localisation absolue à une précision de l'ordre de la dizaine de mètres.

Niveau d'abstraction des données

Les données manipulées sont plus ou moins structurées selon la catégorie considérée. Les techniques d'estimation du mouvement traitent des données brutes, les techniques de recalage les mémorisent, le plus souvent après structuration (modélisation des amers ou du terrain¹⁸). Quant aux techniques de localisation absolue, elles peuvent se baser sur un simple modèle numérique modèle de terrain, mais peuvent aussi nécessiter la capacité *d'identifier* la sémantique de certains éléments afin de les mettre en correspondance avec des données initiales.

Aspects liés au contrôle de l'acquisition des données

C'est le dernier critère qui permet de distinguer les trois méthodes. Pour l'estimation du mouvement, les données sont traitées au fur et à mesure qu'elles sont produites, et il n'y a aucune stratégie particulière pour contrôler leur acquisition.

Pour les méthodes de recalage, il peut être intéressant de contrôler l'acquisition des données, en s'assurant qu'un amer est, par exemple, perçu un nombre de fois suffisamment grand pour permettre de contribuer efficacement à l'estimation de la position

¹⁸Mais notons cependant que les techniques de recalage par indexation d'image se basent sur des simples attributs, et non sur une structuration des données.

ou plus généralement pour limiter la croissance de l'erreur sur l'estimée de la position [Latombe 1991, Lazanas 1992, Hait 1997]. Ce point peut parfois interférer avec les tâches de génération des déplacements, qui doivent alors prendre en compte ces considérations.

En ce qui concerne les méthodes de localisation absolue, elle peuvent nécessiter l'exécution de déplacements spécifiques, voire la mise en œuvre de stratégies de découverte d'informations, dédiés à la confirmation ou l'information d'une hypothèse de localisation.

1.5 Outils nécessaires à la localisation

Étant donnée la grande variété des contributions au problème de la localisation, il est intéressant de recenser les différents outils et formalismes auxquels elles font appel. On peut les classer en trois grandes catégories : ensembles des outils relatifs à la perception, formalismes de la théorie de l'estimation, et techniques relatives à la planification et au contrôle.

Perception

C'est sans aucun doute le domaine le plus vaste où le plus de travail est a été effectué pour la résolution du problème de la localisation.

Tout d'abord, on peut recenser les problèmes de *segmentation* et de *structuration* des données nécessaires à la construction de représentations, avec les difficultés, dans le contexte des environnements naturels, que nous avons mentionnées en introduction de ce chapitre.

Le problème général de la *mise en correspondance* est lui aussi très présent. Il peut se décliner de manières très différentes : mise en correspondance données \leftrightarrow données, données \leftrightarrow modèles et modèles \leftrightarrow modèles. Les modèles sont aussi bien des indexes calculés sur des images (simples attributs) que des graphes topologiques de l'environnement où des représentations sémantiques des objets.

Enfin, le problème de la fusion de données, qui peut faire appel à l'utilisation de théories de l'estimation, se pose fréquemment, pour mettre à jour les représentations de l'environnement notamment.

Théories de l'estimation

Les outils qui permettent d'estimer la position et une erreur associée à celle-ci sont bien évidemment nécessaires, et essentiels, à de nombreuses méthodes de localisation (à part pour les méthodes qualitatives). Ce sont en effet eux qui produisent la solution du problème, à partir des résultats des algorithmes de perception. Certains auteurs ramènent même l'étude du problème SLAM à des développements dans ce contexte [Leonard 1999, Dissanayake 1999].

Parmi ces outils, les techniques de filtrage stochastiques sont incontestablement les plus répandues — et elles s'avèrent particulièrement robustes en environnements intérieurs, sur la base de coupes laser planes [Moutarlier 1991, Alami 2000].

On trouve également les approches possibilistes, markoviennes ou basées sur théorie de l'évidence de Dempster-Shafer. Différents auteurs se sont intéressés à la comparaison de ces méthodes [Maaref 1999, Chateau 2000, Gutmann 1998], mais aucune ne semble s'imposer particulièrement. L'adéquation de telle ou telle approche dépend du contexte, de la nature des données et des imprécisions associées.

Enfin, la difficulté à satisfaire différentes hypothèses sur les modèles d'erreur sur les données a récemment conduit certains auteurs à considérer des approches ensemblistes,

où les erreurs sont représentées par des distributions uniformes à l'intérieur d'intervalles [Marco 2000, Kieffer 2000]. La mise en oeuvre de ces approches est néanmoins délicate lorsque la taille de l'état à estimer croît (à notre connaissance, au maximum 3 variables d'état sont considérées par de telles approches).

Ces techniques sont généralement connues et maîtrisées, mais leur mise en oeuvre est parfois délicate. Ainsi, les non-linéarités des équations du mouvement nécessitent l'utilisation de filtres de Kalman étendus, dont la convergence n'est pas garantie. Mais surtout, l'obtention d'un modèle d'erreur des différents estimateurs de position n'est pas toujours et les données erronées (les *fautes*) qu'ils produisent parfois doivent absolument être détectées afin de ne pas être prises en compte [Sukkarieh 1999b]. Pour fusionner les estimations de positions fournies par les différents algorithmes, l'application de ces techniques requiert donc une bonne expertise et la connaissance d'un modèle d'erreur des algorithmes.

Contrôle et planification

Enfin, nous avons mentionné que les tâches de recalage et de localisation absolue pouvaient nécessiter le contrôle de l'acquisition de données, voire même influencer les déplacements à effectuer et donc intervenir beaucoup dans la manière dont le robot mène à bien ses missions. Cela nécessite de connaître un *modèle de l'ensemble des processus* qui constituent les fonctions de recalage ou de localisation, de l'acquisition des données à la production d'une estimée de la position. C'est à partir de ce modèle et des représentations de l'environnement que l'on peut déterminer l'intérêt en terme de gain en précision de localisation de percevoir tel où tel élément de l'environnement, ou de se rendre à telle ou telle position. C'est dans le contexte d'approches probabilistes que ces problèmes (et d'une manière plus générale l'ensemble des problèmes de la "perception active" au sens de [Bajcsy 1988]) sont résolus de la manière la plus élégante [Seiz 2000, Fox 1998, Fox 1999].

1.6 Conclusions

L'ensemble des contributions présentées dans ce chapitre a montré la diversité des thèmes que le problème de la localisation permet d'aborder. Il s'agit d'une fonctionnalité fondamentale pour la navigation des robots autonomes et quasiment *tous* les algorithmes que l'on pourra trouver à bord d'un robot réel dépendent d'elle.

Le nombre et la variété des solutions proposées est cependant devenu si vaste que nous pourrions nous demander dans quelle mesure le problème n'a pas déjà été résolu. Le rythme des publications sur le sujet semble néanmoins démontrer que les chercheurs ne sont toujours pas satisfaits des réponses proposées.

En fait, il existe un problème de robustesse des algorithmes vis à vis des situations réelles. L'environnement dans lequel évolue le robot est hautement non déterministe et la réalisation de méthodes robustes face à l'ensemble des situations possibles font qu'aucun algorithme ne permet de les gérer toutes. Le problème de la localisation ne sera donc résolu que par l'utilisation d'une *panoplie* de méthodes, éventuellement redondantes, permettant chacune de gérer des situations particulières. Pour la navigation à long terme, il est par exemple nécessaire de disposer d'au moins une méthode dans chaque catégorie de la typologie que nous avons exhibée. Malgré cela, la suite du document montrera que la localisation est en général beaucoup plus robuste si l'on dispose de plusieurs méthodes dans chaque classes.

De plus, l'aspect hétérogène des méthodes nécessaires fait que le problème de l'intégration de celles-ci, au sein d'une architecture informatique permettant leur coopération, est également fondamental et complexe : cette thèse a été effectuée dans cet esprit et le chapitre 5 présentera les difficultés liées à cet aspect de la localisation. À notre connaissance, la littérature fournit peu de contributions présentant un tel niveau d'intégration.

Cependant, il est illusoire d'espérer résoudre un jour le problème de la localisation de façon complète et exhaustive. Il existe de nombreux cas où le robot ne *pourra pas* se localiser, tout simplement à cause du manque d'information. Ainsi, un désert exhibera un nombre très insuffisant d'amers, ou de zones caractéristiques, qui permettraient l'utilisation de méthodes de recalage et de localisation absolue. Il est donc indispensable de concevoir des algorithmes permettant de supporter des erreurs importantes sur la position. Le chapitre 4 propose une méthode de construction de cartes allant dans ce sens.

II

L'odométrie

L'odométrie est certainement la méthode de localisation la plus couramment employée pour les robots disposant d'une structure de locomotion à roues. Son principe consiste à déduire une position, de façon incrémentale, à partir de la vitesse et de la géométrie des roues. La mise en œuvre de cette méthode est des plus simples et ne nécessite qu'une puissance de calcul très limitée.

Cependant, l'odométrie est fragile. Tout d'abord, elle nécessite une connaissance précise de la géométrie du robot : le diamètre des roues doit bien sûr être déterminé, mais d'autres dimensions, comme l'entraxe ou le point de contact avec le sol, doivent également être prises en compte. Ces paramètres sont généralement difficiles à obtenir de façon précise et, en terrains naturels, dépendent fortement de la nature du sol. De plus, les cas pathologiques sont rarement détectables : les glissements ou patinages sont par exemple dramatiques pour l'estimation de la position et ils ne peuvent en général pas être mesurés directement. Selon le type de terrain ils peuvent rendre quasiment inutile l'odométrie en tant que méthode de localisation.

Celle-ci est pourtant incontournable car, outre sa fonction d'estimateur de position, elle permet de réaliser un contrôle très fin des déplacements des plateformes. La position locale est fournie à haute fréquence, autorisant ainsi l'exécution de déplacements élémentaires simples.

Ce chapitre présente les points clefs devant être considérés pour le développement d'une odométrie pour un châssis de type « Marsokhod » comme celui de *Lama*. Il se contente cependant de présenter les difficultés liées à ce type de châssis et aux environnements naturels, sans reprendre les considérations générales de l'odométrie.

La première section explique la commande employée pour des châssis dépourvus de roues directionnelles et la seconde section pose les équations permettant d'établir l'estimation de position dite *odométrique*, en deux dimensions.

Cette base, minimale, ne fournit cependant pas une précision satisfaisante : la section 2.3 montre ses limites, tant du point de vue de l'estimation de position que de celui du contrôle des déplacements. L'ajout d'un compas magnétique, pour améliorer l'estimation de position, et celui d'un gyromètre, pour le contrôle des déplacements,

sont présentés dans la section 2.4. L'utilisation d'inclinomètres permet d'obtenir une position 3-D et les équations de cette odométrie sont présentées dans la section 2.5.

Enfin, une ouverture sur l'analyse du comportement de l'odométrie sur le robot *Lama* et des pistes pour sa qualification *en ligne* concluent le chapitre.

2.1 Commande des châssis de type « Marsokhod »

Le robot *Lama* est un robot mobile de type « Marsokhod » [Kemurdjian 1992] : six roues motrices *indépendantes* et *non-directionnelles* lui permettent de se déplacer. Chacune de ces roues peut être commandée en marche avant ou en marche arrière, dans une gamme continue de vitesse de 0 à environ 17 cm.s^{-1} (vitesse linéaire des essieux). L'annexe A contient une description détaillée du robot et des instruments et capteurs installés à son bord.

De façon naturelle, un tel châssis se commande en vitesse ; sa non-holonomie intrinsèque rend les commandes en position complexes et celles-ci sont généralement produites par des planificateurs de trajectoires, à un niveau supérieur.

L'indépendance des roues permet, au niveau d'un essieu isolé, d'obtenir très simplement un mouvement de rotation grâce à l'application de vitesses *différentes* sur chacune des roues. Ainsi, un essieu exécute à chaque instant une trajectoire *circulaire* autour d'un point fixe : le centre instantané de rotation. La modification dynamique des vitesses permet de changer la position de ce point fixe et d'exécuter tous types de trajectoires compatibles avec la structure du châssis.

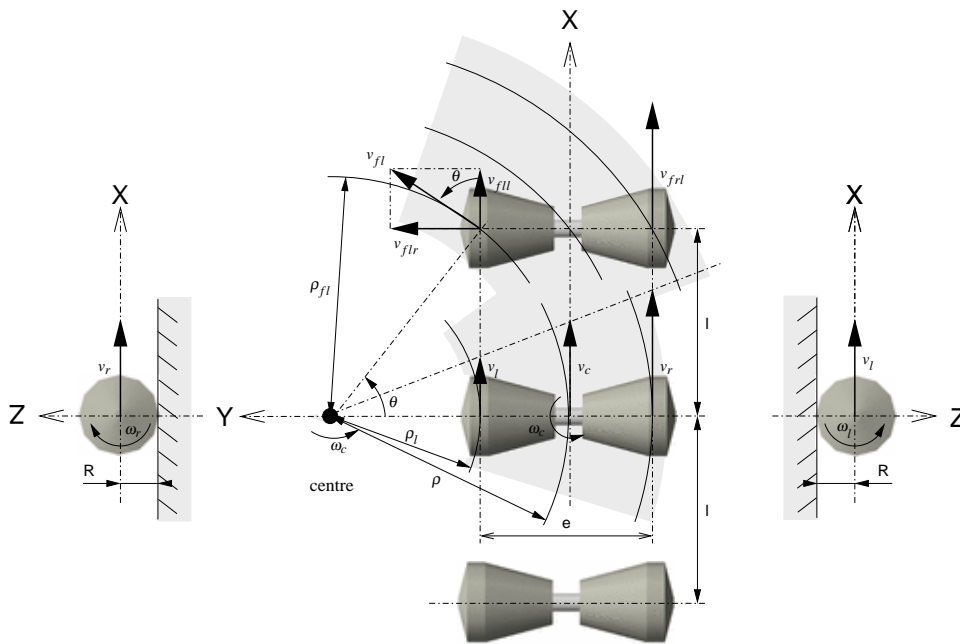


FIG. 2.1 – Commande d'un châssis dépourvu de roues directionnelles : un différentiel de vitesse sur les roues centrales gauches et droites provoque une rotation de cet essieu. Il est alors nécessaire d'appliquer sur les essieux avant et arrière des vitesses compatibles avec ce mouvement pour autoriser la rotation globale du châssis. La connaissance de la géométrie des roues — le rayon R au niveau du contact avec le sol et l'entraxe e entre les points de contact — permet de calculer les vitesses linéaires v_l et v_r des roues centrales. La vitesse du châssis v_c et ω_c se déduit de ces vitesses.

La figure 2.1 illustre ce principe : deux vitesses ω_l et ω_r sont appliquées sur les roues gauche et droite de l'essieu central, provoquant la rotation du châssis à la vitesse

angulaire $\omega_c = v_c/\rho$, où ρ est le rayon instantané de gyration. L'hypothèse de roulement sans glissement des roues permet d'exprimer la vitesse du centre du robot en fonction des vitesses des roues centrales, de leur diamètre R et de l'entraxe e^1 :

$$\begin{aligned} v_c &= R.(\omega_r + \omega_l)/2 \\ \omega_c &= R.(\omega_r - \omega_l)/e \end{aligned} \quad (2.1)$$

L'absence de roues directionnelles fait que seules les roues *centrales* provoquent un mouvement de rotation ; les paires de roues avant et arrière ne peuvent participer qu'au seul mouvement de translation. Le choix de leurs vitesses doit donc être compatible avec le mouvement de l'essieu central du robot.

Sur la figure 2.1, le cas de la roue avant gauche est représenté : la vitesse v_{fl} est la vitesse *commandable* et la vitesse v_{fl} est la vitesse *réelle*, imposée par l'essieu central. Le châssis étant rigide, tout point de celui-ci se déplace à une vitesse $v = \rho.\omega_c$, ρ étant le rayon instantané de gyration du point considéré. La vitesse compatible avec le mouvement du robot doit donc être telle que, pour cette roue :

$$v_{fl} = v_{fl}.\cos(\theta) = \rho_{fl}.\cos(\theta).\omega_c = \rho_l.\omega_c = v_l$$

Un raisonnement similaire sur les autres roues montrerait que la meilleure vitesse de commande pour une roue externe (avant ou arrière) est la même que celle de la roue centrale située du même côté. Les déplacements sont donc effectués à la manière d'un engin à chenilles, en appliquant la *même* vitesse ω_l aux trois roues de gauche et la *même* vitesse ω_r aux trois roues de droite.

Un autre mode de déplacement — dit *péristaltique* — est possible avec le châssis du robot *Lama* [Andrade-Barosso 2000]. Il permet de déplacer chaque essieu indépendamment des autres, et ainsi de garder quatre des six roues toujours fixes. Ce mode fournit une meilleure *traction*, en interdisant toutefois les rotations : il est avant tout destiné aux zones très pentues ou meubles. Il n'est pas présenté dans ce document car il n'est pas encore utilisé de façon *automatique* à bord du robot. Néanmoins, l'odométrie que l'on pourrait réaliser avec ce mode serait très similaire à celle présentée dans ce chapitre.

2.2 Calcul de la position odométrique

La position odométrique du robot s'obtient par cumul de *déplacements élémentaires*, calculés *périodiquement* tous les Δt . La fréquence du calcul doit être suffisamment élevée pour que la vitesse des roues puisse être supposée *constante* entre deux périodes et les déplacements élémentaires peuvent alors être approximés par des *arcs de cercles* le long desquels le robot parcourt une distance Δs_a pour une variation angulaire $\Delta\theta$ comme illustré par la figure 2.2. La relation liant Δs_a et $\Delta\theta$ aux vitesses des roues se déduit des relations (2.1) [Ferrand 1991] :

$$\begin{aligned} \Delta s_a &= \Delta t.R.(\omega_r + \omega_l)/2 \\ \Delta\theta &= \Delta t.R.(\omega_r - \omega_l)/e \end{aligned} \quad (2.3)$$

¹L'entraxe se définit comme la distance moyenne entre la surface de contact avec le sol et dépend, en toute rigueur, de la nature de ce dernier.

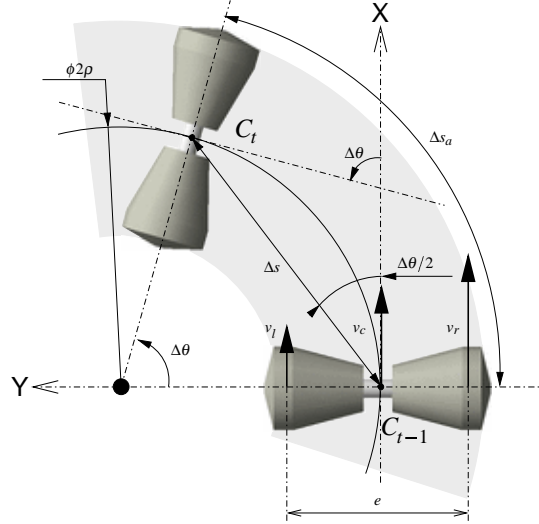


FIG. 2.2 – Calcul de la position odométrique : connaissant les vitesses de chacune des roues, il est possible de calculer périodiquement un déplacement élémentaire (représenté de façon agrandie ici) que l'on peut approximer par un arc de cercle de longueur Δs_a et de variation angulaire $\Delta\theta$. La longueur Δs de la corde de cet arc peut, la plupart du temps, être approxmée par la longueur Δs_a de l'arc. Les déplacements élémentaires sont cumulés et la position 2-D peut ainsi être exprimée dans un repère global.

En l'absence d'autres informations, le mouvement du robot est supposé *plan* et *horizontal* et la figure 2.2 permet de remarquer que le déplacement élémentaire $(\Delta x, \Delta y)$ s'exprime en fonction de Δs et $\Delta\theta$ dans le repère lié à la position à l'instant $t - 1$:

$$\begin{aligned}\Delta x &= \Delta s \cdot \cos(\Delta\theta/2) \\ \Delta y &= \Delta s \cdot \sin(\Delta\theta/2)\end{aligned}$$

L'odométrie fournit la mesure de la longueur Δs_a de l'arc de cercle parcouru. Il est donc nécessaire de calculer la translation Δs en fonction de Δs_a :

$$\Delta s = \begin{cases} \Delta s_a & \text{si } \Delta\theta = 0, \\ \frac{2\Delta s_a}{\Delta\theta} \sin\left(\frac{\Delta\theta}{2}\right) & \text{si } \Delta\theta \neq 0. \end{cases} \quad (2.4)$$

Comme on le voit, ce calcul introduit une singularité pour le cas de la ligne droite ($\Delta\theta = 0$) et il est courant de confondre Δs et Δs_a pour éviter des problèmes d'instabilité numérique lorsque $\Delta\theta$ est petit. Cette approximation est d'autant plus valable que la fréquence d'échantillonnage des vitesses est grande devant la vitesse du robot : sur *Lama* l'erreur commise est inférieure à 4.10^{-4} % et s'avère donc négligeable.

La position 2-D (x, y, θ) du robot s'obtient ensuite de façon triviale par cumul du déplacement élémentaire avec la position à l'instant $t - 1$, exprimée dans un repère global :

$$\begin{aligned}x_t &= x_{t-1} + \Delta s \cdot \cos(\theta_{t-1} + \Delta\theta/2) \\ y_t &= y_{t-1} + \Delta s \cdot \sin(\theta_{t-1} + \Delta\theta/2) \\ \theta_t &= \theta_{t-1} + \Delta\theta\end{aligned} \quad (2.5)$$

La seule connaissance des vitesses de *consigne* appliquées sur chacune des roues permet donc d'obtenir une estimation de position. En cela, l'odométrie est puissante car elle ne requiert *aucun* capteur ou instrument particuliers.

Cependant, la section suivante montre que la consigne appliquée sur les moteurs n'est pas toujours bien suivie, et il est donc préférable d'utiliser l'information de vitesse retournée par des codeurs odométriques installés sur les roues, plutôt que d'utiliser la vitesse de consigne brute. Idéalement, ces codeurs devraient être distinct des roues motrices² afin d'éviter les glissements et de réduire la surface d'appui pour obtenir un entraxe le plus constant possible. Le robot *Lama* n'étant pas conçu de la sorte, [Andrade-Barosso 2000] a proposé une commande sous la forme de « roues molles » permettant de contrôler les glissements et améliorer la précision dans les terrains meubles.

2.3 Contrôle du mouvement

Le mode particulier de déplacement des châssis de type « Marsokhod » engendre, au cours des rotations, de fortes *contraintes* entre le sol et les roues. En effet, seul des glissements *latéraux* permettent à l'ensemble du châssis de pivoter ; plus le sol est meuble et plus il *s'oppose* à ce mouvement. On peut remarquer sur la photographie de la figure 2.3 que la conception des roues tient compte de cette contrainte : les ailettes destinées à améliorer la traction sont *orientées* de façon à ne pas créer une résistance supplémentaire au cours de rotations *moyennes*.



FIG. 2.3 – Conception des roues du châssis de Lama : les ailettes destinées à améliorer la traction sont orientées de façon à limiter les contraintes exercées par le sol au cours des rotations.

2.3.1 Contrôle de la vitesse de rotation des roues

La résistance à la rotation et le poids important de la plateforme font qu'il est nécessaire de *contrôler* et *corriger* en permanence la vitesse des six roues. Pour cela, des *codeurs odométriques* précis sont installés (présentés en annexe A.2.2) et la mesure fiable des vitesses de rotation de chacune des roues est rendue possible. Cette mesure permet de réaliser un *asservissement* en vitesse et garantit ainsi une exécution correcte des consignes commandées comme illustré par la figure 2.4.

On peut remarquer que l'estimation de la vitesse de rotation des roues n'est pas toujours évidente : selon la résolution des codeurs odométriques, l'estimation des vitesses faibles pourra s'avérer impossible. L'installation sur *Lama* de codeurs précis a néanmoins permis de résoudre ce problème.

²on parle souvent de « roues folles ».

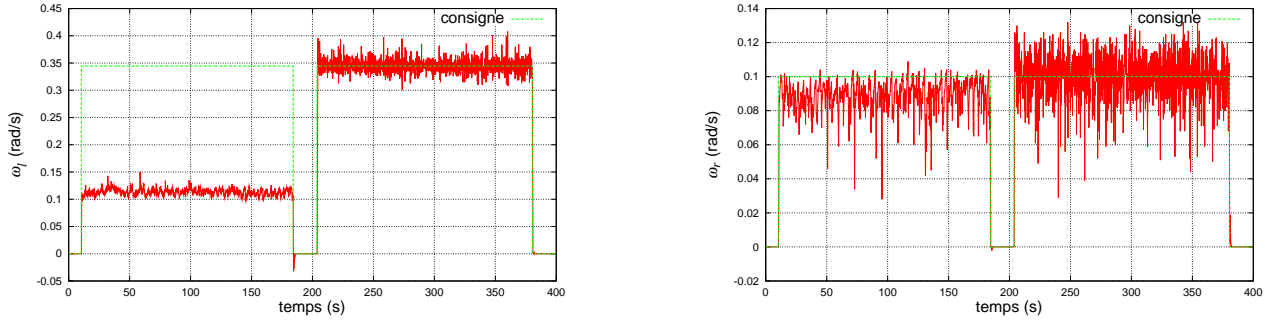


FIG. 2.4 – Asservissement de type PID sur les vitesses des roues : ces deux diagrammes montrent la vitesse de rotation instantanée des roues centrales (roue gauche sur le diagramme de gauche et roue droite sur le diagramme de droite) lors d’un mouvement circulaire vers la droite. Au cours d’une première phase, le robot évolue sans asservissement particulier : on voit nettement que la vitesse mesurée est inférieure à la consigne demandée, surtout dans le cas de la roue gauche (roue extérieure dans cette expérience) qui subit le plus de contraintes. Lors d’une deuxième phase, un asservissement de type PID est réalisé : les vitesses effectives correspondent alors aux consignes.

2.3.2 Contrôle de la vitesse de rotation du châssis

Une fois le contrôle correct des vitesses individuelles des roues effectué, il est intéressant d’observer le *comportement* du robot et le résultat de l’estimation de *position* odométrique. Pour cela, une expérience particulière est présentée dans cette section. Celle-ci a, historiquement, révélé différents problèmes liés à l’odométrie et elle servira donc de fil conducteur à la présentation des solutions à ces problèmes.

L’expérience est la suivante : une consigne $v_c = 5 \text{ cm.s}^{-1}$ et $w_c = 0.05 \text{ rad.s}^{-1}$ constante est exécutée. La trajectoire théorique résultant de cette commande est un cercle de diamètre 2 m et le robot a été stoppé après l’exécution d’un tour complet. La figure 2.5 compare la trajectoire réelle, mesurée par GPS (voir annexe A.2.5), et la position odométrique.

Cette figure exhibe trois phénomènes remarquables. D’une part, le robot *n’exécute pas correctement* la trajectoire puisque le GPS montre qu’un cercle de plus de 3 m de diamètre a été réalisé (au lieu des 2 m théoriques). La *position* déduite des seules informations odométriques apparaît donc très *erronée*, mais *correspond* toutefois à la consigne demandée ; ce point permettant d’affirmer que l’asservissement en vitesse des chacune des roues est correct. Enfin, l’abscisse curviligne mesurée le long de la trajectoire se trouve être quant à elle très précise sur le terrain *favorable* sur lequel a évolué le robot lors de l’expérience.

Le fait que le robot ne suive pas la trajectoire théorique s’explique par la résistance au glissement latéral des roues avant et arrière qui empêchent le robot de pivoter sur lui-même³. La trajectoire réalisée aura donc, en général, un rayon de courbure plus *faible* que celui initialement demandé.

Cette expérience permet donc de constater que l’odométrie implémentée permet d’avoir une estimation *relativement* fiable de l’abscisse curviligne (ne dépendant que de la nature du sol et de l’importance du patinage).

Par contre, on remarque qu’un asservissement individuel de chacune des six roues *n’est pas* suffisant : il est indispensable de corriger *dynamiquement* le mouvement global du robot et, en particulier, sa vitesse instantanée de rotation pour obtenir un contrôle satisfaisant. Le calcul de la vitesse de rotation *globale* du robot à partir de la seule information de vitesse des roues *n’apparaissant pas* satisfaisant, il est indispensable d’obtenir cette information — afin de la corriger — par un autre moyen.

³un glissement longitudinal se traduirait par une trajectoire plus courte, ce qui n’est pas le cas.

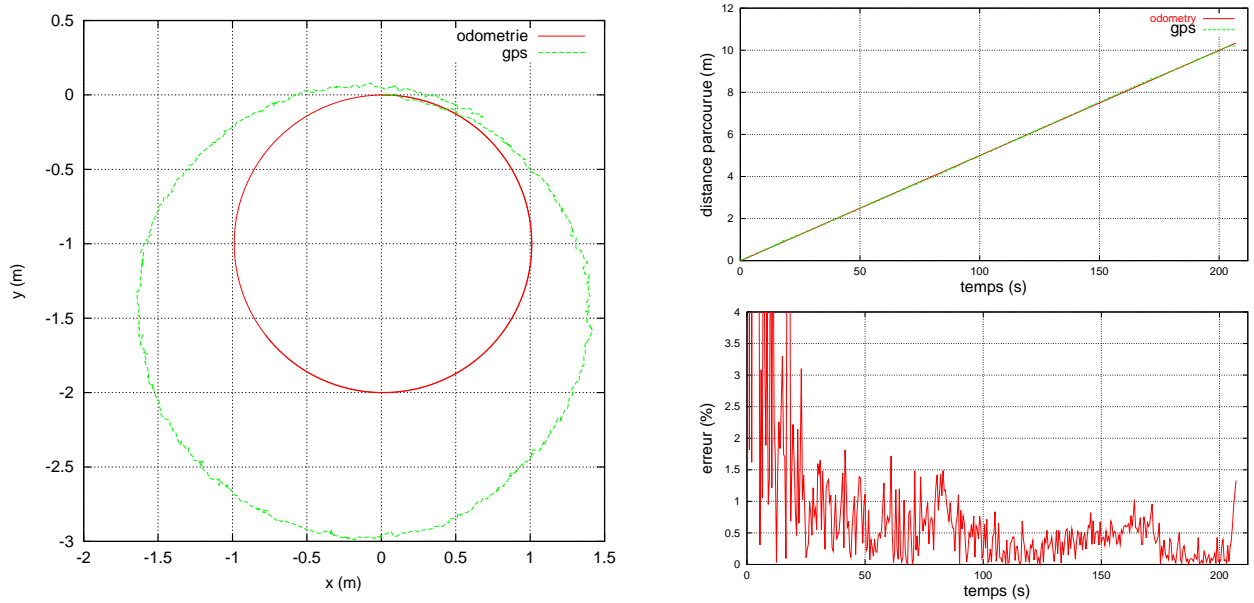


FIG. 2.5 – Exécution d'un cercle sans asservissement global : la trajectoire calculée par l'odométrie seule (petit cercle, à gauche) correspond effectivement au cercle de diamètre demandé (grâce à l'asservissement individuel des roues). En revanche, la trajectoire réalisée (mesurée par GPS) ne correspond pas du tout à la commande. On peut, d'autre part, constater que l'abscisse curviligne mesurée par l'odométrie est très précise en l'absence de glissement : le graphique en bas à droite montre que l'erreur relative est de l'ordre de 0.5% (le bruit observable pendant les 25 premières secondes provient de la faible distance parcourue devant les erreurs du GPS). Ce résultat peut paraître surprenant au regard des deux courbes de gauche, mais il faut remarquer que lorsque le GPS mesure qu'un peu plus d'un tour complet a été effectué (ce que le robot a effectivement fait), l'odométrie mesure quant à elle quasiment 2 tours (qui ne sont pas visibles sur la figure car parfaitement superposés).

La section suivante présente deux capteurs supplémentaires classiquement utilisés dans ce but : le compas, qui permet d'améliorer le calcul de la position, et le gyromètre, qui permet d'améliorer le contrôle et l'exécution des trajectoires.

2.4 Utilisation d'un compas et d'un gyromètre

L'ajout d'un compas permet de disposer d'une référence *absolue* et sans dérive pour l'orientation du robot. Le compas magnétique monté sur *Lama* est présenté dans l'annexe A.2.3 ; il fournit un angle de cap par rapport au nord, dans un plan *horizontal*, à une résolution de 0.1° et une précision de $\pm 1^\circ$, toutes les 100 ms .

Le gyromètre permet quant à lui de mesurer une *vitesse angulaire*. Par intégration, il permet également d'en déduire une orientation, exprimée cette fois dans le plan du robot. Contrairement au compas, l'angle ainsi calculé est sujet à une dérive au cours du temps et il est donc moins *fiable* à long terme. Le gyromètre à fibre optique monté sur *Lama* est décrit dans l'annexe A.2.4 ; il fournit une vitesse angulaire avec une résolution de $0.00305\text{ deg.s}^{-1}$, toutes les 100 ms , et une dérive de 0.3° par heure.

2.4.1 Estimation de la position

Comme l'a montré la section précédente, la variation angulaire élémentaire $\Delta\theta$ de l'équation (2.5) n'est pas correctement estimée par l'odométrie seule et la position ainsi calculée s'en trouve erronée. Le compas ou le gyromètre peuvent être utilisés pour réaliser cette mesure *directement* : on obtient alors une estimation de position en utilisant le calcul du déplacement odométrique Δs intégré le long de la variation

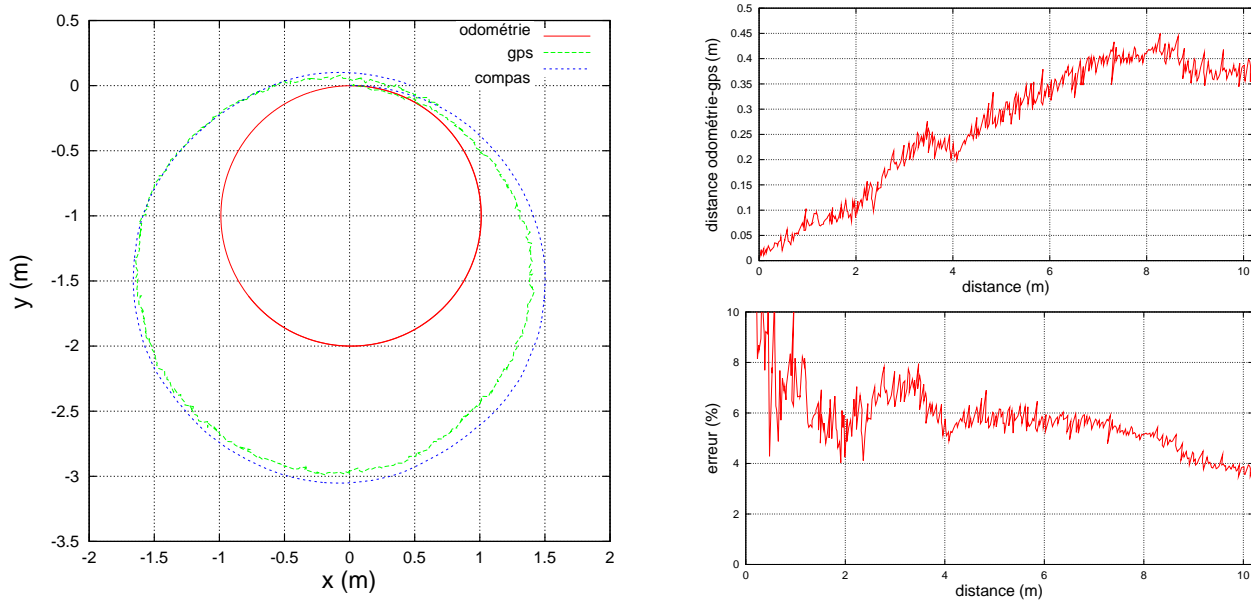


FIG. 2.6 – Comparaison des positions odométriques, compas et GPS. L'expérience de la figure 2.5 est réitérée, en mesurant cette fois les variations angulaires élémentaires à l'aide du compas et les déplacements à l'aide de l'odométrie. On constate que la position ainsi calculée est bien meilleure : les graphiques de droite montrent la distance entre la position GPS et la position déduite du compas. L'erreur commise est approximativement de 5%. En revanche, la trajectoire n'est toujours pas correctement réalisée puisque la commande est restée inchangée.

angulaire $\Delta\theta$ mesurée.

La figure 2.6 montre le résultat de l'expérience de la figure 2.5 où la position calculée avec l'aide du compas est superposée : l'amélioration sur la qualité de la position est nette — mais la trajectoire reste toujours aussi mal exécutée puisque ces nouvelles données n'ont pas encore été intégrées à la commande.

Étant donné que le compas fournit un angle *sans dérive*, il semble *a priori* préférable d'utiliser ce dernier plutôt que le gyromètre pour mesurer les variations angulaires. Cependant, la sensibilité du compas aux perturbations électromagnétiques et aux masses métalliques font que la précision de la mesure dépend de la qualité de son calibrage (voir l'annexe A.2.3). Le choix entre compas et gyromètre est donc difficile, et dépend avant tout du matériel utilisé. Sur *Lama*, les deux options sont possibles ; sur des expérimentations de faibles durées — jusqu'à une vingtaine de minutes — la dérive du gyromètre est négligeable et, compte tenu des difficultés liées au calibrage du compas, l'utilisation du gyromètre est préférable. L'idéal serait d'utiliser le compas comme un moyen d'éviter la dérive du gyromètre, en fusionnant les deux angles calculés.

2.4.2 Asservissement du mouvement

Un autre point amélioré par une instrumentation supplémentaire concerne l'exécution des trajectoires : la section précédente a montré qu'il est indispensable d'asservir de façon dynamique le mouvement global du robot sur la consigne choisie. Pour cela, il est cette fois nécessaire d'obtenir la vitesse de rotation instantanée ω_c du châssis.

Le compas n'est d'aucune utilité pour cette tâche, comme le montre la figure 2.7 : la dérivée de l'angle de cap en fonction du temps — *i.e.* la vitesse angulaire — donne un résultat quasiment inexploitable car excessivement bruité. Ce niveau impressionnant de bruit pourrait certainement être filtré, mais plusieurs tentatives ont été faites sans parvenir à une solution satisfaisante. De plus, il *ne semble pas* pertinent de vouloir inférer

une grandeur qui n'est pas mesurée explicitement : le compas est conçu pour mesurer des angles ou des variations angulaires et la vitesse est une grandeur physiquement différente. Le gyromètre semble donc être ici le capteur à utiliser.

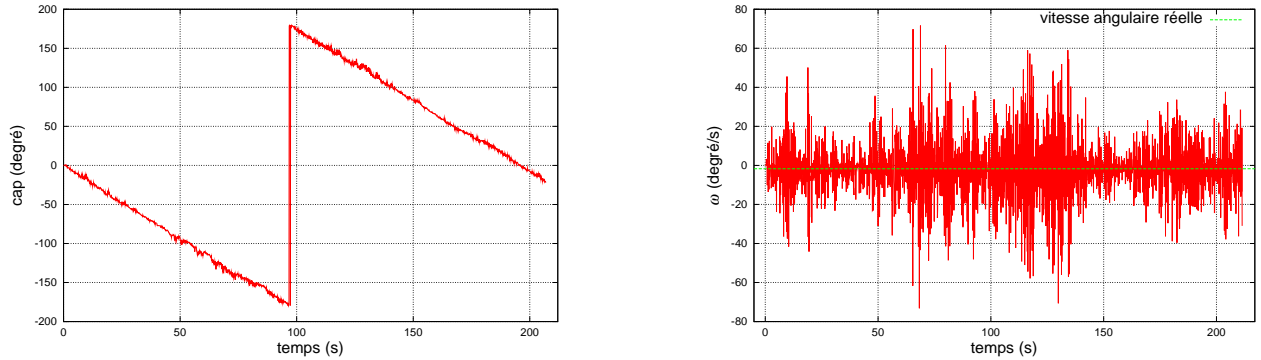


FIG. 2.7 – Calcul de la vitesse angulaire à partir du cap donné par le compas. Sur le graphique de gauche, on observe l'évolution du cap en fonction du temps, au cours d'une rotation du robot. L'imprécision de mesure, perturbée par les vibrations et les courants, est telle que l'utilisation de la vitesse angulaire déduite de la variation angulaire élémentaire est quasi-impossible (graphique de droite).

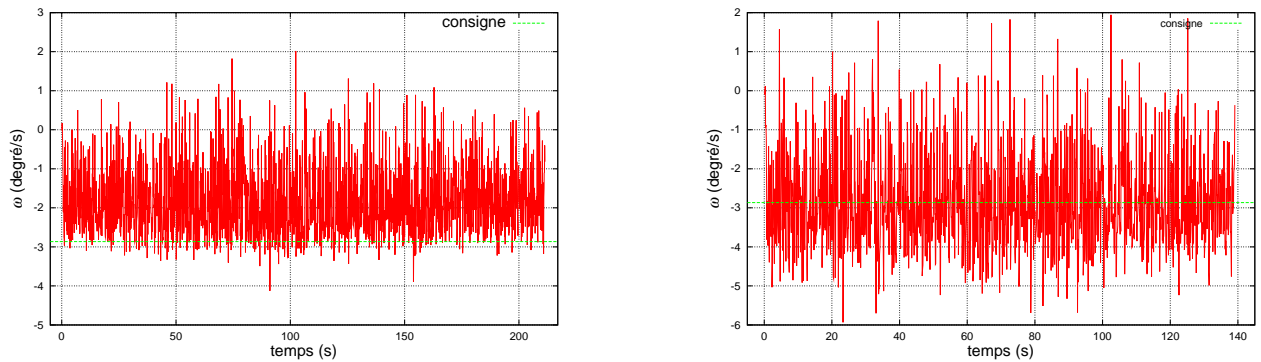


FIG. 2.8 – Vitesse angulaire globale mesurée par gyromètre. Le graphique de gauche correspond à la trajectoire de la figure 2.5 où le mouvement a été réalisé sans asservissement global : la vitesse angulaire était effectivement inférieure à la consigne demandée, produisant ainsi une trajectoire erronée. Le graphique de droite correspond à la trajectoire de la figure 2.9, réalisée avec asservissement global de type PID : la consigne est suivie en moyenne, malgré le bruit relativement élevé du gyromètre.

À partir de la vitesse angulaire ω_c , l'asservissement du mouvement global consiste à calculer une consigne virtuelle, déduite de la mesure de la vitesse de rotation réelle et de la consigne demandée. Les vitesses des roues sont à leur tour modifiées, pour exécuter cette consigne virtuelle. Bien-sûr, cette procédure *invalide* complètement le calcul *odométrique* de la variation angulaire du robot $\Delta\theta$ et l'utilisation du compas ou du gyromètre devient *indispensable* pour obtenir un calcul de position correct.

Un retour sur l'expérience de la figure 2.5 permet de constater que la mise en place de l'asservissement global permet au robot d'évoluer selon la trajectoire demandée (figures 2.8 et 2.9). Grâce au calcul de la position par compas, on obtient ainsi une première odométrie *2-D*, satisfaisante tant du point de vue du *contrôle* que de l'estimation de *position*.

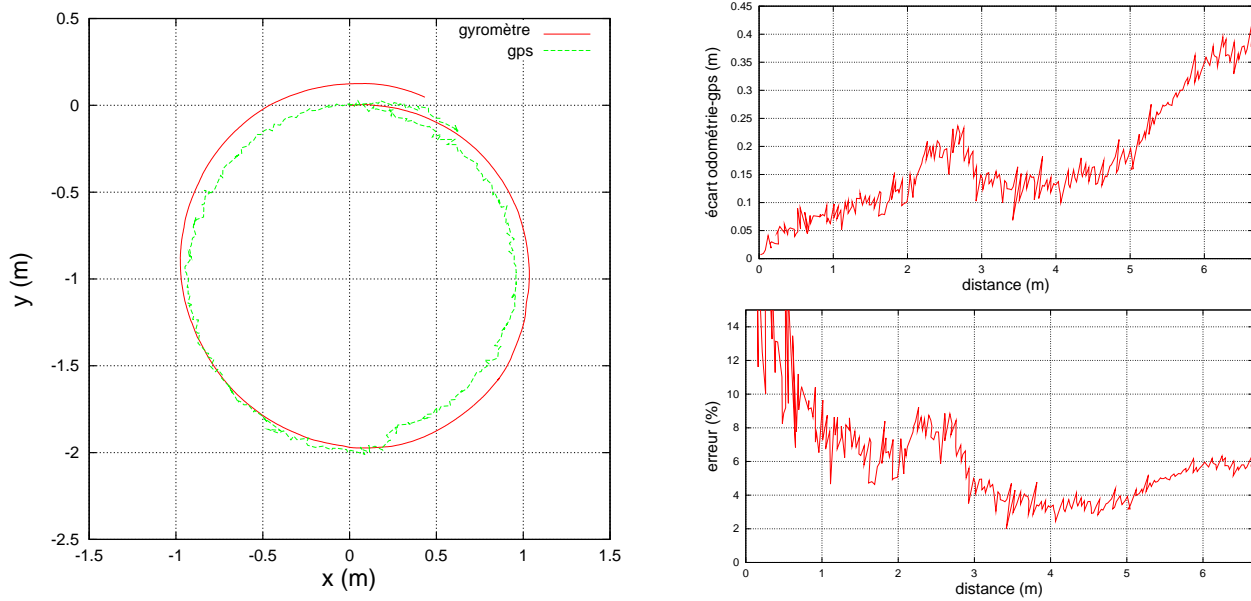


FIG. 2.9 – Comparaison des positions odométriques (avec gyromètre) et GPS avec asservissement global : sur le graphique de droite on peut remarquer que la trajectoire réalisée est approximativement un cercle de diamètre 2 m, et que la position calculée est relativement bonne. Les courbes à droite montrent la distance entre la position odométrique et la position fournie par le GPS. L'erreur relative est de l'ordre de 5 %.

2.5 Utilisation d'inclinomètres pour une position 3-D

Le robot *Lama* est équipé de deux inclinomètres mesurant le site et le gîte instantanés de l'essieu central. Utilisés conjointement à l'odométrie 2-D, ils permettent de calculer une position 3-D.

Une attention particulière doit être apportée à la définition des angles utilisés : selon la représentation choisie, la position s'exprimera de façon plus ou moins intuitive.

[Roumeliotis 1999] propose par exemple l'utilisation de quaternions, affirmant qu'ils permettent d'éliminer toute singularité dans la représentation de la position et que l'attitude du robot s'exprime ainsi de façon algébrique, sans l'aide de fonctions transcendentes. Cette représentation est particulièrement adaptée à l'utilisation d'un gyromètre qui mesure, généralement, un vecteur rotation normal au plan du robot.

Dans le cas du robot *Lama*, tous les capteurs utilisés à l'exception du gyromètre mesurent des angles (ou des rotations) définis à partir du plan (X, Y) horizontal. Les inclinomètres fournissent un site ϕ et un gîte ψ respectivement définis comme une rotation autour des axes Y et X (voir la figure 2.10) et le compas mesure un cap θ défini comme une rotation autour de l'axe Z de ce même repère⁴. Les angles θ , ϕ ou ψ utilisés dans cette section *ne sont donc pas* les angles d'Euler classiques.

La représentation choisie ici permet l'utilisation directe des angles lus sur les instruments et fournit une expression très simple de la position où les seules fonctions transcendentes sont les fonctions trigonométriques *sin* et *cos*. En revanche, l'expression de cette position à l'aide d'angles d'Euler est plus complexe [Bonnifait 1997, Ferrand 1991].

Les déplacements élémentaires $(\Delta x, \Delta y, \Delta z)$ s'expriment en fonction de l'abscisse curviligne Δs , de la variation d'angle de cap $\Delta\theta$ et de l'inclinaison $\Delta\phi$ du plan dans

⁴d'autres modèles de compas peuvent fournir un angle équivalent à celui du gyromètre, dans le plan du robot.

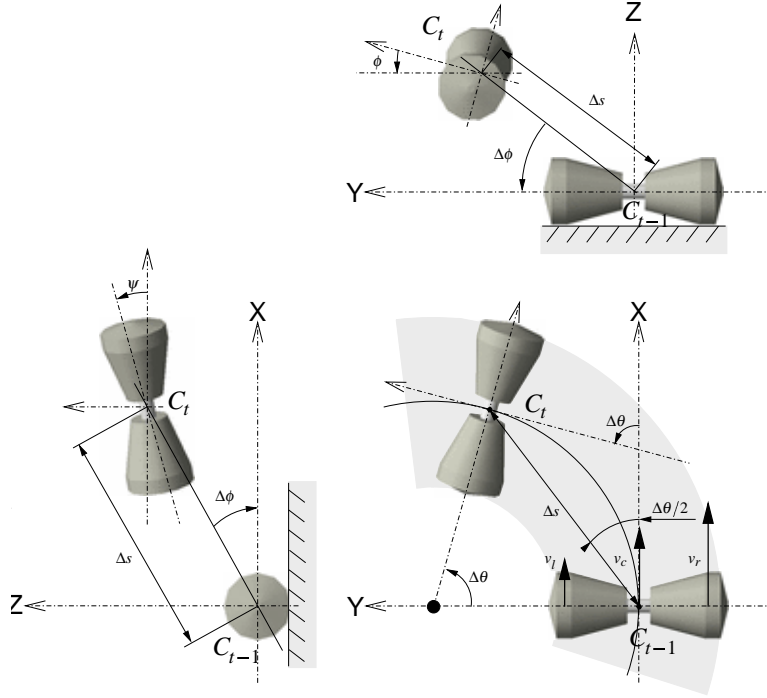


FIG. 2.10 – Calcul de la position odométrique 3-D : contrairement au cas 2-D représenté sur la figure 2.2, le robot évolue dans un plan qui n'est plus horizontal. L'inclinaison moyenne $\Delta\phi$ de ce plan est souvent approximée par la moyenne des angles de site entre deux positions successives. On constate par ailleurs que l'angle de gîte ψ n'influence pas la position de l'essieu.

lequel le robot évolue, et on remarque que la représentation choisie fait que l'angle de gîte ψ n'influence pas les autres paramètres de la position du robot :

$$\begin{aligned}\Delta x &= \Delta s \cdot \cos(\Delta\theta/2) \cdot \cos(\Delta\phi) \\ \Delta y &= \Delta s \cdot \sin(\Delta\theta/2) \cdot \cos(\Delta\phi) \\ \Delta z &= -\Delta s \cdot \sin(\Delta\phi)\end{aligned}$$

L'inclinaison $\Delta\phi$ peut se calculer grâce à l'approximation consistant à dire que durant le laps de temps $[t; t+1]$ le robot évolue dans un plan dont le site est la moyenne des sites aux instants t et $t+1$. Les angles de site et de gîte sont directement lus sur les inclinomètres et la position (x, y, z, θ) s'obtient alors à l'aide des équations suivantes :

$$\begin{aligned}x_t &= x_{t-1} + \Delta s \cdot \cos(\theta_{t-1} + \Delta\theta/2) \cdot \cos\left(\frac{\phi_t + \phi_{t-1}}{2}\right) \\ y_t &= y_{t-1} + \Delta s \cdot \sin(\theta_{t-1} + \Delta\theta/2) \cdot \cos\left(\frac{\phi_t + \phi_{t-1}}{2}\right) \\ z_t &= z_{t-1} - \Delta s \cdot \sin\left(\frac{\phi_t + \phi_{t-1}}{2}\right) \\ \theta_t &= \theta_{t-1} + \Delta\theta\end{aligned}\tag{2.6}$$

Ces équations forment l'odométrie du robot *Lama* telle qu'elle est implémentée aujourd'hui. Dans la pratique, un filtrage à moyenne glissante est utilisé pour la lecture des inclinomètres : selon le terrain, le bruit sur ces capteurs peut être important. Ce filtre n'est pas présenté ici car une étude plus poussée doit être effectuée pour en valider la nécessité et en déterminer les spécifications.

2.6 Comportement de l'estimation de position

Les sections précédentes ont montré que l'estimation de position de l'odométrie peut être très précise lorsqu'elle est utilisée dans un contexte adapté. Son principe fait cependant l'hypothèse qu'il est possible de convertir les rotations des roues en un déplacement linéaire du châssis, c'est-à-dire que le robot roule sans glisser sur le sol. Dans la pratique, cette hypothèse n'est pratiquement jamais vérifiée et la position obtenue est d'autant plus inexacte que ces glissements sont importants.

De plus, les équations de la position 3-D (2.6) ou 2-D (2.5) montrent qu'un phénomène de cumul de l'erreur est inévitable : l'imprécision sur chaque capteur se répercute sur la position du robot, de façon parfois très importante si l'on considère les bras de leviers introduits par les erreurs sur les rotations.

Le principal problème de l'odométrie est donc sa qualification : déterminer la validité de la position calculée et établir un modèle de l'erreur commise est indispensable dès que l'on considère des applications réalistes pour lesquelles des enjeux de robustesse ou de sûreté de fonctionnement sont primordiaux. De plus, l'utilisation conjointe avec d'autres méthodes de localisation nécessitera l'utilisation d'un tel modèle, comme le montreront les chapitres suivants.

Cette section se propose d'ouvrir quelques pistes allant dans le sens d'une qualification *en ligne* de l'odométrie, sur la base des nombreuses expérimentations réalisées sur le robot *Lama*. Nous recensons la somme des observations que nous avons pu obtenir, afin de poser les bases d'un travail futur qu'il nous semble pertinent de mener.

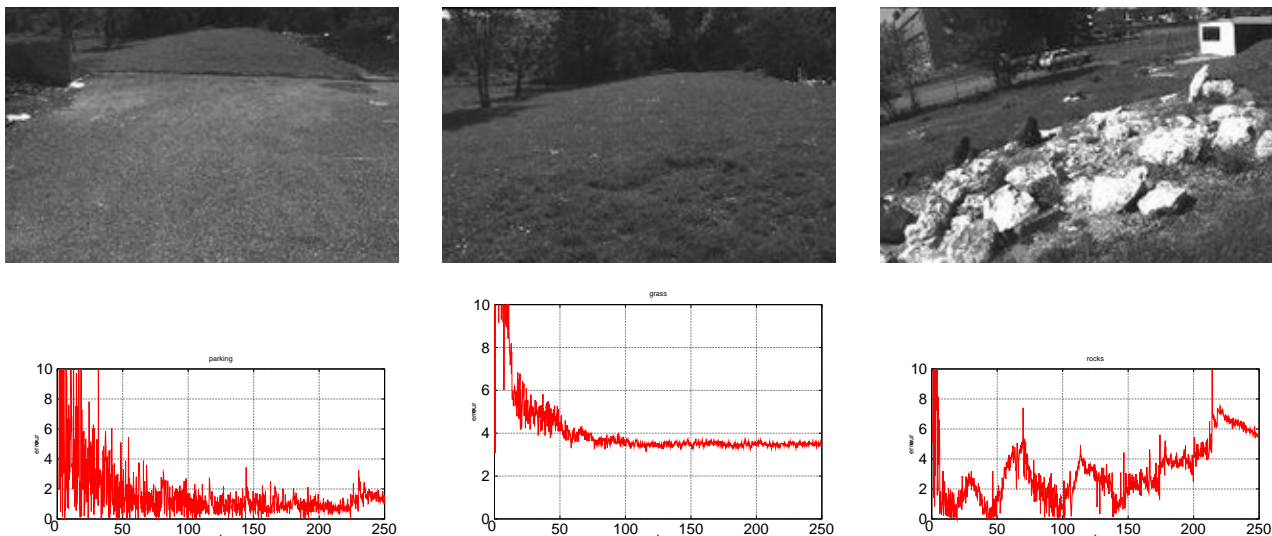


FIG. 2.11 – Comportement de l'odométrie sur trois types de terrain : les trois courbes montrent l'erreur relative sur l'estimation de la distance parcourue, obtenue grâce au GPS. L'odométrie est bien-sûr meilleure sur les terrains les plus réguliers, là où les glissements sont les plus faibles. Si le comportement des deux terrains à gauche (goudron ou herbe) semble modélisable par une erreur systématique, la difficulté consiste à détecter et qualifier le cas de droite (rochers), où les erreurs sur la position apparaissent erratiques et non-modélisables. On peut remarquer sur toutes ces courbes que l'erreur (calculée comme le rapport de l'écart en position sur la distance parcourue) est beaucoup importante lors des premières secondes du déplacement : ce phénomène est dû à la petitesse de la distance parcourue devant le bruit de mesure du GPS.

Pour qualifier l'odométrie, il est intéressant de distinguer *deux* sources d'erreurs : les erreurs de *précision*, qui sont essentiellement dues à des défauts de calibrage ou à la résolution des instruments, et les erreurs de *mesure*, qui se produisent lors de

déplacements *non mesurés*. Par exemple, le glissement d'une roue sur le sol se traduira par une erreur de *mesure* mais une erreur sur la détermination du diamètre d'une roue se traduira par une erreur de *précision*. [Borenstein 1995] utilise les termes d'erreurs *systématiques* ou *non-systématiques* pour désigner les erreurs de précision ou de mesure.

La majorité des contributions sur la qualification de l'odométrie portent sur la modélisation des erreurs de précision [Roumeliotis 1999, Fuke 1996, Volpe 1999] : comme le mentionnait Borenstein, celles-ci sont *systématiques* et donc *modélisables*. Les techniques de filtrage et d'estimation classiques peuvent alors s'appliquer.

Si les problèmes de glissements et de dérapages sont inexistantes ou négligeables, cette qualification peut alors être utilisée en ligne pour obtenir une incertitude sur la position, ou même en *améliorer* la précision ([Borenstein 1995] propose une méthode permettant d'éliminer *en ligne* les erreurs systématiques dues à des biais dans les modèles).

Les erreurs de mesure sont par contre non modélisables. À cause d'elles, l'erreur de la position odométrique n'est pas bornée et l'expression d'un modèle statistique *général*, valable pour un environnement quelconque, n'a pas de sens. Néanmoins, en fonction du terrain, on peut distinguer des comportements *remarquables* : la figure 2.11 montre le calcul de l'erreur sur la distance parcourue (par comparaison avec le GPS) sur trois terrains caractéristiques, lors de l'exécution d'une ligne droite à la vitesse constante de 5 cm.s^{-1} .

On peut remarquer sur ces courbes d'erreur que dans le cas d'un terrain plat et horizontal (« goudron » sur la figure) les erreurs *non-systématiques* sont négligeables et le comportement de l'odométrie peut se modéliser. En revanche, dans le cas d'un terrain accidenté (« rochers » sur la figure), les glissements, dérapages et vibrations du châssis perturbent énormément ce modèle et le comportement semble erratique.

La difficulté consiste donc à *détecter* les cas pathologiques pour *décider* si l'odométrie se comporte de façon normale ou aberrante, c'est-à-dire si les erreurs de mesures sont prépondérantes ou non. Malheureusement, *Lama* n'a pas été instrumenté dans ce but et l'obtention d'informations pertinentes semble difficile sans l'aide de méthodes ou capteurs supplémentaires. Néanmoins, plusieurs pistes peuvent être étudiées.

Une première source d'information réside dans la vitesse des six roues. L'odométrie présentée jusqu'ici ne fait usage que des vitesses des roues centrales et l'utilisation des vitesses des six roues pourrait permettre d'une part d'améliorer la précision de la position et, d'autre part, de détecter les glissements trop importants.

La figure 2.12 montre l'évolution de la vitesse de deux des six roues de *Lama* au cours des trois expériences de la figure 2.11. Si ces vitesses évoluent de manière similaire pour les deux premiers types de terrain, on remarque à la fois un bruit et des différences beaucoup plus importantes dans le dernier cas (on note cependant, en particulier pour le cas de « l'herbe », des bruits soudains et sans explication apparente).

Théoriquement, ces vitesses devraient être égales, puisque le robot était commandé en ligne droite. Le fait qu'elles ne le soient plus pour le cas des rochers provient du fait que l'asservissement réalisé est imparfait. Ce phénomène peut néanmoins être d'une utilité certaine pour détecter la présence des erreurs *non-systématiques* : on pourrait supposer que celles-ci *surviendraient* lors de l'apparition d'un bruit élevé sur les vitesses des roues. Il serait alors possible de déclencher un autre méthode de localisation, ou peut-être d'améliorer l'estimation de la position en utilisant, à la place de la seule vitesse des roues centrales, une vitesse moyenne comme sur la figure 2.13.

Une autre source d'information actuellement disponible à bord de *Lama* est la mesure des courants consommés dans les moteurs des roues. Intuitivement, il semble raisonnable de supposer que ceux-ci pourraient être reliés, d'une façon ou d'une autre, à

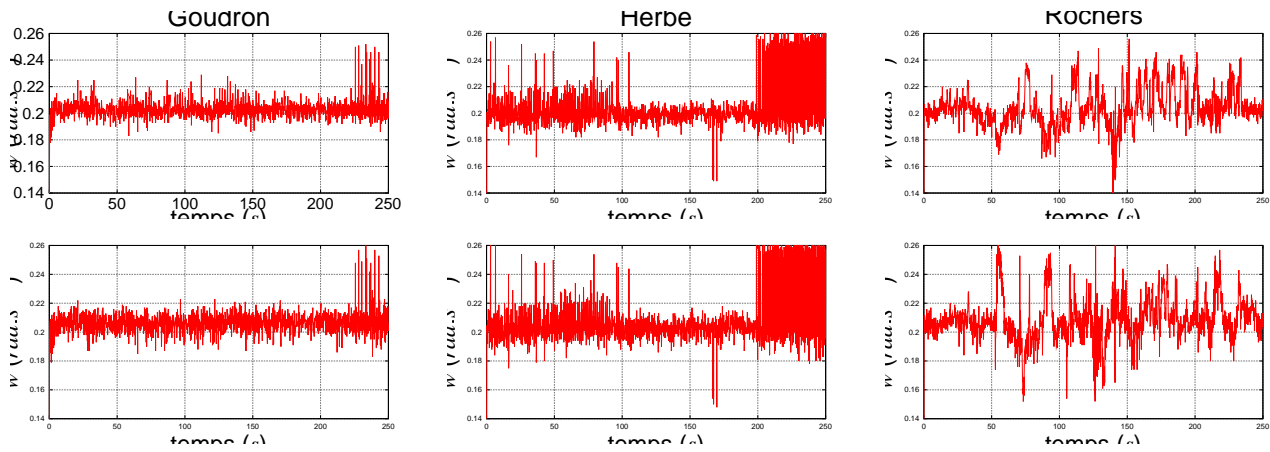


FIG. 2.12 – Vitesses des roues en fonction du terrain. Sur les sols réguliers comme le goudron ou l'herbe, les vitesses mesurées sur les roues sont semblables (seules les vitesses de roues avant et arrière gauche sont représentées ici). En revanche, lorsque le terrain devient plus accidenté (graphique de droite), le bruit augmente et on observe des différences notables entre les vitesses théoriquement égales.

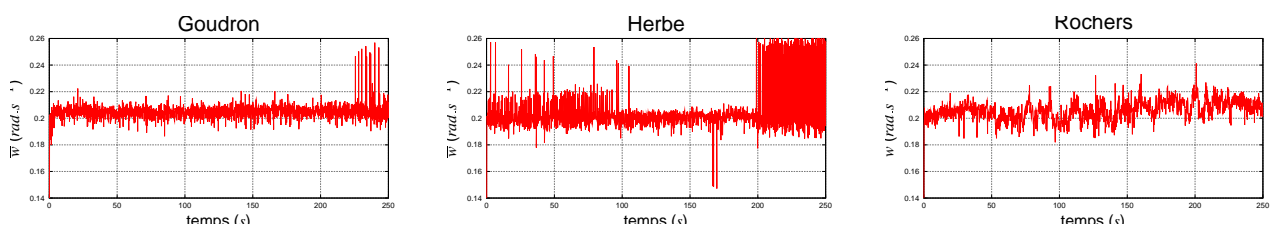


FIG. 2.13 – Moyennage des vitesses des six roues. L'utilisation d'une vitesse moyennée sur les six roues du robot peut certainement améliorer l'estimation de position : sur le graphique de droite, on constate une nette diminution du bruit, en comparaison avec les courbes de la figure 2.12.

la nature du terrain. En effet, en première approximation, on sait que la tension appliquée sur les moteurs est proportionnelle à leur vitesse de rotation et que le courant consommé est proportionnel aux efforts développés. Plus le terrain est difficile, plus la consommation de courant *devrait* être importante.

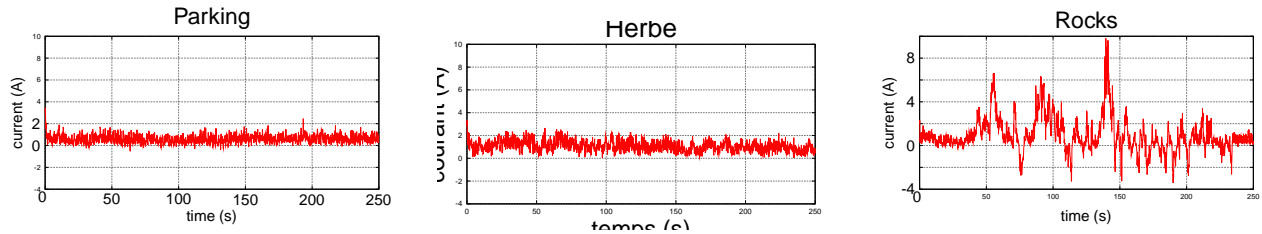


FIG. 2.14 – Courants consommés dans les moteurs des roues : l'influence du terrain et de sa géométrie est assez nette.

La figure 2.14 montre, toujours au cours des trois mêmes expériences, l'évolution du courant consommé par une roue particulière (les autres ayant des comportements semblables). Les conclusions que l'on peut tirer sont à peu près les mêmes que pour les vitesses de rotation : plus le terrain est difficile, plus le bruit mesuré devient important. Par exemple, on remarque bien l'influence de la pente du terrain « herbe », qui provoque une consommation de courant plus élevée que pour le terrain plat « goudron ». Par ailleurs, on constate que sur le terrain « rochers », parcouru en pente descendante, le robot se trouve parfois à produire du courant.

2.7 Conclusions

L'observation du comportement de l'odométrie permet d'ores et déjà d'affirmer qu'il est possible de proposer un modèle empirique d'erreur, pour l'odométrie sur un terrain favorable et connu (par exemple, pour une zone relativement plane). Ce modèle, bien qu'empirique, pourra néanmoins permettre l'utilisation combinée de l'odométrie avec d'autres estimateurs de position. Mieux encore, une étude statistique effectuée sur un vaste ensemble de types de terrains — toujours favorables — permettra l'identification de ce modèle en ligne, grâce à l'observation des diverses grandeurs présentées dans cette section. De plus, et malgré le manque d'instrumentation à bord, il sera certainement possible de détecter les zones où les glissements deviennent prépondérants — et automatiquement déclarer l'odométrie comme inadaptée — grâce à, par exemple, l'emploi de techniques telles que celles que l'on trouve dans le domaine de la surveillance de processus (détection de rupture de modèle).

Ce travail permettra donc à la fois d'améliorer encore le calcul de la position, notamment en fournissant une imprécision sur les paramètres estimés, et également de stopper le calcul de la position lorsqu'il n'est plus pertinent et déclencher d'autres méthodes de localisation. Ainsi, le chapitre suivant présente une méthode de localisation *exteroceptive* qui, utilisée conjointement avec l'odométrie, permet de détecter de façon peut-être plus sûre les cas d'échec de cette dernière.

III

L'odométrie optique

L'odométrie ne peut remplir à elle seule le rôle des estimateurs de mouvement. Le chapitre précédent montre que même si celle-ci peut être précise dans certaines conditions, elle reste — à l'heure actuelle — difficilement capable de produire un diagnostic fiable sur la position qu'elle calcule. La source principale de ce problème est que l'odométrie n'utilise que des capteurs mesurant des grandeurs *internes* au robot : on parle de méthode *proprioceptive*. Elle n'est donc pas en *interaction* directe avec l'environnement. De plus, seule une partie du mouvement du robot est mesurée (aucun capteur ne détecte, par exemple, les glissements latéraux) et le calcul de la position est dépendant d'hypothèses fortes de roulement sans glissement.

L'utilisation d'autres méthodes d'estimation de mouvement permet, d'une part, d'améliorer l'estimation du déplacement local du robot mais aussi, et surtout, de détecter les cas d'échec de l'odométrie.

Parmi les nombreuses autres méthodes étudiées dans la littérature, le premier chapitre a déjà souligné l'abondance des différentes techniques basées sur l'utilisation d'images de profondeurs ou de vision monoculaire. Cependant, la plupart de ces contributions requièrent soit une segmentation préalable des scènes perçues (images de profondeur), soit la détermination conjointe du mouvement et de la géométrie de l'environnement (vision monoculaire). Ces besoins soulèvent tous deux des problèmes encore largement ouverts ; les réponses apportées ne sont encore que partielles et limitées à certains types de scènes particulières.

Ce chapitre propose une méthode d'estimation de mouvement, basée sur l'utilisation *directe* d'images de profondeur *non-structurées*. Ainsi, la complexité du problème est doublement réduite puisque, d'une part, la géométrie de l'environnement est connue à chaque instant et que, d'autre part, le problème de la segmentation est évité.

La technique proposée estime des déplacements élémentaires et fournit, par cumul, une estimation de la position du robot. En cela elle est équivalente à l'odométrie, ce qui justifie le qualificatif d'*odométrie optique*. Selon la typologie exhibée dans la section 1.4, il

s'agit bien d'une méthode d'estimation de mouvement : croissance des erreurs non bornée, fréquence d'activation très rapide, utilisation des données brutes et aucun contrôle nécessaire.

Une différence majeure, cependant, avec l'odométrie est que résultat de ces mesures provient d'une *observation* de l'environnement : il s'agit d'une méthode *exteroceptive*. Ce « détail fondamental » permet d'obtenir une estimation *généralement* beaucoup plus précise des mouvements, *indépendante des terrains traversés*.

La section suivante expose le principe général de la méthode, telle qu'elle fonctionne actuellement à bord du robot. Des expériences montrant les résultats typiquement obtenus avec deux robots différents sont ensuite présentées dans la section 3.2. Enfin, une analyse de la précision et de l'influence de certains paramètres est effectuée dans la section 3.3 et quelques améliorations possibles sont proposées.

3.1 Principe

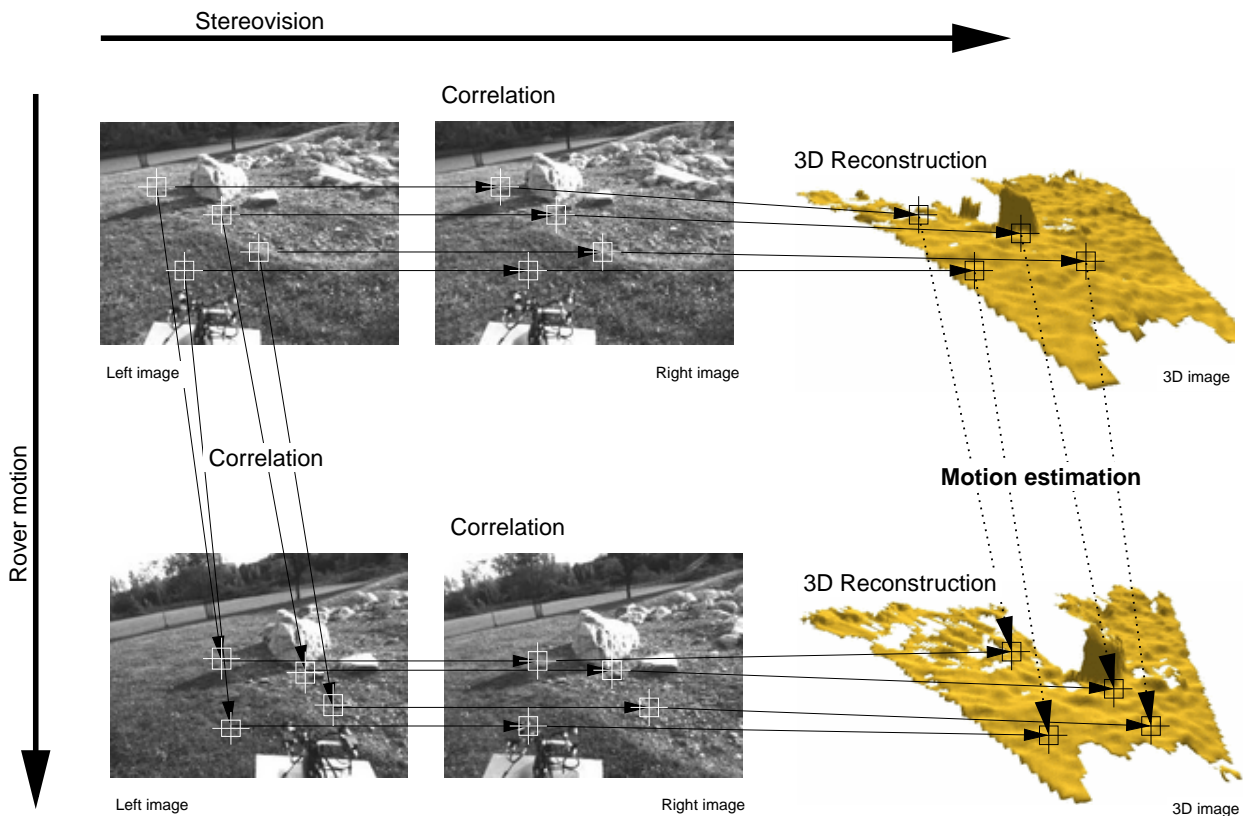


FIG. 3.1 – Principe général de l'odométrie optique. Cette figure schématise les trois phases du processus de calcul d'un déplacement dans une séquence d'images. i) une paire stéréoscopique d'images est acquise et permet de calculer une image 3-D de l'environnement (corrélation, triangulation et reconstruction). Dans le même temps, un ensemble de pixels est sélectionné dans une des images (seuls quatre pixels sont dessinés ici). ii) Les pixels sélectionnés sont suivis dans une séquence d'images monoculaires par corrélation. iii) Enfin, lorsqu'une nouvelle image 3-D est disponible, deux ensembles de points 3-D sont appariés grâce à la phase de suivi et les six paramètres du déplacement sont calculés (estimation du déplacement).

L'odométrie optique présentée dans ce chapitre permet de mesurer les *six* paramètres des déplacements 3-D effectués par le robot, à l'aide d'une séquence d'images vidéo et 3-D acquises périodiquement à relativement grande fréquence. Dans ces séquences sont

réalisés des appariements de points mesurant les déplacements. Ce principe général, résumé dans la figure 3.1, a initialement été proposé dans [Carre 1998] et considérablement amélioré dans [Lacroix 1999]. Une technique similaire a également été proposée, de façon indépendante, dans [Stentz 1999].

Les sections suivantes détaillent les principales phases de la méthode.

3.1.1 Reconstruction tridimensionnelle

La reconstruction de scènes tridimensionnelles par stéréo-vision permet, à partir de paires d'images stéréoscopiques, de déterminer de façon très simple la structure de l'environnement. Le principe de cette méthode est largement répandu et son implémentation sur le robot *Lama* a été publiée dans [Haddad 1998a].

Pour l'odométrie optique réalisée, une image de disparité dense — équivalente à une image de profondeur — est produite à partir des images vidéo grâce à un appariement de pixels par corrélation (le score de corrélation utilisé est le ZNCC¹[Martin 1995]). La figure 3.2 résume ce principe et la figure 3.3 montre un exemple typique d'image 3-D obtenue.

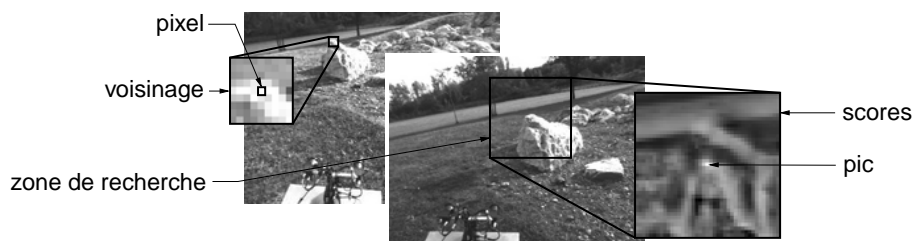


FIG. 3.2 – Principe de la corrélation de pixel. Dans une première image (à gauche), un voisinage autour de chaque pixel devant être apparié est extrait et représente le modèle des ces pixels. Dans une seconde image (au centre), une zone dans laquelle doit être effectuée la corrélation est définie, pour chacun des pixels à corrélérer. Le calcul du score de corrélation est effectué entre les modèles et le voisinage de chacun des pixels de la zone de recherche (à droite). Le score le plus élevé correspond à la position du pixel recherché.

Les faux appariements sont rejetés grâce à une corrélation inverse et à divers seuils sur la forme de la courbe de corrélation (principalement, la valeur du pic et la différence avec le second pic). Cependant, on remarque que des erreurs *systematiques* apparaissent sur la frontière entre deux zones de luminosité et de profondeurs très différentes : les obstacles sont artificiellement grossis de la moitié de la taille de la fenêtre de corrélation [Sara 1997] (voir annexe C). Ces erreurs ne peuvent pas être filtrées grâce aux différents seuils ni à un algorithme de filtrage par régions. De plus, la précision de ces points erronés tend, paradoxalement, à être très bonne : il est quasiment impossible de les éliminer en considérant le seul modèle d'erreur sur la stéréo-vision. Le paragraphe 3.1.4 présente une technique permettant malgré tout d'éliminer ces points.

3.1.2 Suivi de pixels dans une séquence d'images

Cette étape permet de réaliser les appariements de pixels entre deux images monoculaires consécutives, acquises au cours du déplacement du robot : il s'agit donc de la phase qui conditionne le plus la précision de l'estimation du déplacement.

Dans les environnements texturés, les techniques à base de corrélation de pixels sont très efficaces pour réaliser des appariements et calculer des ressemblances entre

¹Zero Mean Normalized Cross Correlation.

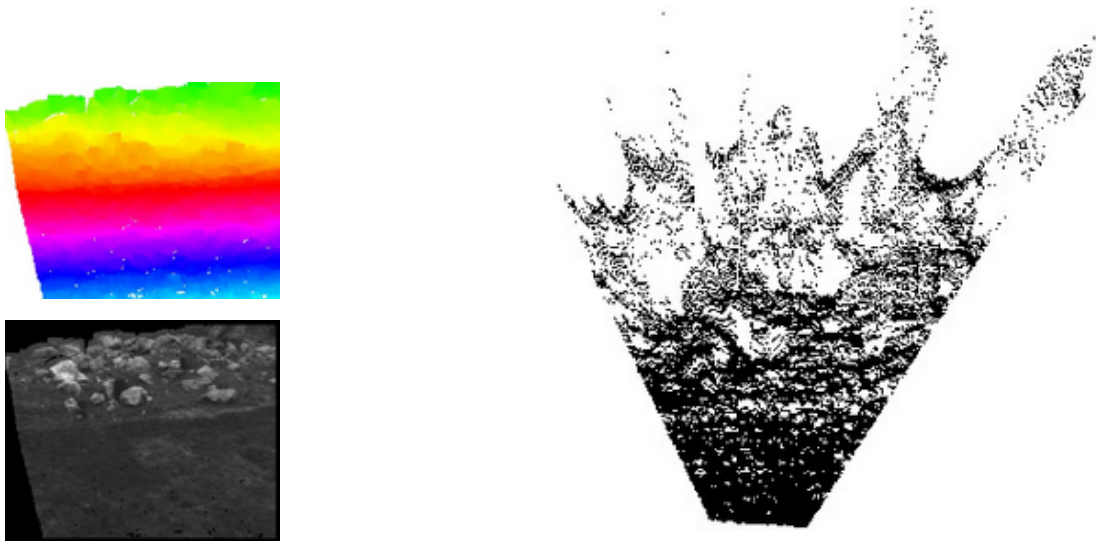


FIG. 3.3 – Image non structurée de points 3-D acquise sur le robot Lama : l'image en bas à gauche montre une image vidéo ayant servi à la reconstruction de l'image 3-D vue de dessus, à droite. L'image de disparité (équivalente à une image de profondeur) correspondante est montrée en haut à gauche.

des images ou des zones d'images. Ainsi, la stéréo-vision fonctionne remarquablement bien et la méthode choisie pour effectuer les appariements est, sur le principe, identique : étant donné un pixel et son voisinage (de taille paramétrable), une corrélation entre ce voisinage et des régions de même taille dans la nouvelle image permet, la plupart du temps, de retrouver la position des pixels recherchés.

Différents scores de ressemblance peuvent être utilisés : [Martin 1995] propose une comparaison de ceux-ci en fonction de paramètres tels que la quantité de lumière ou le bruit présent dans les images. Comme pour la stéréo-corrélation, il apparaît que les deux scores les plus robustes sont le ZNCC et un score basé sur une transformation « census » des images [Zabih 1994].

Contrairement au travaux de [Zhang 1991], chaque pixel est donc considéré individuellement : aucune optimisation n'est utilisée pour prendre en compte des hypothèses de rigidité de l'environnement qui permettraient de focaliser les recherches une fois que quelques point auraient été appariés. Malgré le côté séduisant de ces techniques plus sophistiquées, les résultats actuellement obtenus ne nous ont pas incités à les explorer (voir figure 3.4).

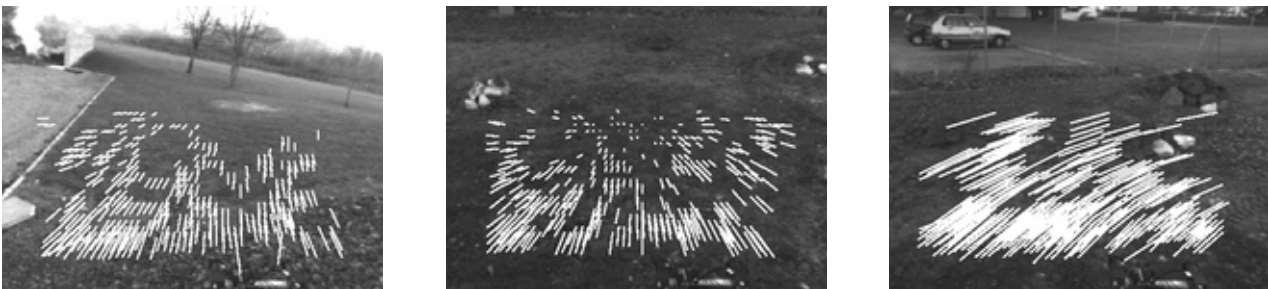


FIG. 3.4 – Exemple de suivi de pixels pour trois types de mouvement : ces scènes comportent environ 300 points suivis entre des paires d'images distantes d'environ 10 cm.

3.1.3 Prédiction de la position des pixels

La phase de suivi nécessite une prédiction sur la nouvelle position de chaque pixel, afin de limiter la zone dans laquelle la corrélation doit être effectuée et diminuer autant que possible les temps de calcul nécessaires.

Pour cela, on utilise à la fois la dernière image $3-D$ produite et l'odométrie. La nouvelle position $3-D$ de chaque point peut ainsi être estimée et, connaissant les paramètres intrinsèques des caméras, les zones de recherche peuvent être centrées autour des positions $2-D$ prévues. La taille de ces zones est calculée en fonction de l'incertitude que l'on autorise sur l'estimation mais, à l'heure actuelle, elle est constante et indépendante du point considéré. Un raisonnement géométrique permet pourtant de constater que plus un point est éloigné de la caméra, moins son mouvement au cours d'une translation sera important ; la distance n'influence par contre pas les déplacements des points au cours de *rotations* du robot.

Une amélioration substantielle serait donc d'utiliser la prédiction du mouvement par l'odométrie pour calculer la taille des zones de recherche.

3.1.4 Sélection des pixels

La phase de suivi est, du point de vue des temps de calculs, la partie la plus coûteuse de la méthode : une attention particulière a été apportée à la sélection *a priori* des pixels à suivre.

À cause du bruit dans les images vidéo et du sous-échantillonnage effectué par les caméras, le suivi de pixels dérive toujours. Pour éviter les faux appariements, il est nécessaire de s'assurer *a priori* que les pixels choisis vont pouvoir être suivis et que les points $3-D$ correspondants sont connus avec une bonne précision. Le choix des pixels les plus intéressants vis-à-vis de ces critères se fait en trois étapes.

Une première sélection *a priori* est faite sur la base des images vidéo et de la précision estimée des points $3-D$. Pour éviter la sélection de ces pixels difficilement suivables, un filtrage est effectué à partir d'une image représentant la similitude d'un pixel avec ses voisins. Le score de corrélation du pixel avec chacun de ses voisins est calculé en utilisant le même score et la même taille de fenêtre que l'algorithme de suivi proprement dit (voir figure 3.5). Cette auto-corrélation aboutit à une surface dont la forme indique la ressemblance d'un pixel avec ses voisins et les points étant les plus différents de leurs voisins sont sélectionnés en priorité. Cette mesure donne une indication sur la *précision* de l'étape de suivi, mais pas sur la *certitude* : calculer une mesure d'ambiguïté nécessiterait le calcul de scores de corrélation pour un vaste voisinage, qui est très coûteux en temps de calcul.

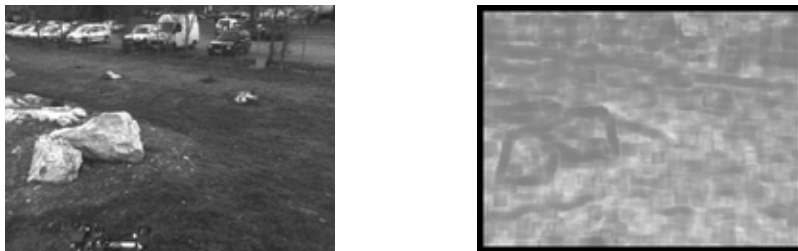


FIG. 3.5 – Mesure de ressemblance locale calculée sur une image complète. L'image de droite, calculée à partir de l'image de gauche, montre la mesure de ressemblance locale en niveaux de gris : les pixels les plus clairs sont les plus discriminants localement, et donc les meilleurs candidats pour le suivi. On voit comment les contours des objets, zones de forts contrastes selon une seule direction, sont éliminés.

Ensuite, à l'issue de la phase de suivi, un modèle empirique de l'algorithme de corrélation (voir l'annexe ??) est utilisé pour éliminer les pixels ayant de fortes chances d'avoir dérivé. Il s'agit donc d'une sélection *a posteriori*, qui ne permet pas d'éviter le suivi de pixels inutiles. Plus la phase de sélection *a priori* sera efficace, moins ce filtrage éliminera de points, économisant ainsi un temps de calcul précieux.

Enfin, un dernier filtrage *a posteriori* est effectué lors de la phase de calcul du déplacement, sur la base de critères géométriques.

3.1.5 Estimation du déplacement entre deux ensembles de points 3-D

Lorsque deux paires stéréo sont disponibles, deux ensembles de points 3-D $\{p_i\}$ et $\{p'_i\}$ sont alors appariés deux à deux : chacun des points p_i et p'_i sont reliés par le même déplacement de sorte que

$$p'_i = R.p_i + T + N_i \quad (3.1)$$

où R est une matrice de rotation, T un vecteur correspondant à une translation et N_i un bruit, provenant des erreurs dans les coordonnées 3-D des points.

L'estimation du déplacement consiste donc à trouver les meilleures rotation \hat{R} et translation \hat{T} s'approchant le plus du déplacement réel. [Huang 1994] présente une revue des différentes méthodes ayant été proposées pour résoudre ce problème. Dans le cas de correspondances de points 3-D à points 3-D, [Arun 1987] a proposé une méthode, très performante, basée sur une décomposition en valeurs singulières.

L'algorithme utilisé estime \hat{R} et \hat{T} au sens des moindres carrés : il consiste donc à minimiser

$$\Sigma^2 = \sum_{i=1}^N \|p'_i - (R.p_i + T)\|^2 \quad (3.2)$$

Si le bruit N_i est à moyenne nulle, on peut remarquer que $\{p'_i\}$ et $\{p''_i = \hat{R}.p_i + \hat{T}\}$ ont le même barycentre. C'est à dire que

$$p' = p''$$

avec

$$p = \frac{1}{N} \cdot \sum_{i=1}^N p_i \quad ; \quad p' = \frac{1}{N} \cdot \sum_{i=1}^N p'_i \quad ; \quad p'' = \frac{1}{N} \cdot \sum_{i=1}^N \hat{R}.p_i + \hat{T}$$

Si l'on note

$$q_i = p_i - p \quad ; \quad q'_i = p'_i - p'$$

alors

$$\Sigma^2 = \sum_{i=1}^N \|q'_i - R.q_i\|^2 \quad (3.6)$$

Le problème se résume donc à trouver \hat{R} minimisant Σ^2 dans l'équation (3.6). La translation \hat{T} est ensuite obtenue de façon exacte grâce à (3.7) :

$$\hat{T} = p' - \hat{R}.p \quad (3.7)$$

[Arun 1987] propose une technique non-itérative pour estimer \hat{R} . Elle repose sur une décomposition en valeurs singulières de la matrice 3×3

$$H = \sum_{i=1}^N q_i \cdot {}^t q'_i$$

où l'opérateur ${}^t X$ représente la transposition de la matrice X .

En décomposant H en valeurs singulières, on obtient trois matrices U , W et V telles que

$$H = U.W.{}^t V$$

Dans le cas où les points $\{p_i\}$ et $\{p'_i\}$ ne présentent pas de singularités, c'est à dire qu'ils ne sont ni coplanaires, ni colinéaires et en nombre suffisant vis à vis du bruit, la matrice de rotation \hat{R} est donnée par

$$\hat{R} = V.{}^t U \quad (3.10)$$

La démonstration de cet algorithme ainsi que l'étude des cas singuliers mentionnés ci-dessus se trouve dans [Arun 1987]. Il est à noter qu'expérimentalement, nous ne nous sommes jamais trouvés confrontés aux cas dégénérés évoqués étant donné le grand nombre de points utilisés.

3.2 Premiers résultats

La méthode, telle que décrite dans la section précédente, fonctionne à bord du robot *Lama* depuis 1999. Elle est utilisée dans la plupart des démonstrations de navigation car elle s'est avérée généralement plus précise que l'odométrie, même sur les terrains où cette dernière fonctionne bien. Cette section présente, à l'aide d'un exemple, les résultats typiquement obtenus aujourd'hui.

D'autre part, la naissance d'un projet de contrôle automatique d'un ballon dirigeable a permis d'utiliser une méthode d'odométrie optique dérivée de celle présentée dans ce chapitre. Des premiers résultats sont présentés à la fin de cette section.

3.2.1 Estimation de position sur *Lama*

La figure 3.6 permet d'observer le comportement de l'odométrie optique lors de l'exécution d'une trajectoire réelle. Loin d'être particulière, cette expérience montre le comportement typiquement observé depuis que cette méthode d'estimation de position a été implémentée sur le robot. Des résultats similaires ont été publiés pour la première fois dans [Chatila 1999]. [Mallet 2000] présente des résultats supplémentaires et intègre l'expérience acquise au cours d'un an d'utilisation du logiciel développé.

La trajectoire réalisée représente un déplacement total d'environ 70 m effectué à la vitesse de 2 cm.s⁻¹. Le sol était majoritairement plat et composé d'herbe, pour permettre l'utilisation de l'odométrie classique. Pour des raisons techniques, l'utilisation du gyromètre n'était pas possible, et la position odométrique a donc été estimée avec l'aide du compas magnétique. La position réelle du robot a été mesurée par le GPS décrit dans l'annexe A.2.5, à la fréquence de 2 mesures par seconde.

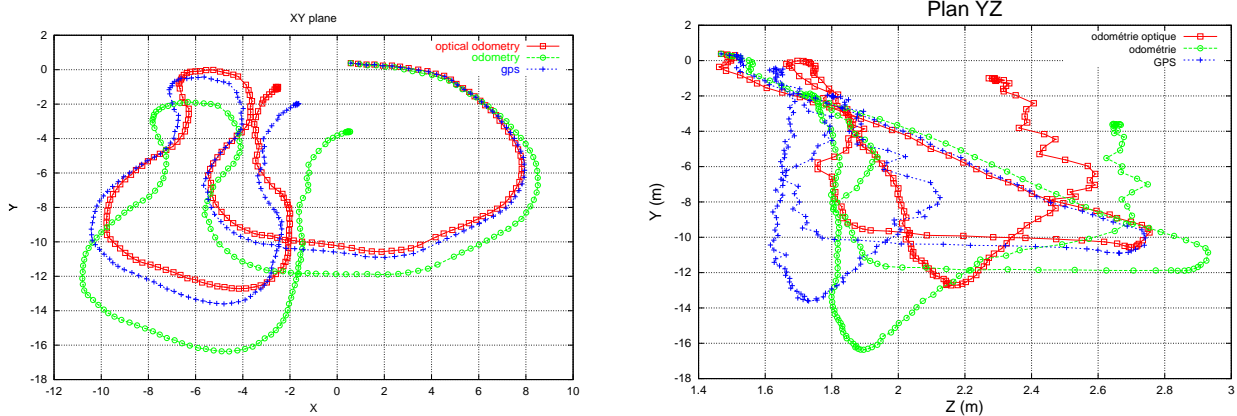


FIG. 3.6 – Trajectoire réelle mesurée par odométrie optique, odométrie et GPS. Les deux graphiques montrent une vue de dessus (plan XY horizontal, à gauche) et de profil (plan YZ vertical, à droite) d'une trajectoire réalisée à la vitesse de 2 cm.s^{-1} sur une distance d'environ 70 m. Le terrain sur lequel a évolué le robot était majoritairement composé d'herbe, ce qui permet une comparaison objective entre odométrie et odométrie optique.

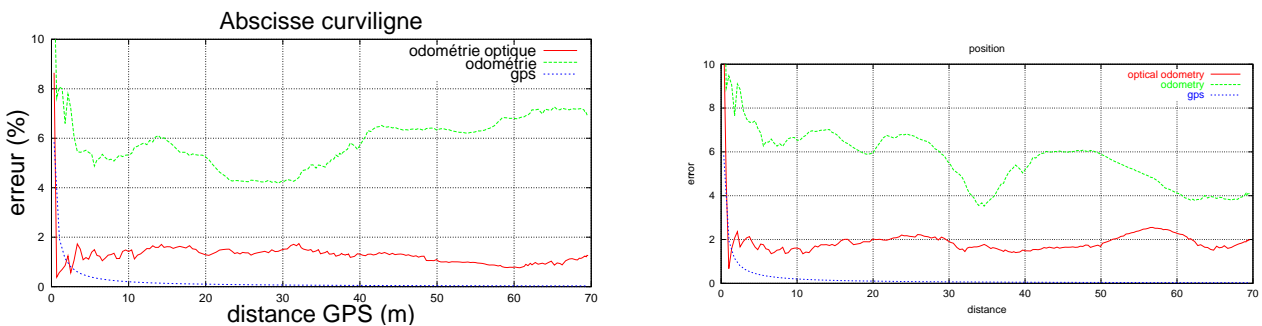


FIG. 3.7 – Comparaison des erreurs en translation entre odométrie optique et odométrie. Sur le graphique de gauche est représentée l'erreur sur la mesure de la distance parcourue pour la trajectoire de la figure 3.6. Sur le graphique de droite on peut observer la distance relative entre la position mesurée par le GPS et les positions calculées par l'odométrie optique et l'odométrie, pour cette même trajectoire. La courbe de précision du GPS est superposée, montrant que le bruit sur la position est rapidement négligeable.

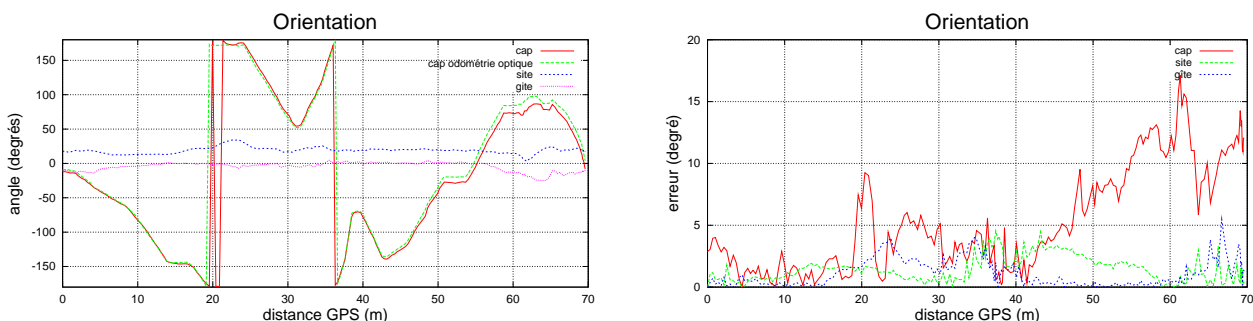


FIG. 3.8 – Erreurs de l'odométrie optique sur l'estimation des angles. Les angles de cap, site et gîte sont mesurés de façon absolue sur le robot et le graphique montre l'erreur entre ces valeurs et le résultat de l'estimation par l'odométrie optique. L'erreur importante sur l'estimation du cap s'explique certainement, en regard des courbes 3.7 sur l'erreur en position, par des problèmes de calibrage du compas.

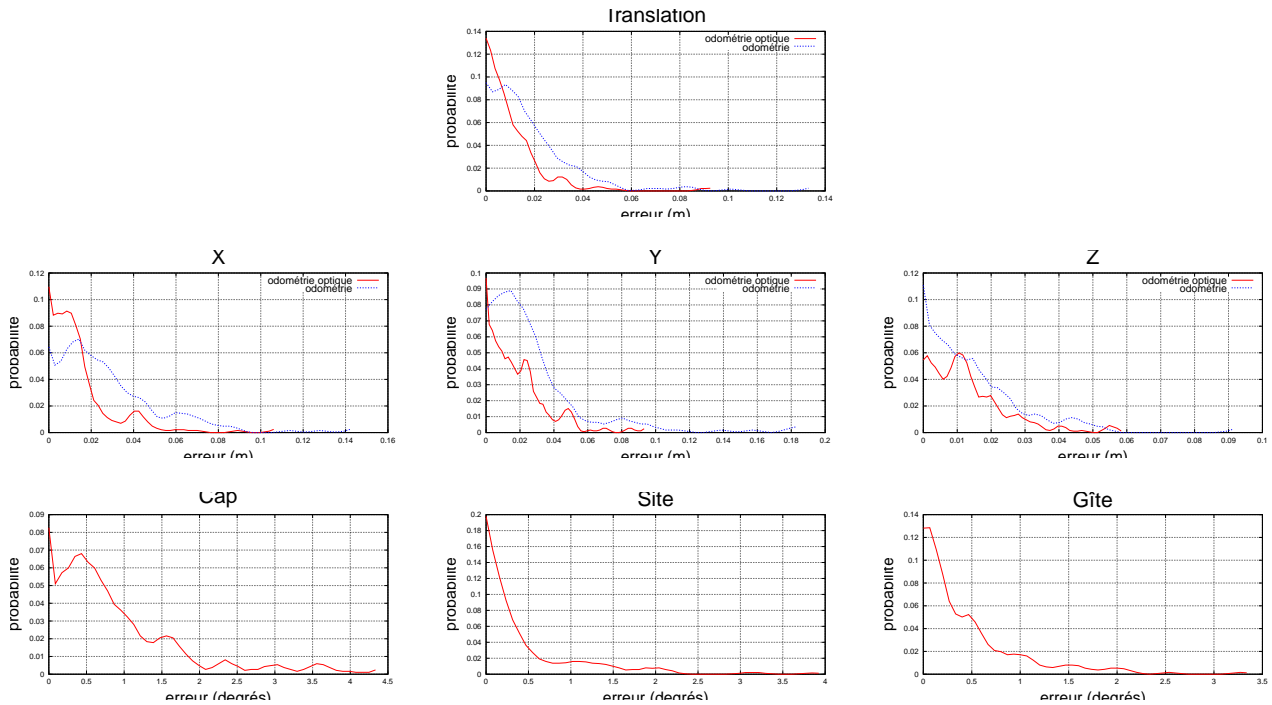


FIG. 3.9 – Histogramme des erreurs en translation pour l’odométrie et l’odométrie optique. Pour chacune des 200 images acquises au cours du déplacement, les erreurs sur l’estimation des translations locales ont été calculées. Les histogrammes normalisés sur chacun des axes d’un repère global montrent que l’odométrie optique commet plus fréquemment que l’odométrie des erreurs faibles et se rapproche donc plus de l’histogramme « idéal » qui serait un pic de Dirac. Il est à noter que la précision du GPS est de ± 1 cm sur X et y et de ± 2 cm sur Z : en dessous de ce seuil on peut considérer que les histogrammes ne sont pas significatifs.

À titre indicatif, la taille des images vidéo utilisées pour la stéréo-corrélation et l’estimation de la position était de 256×192 pixels et l’odométrie optique a été utilisée avec un maximum de 500 appariements, sur un total de 200 images acquises régulièrement. Le temps de calcul nécessaire pour effectuer une estimation était de l’ordre de 5 secondes, sur le PowerPC G3 équipant *Lama*, approximativement réparties en 20% pour la stéréo-vision et 80% pour le suivi des pixels.

Les positions calculées par l’odométrie et l’odométrie optique sont comparées sur la figure 3.7. La première constatation est que l’estimation de la distance parcourue est bien meilleure avec l’odométrie optique (erreur d’environ 1 à 2 %) que pour l’odométrie (erreur d’environ 6 à 8 %). L’erreur sur la position absolue, résultant de l’intégration des déplacements élémentaires, est elle aussi plus faible pour l’odométrie optique (environ 2 % contre 4 à 6 % pour l’odométrie).

De façon surprenante, l’odométrie s’avère meilleure sur l’estimation de la position absolue que sur la distance parcourue : ce phénomène est lié à la forme de la trajectoire réalisée et n’est donc pas significatif (une trajectoire en ligne droite aurait par exemple donné la même erreur sur la position et la distance parcourue). Par ailleurs, et contrairement à l’odométrie optique, les angles de configuration du robot sont mesurés de façon absolue et ne contribuent donc pas à la dérive de l’odométrie.

L’observation des erreurs locales commises pour chaque estimation de déplacement élémentaire permet de confirmer qu’en moyenne l’odométrie optique se comporte mieux que l’odométrie. La figure 3.9 montre des histogrammes réalisés à partir des erreurs commises sur les translations, au cours de l’exécution de la trajectoire. Pour l’estimation des translations, l’odométrie optique commet plus souvent des erreurs faibles et les

histogrammes se rapprochent plus du cas « idéal » ou une erreur de 0 a une probabilité de 1.

La comparaison entre odométrie et odométrie optique se limite cependant à l'étude des translations : l'odométrie utilise les angles de configuration mesurés de façon absolue. Il est néanmoins possible de comparer l'estimation des angles calculés par l'odométrie optique et ceux mesurés par les instruments de bord.

On constate sur la figure 3.8 que si le site et le gîte sont correctement estimés à quelques degrés près, le cap est quant à lui très différent, à partir de la moitié de la trajectoire, de celui donné par le compas magnétique. L'observation de l'évolution du cap ne montre pas de phénomène particulier pouvant expliquer cette erreur mais deux causes sont néanmoins possibles. Soit l'odométrie optique a commis une erreur localement qui a introduit une dérive dans l'angle de cap estimé, soit le compas a été victime d'une perturbation à partir de cet endroit. L'erreur en position absolue ne montre pas d'augmentation brutale ou de dérive particulière et, au contraire, l'observation de la dernière partie de la trajectoire mesurée a tendance à montrer que l'odométrie n'est plus parallèle au GPS. Une perturbation semble donc être la cause la plus probable.

Ces résultats sont donc d'ores et déjà très prometteurs. Néanmoins, la section 3.3 présente différents moyens dont l'analyse nous semble pertinente pour encore améliorer les positions calculées.

3.2.2 Estimation de position pour un ballon dirigeable

La méthode a été également utilisée pour l'estimation de la position d'un ballon dirigeable [Lacroix 2001b], présent au LAAS depuis peu. Les résultats ne sont que très préliminaires mais tout aussi encourageant que pour *Lama*.

Au cours de l'expérience présentée ici, le ballon a effectué une trajectoire d'une centaine de mètres, en revenant à son point de départ, et une centaine d'images stéréoscopiques ont été acquises grâce à une paire de caméras montées à bord. Le prototype ne disposant pour l'instant pas d'informatique embarquée, les images 3-D correspondantes ont été reconstruites au sol, hors ligne.

Dans le but de construire un modèle 3-D de la zone ainsi explorée, une estimation de position était nécessaire. Cependant, une différence fondamentale avec le robot *Lama* est que ce ballon ne dispose pas d'odométrie, ni d'aucune autre méthode fournissant une estimation du mouvement. La prédiction permettant de calculer la position des zones de recherche pour le suivi de pixel n'a donc pas pu être utilisée. Même si, au sol, le temps de calcul n'est pas une contrainte, la recherche exhaustive de pixels, dans de grandes zones, n'était pas satisfaisante du point de vue d'une future implémentation de cette méthode à bord. Un autre algorithme de mise en correspondance de pixels [Jung 2001] a donc été employé.

La figure 3.10 montre le résultat de la fusion des images 3-D en un modèle global, sous la forme d'une carte d'élévation (voir chapitre 4). Les informations de luminance ont permis de texturer le modèle obtenu et, visuellement, les résultats sont très satisfaisants : la précision obtenue dans le plan horizontal est inférieure à la taille d'une cellule du modèle, soit 5 cm.

De futurs travaux devraient permettre de qualifier plus précisément les résultats que l'on peut espérer sur ce type d'engin. De plus, il n'est pas exclu d'utiliser cette nouvelle technique de mise en correspondance, qui semble particulièrement efficace. Elle est cependant, à l'heure actuelle, encore relativement coûteuse en temps de calcul.

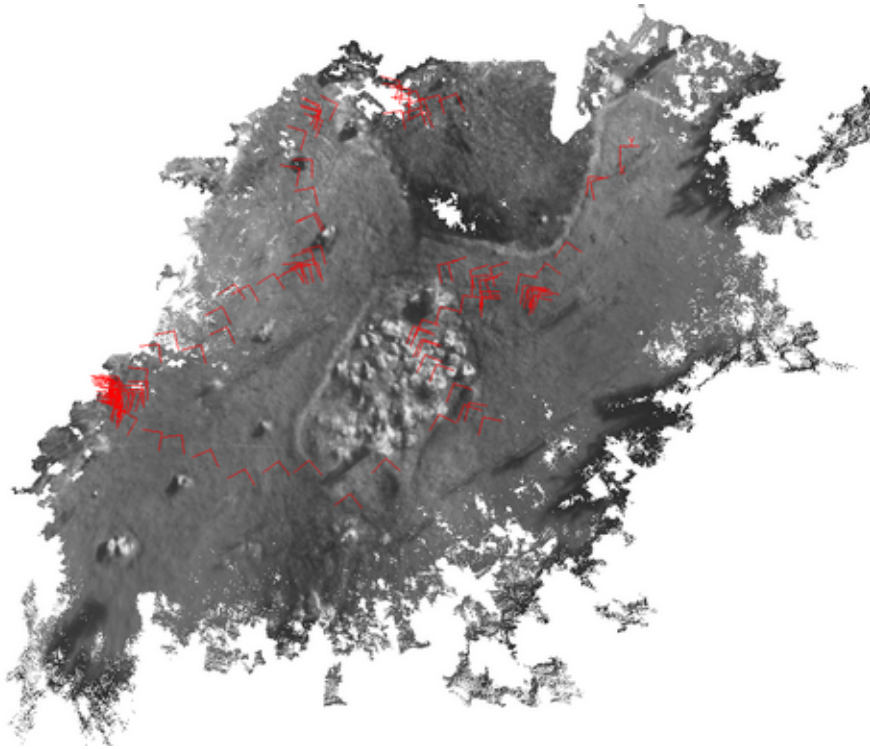


FIG. 3.10 – Localisation pour la construction d'un modèle 3-D aérien. Le modèle construit est une carte d'élévation, représentant l'altitude du terrain sur un plan horizontal. Le modèle, ici vu de dessus, est constitué de la fusion d'une centaine d'images 3-D et mesure environ 50×50 mètres. La précision de la fusion est remarquable (de l'ordre de 5 cm) puisque on ne distingue visuellement pas d'erreurs.

3.3 Analyse et qualification de la méthode

La section précédente permet d'affirmer que l'odométrie optique peut être performante et qu'il s'agit d'une méthode utilisable à bord d'un robot. Son domaine d'application est *a priori* beaucoup plus vaste que celui de l'odométrie (comme le montre l'exemple du ballon dirigeable) et les objectifs actuels des développements autour de cette méthode convergent vers une diffusion plus large que le seul cadre du robot *Lama* [Lacroix 2001d].

Telle qu'elle fonctionne actuellement l'odométrie optique demande un niveau d'expertise important pour donner de bons résultats : quantité de paramètres ont été fixés par analyse empirique du comportement et, en fonction des scènes perçues, de la vitesse du robot ou de la taille des images, ces paramètres doivent être modifiés, de façon parfois importante, et adaptés à l'environnement courant.

D'autre part, un grand défaut empêchant toute utilisation au sein d'une application réelle est l'absence de *qualification* précise. Les positions sont calculées au mieux mais aucun diagnostic précis ne permet encore de décider si l'algorithme a bien plutôt bien fonctionné ou non, ou si la position est plutôt précise ou plutôt imprécise.

Afin de permettre l'utilisation de cette odométrie par des personnes désirant développer leurs applications *autour* de cette méthode, il serait donc nécessaire de fournir une estimation de sa *précision* en ligne, et cette section présente l'état actuel des recherches allant dans ce sens. Ce travail est complexe, et le plus grand problème consiste à établir des procédures rigoureuses permettant d'isoler l'influence de chaque paramètre. Les pistes présentées ici ne sont donc que *préliminaires* et ne constituent pas

une validation rigoureuse. Néanmoins, des résultats d'ores et déjà intéressants émergent des considérations présentées.

3.3.1 Sensibilité au bruit

La méthode d'estimation du déplacement $3-D$ est sensible au bruit sur les coordonnées des points et surtout aux faux appariements, issus d'erreurs dans la phase de suivi de pixels. Malgré le soin apporté à la sélection des pixels et les précautions prises lors du suivi de chacun d'eux, des erreurs sont inévitables. La méthode d'estimation se doit donc d'être capable de détecter ces erreurs.

[Chaudhuri 1991] a montré que le moindre carré contraint utilisé est la meilleure méthode vis à vis de ces considérations, c'est-à-dire la méthode la *moins* sensible à un taux plutôt élevé de données aberrantes. Néanmoins, il est évident qu'un filtrage permettrait d'améliorer la précision des estimations si ces mauvais appariements s'avéraient trop nombreux et l'étude de méthodes telles que les moindres carrés *médians* pourraient certainement augmenter la robustesse de la phase d'estimation de position.

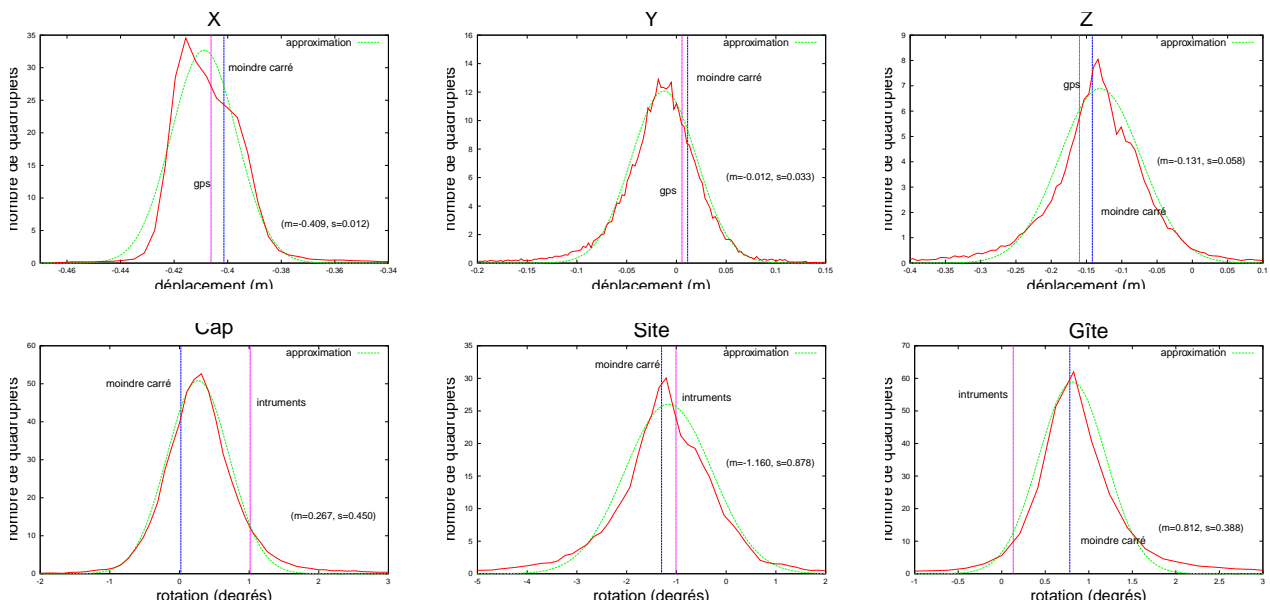


FIG. 3.11 – Analyse du niveau de bruit pour une estimation de déplacement. Ces histogrammes ont été réalisés en estimant le déplacement à l'aide de quadruplets de pixels sélectionnés dans un ensemble de pixels suivis et appariés entre deux paires stéréoscopiques. Pour chacun des paramètres estimés, on remarque qu'une approximation gaussienne est valable : la moyenne et l'écart-type de ces distributions sont précisées sur chacun des graphiques.

La figure 3.11 présente une analyse effectuée sur deux paires stéréoscopiques successives, pour lesquelles environ 300 appariements ont été effectués. Parmi ces appariements, environ 50 000 quadruplets ont été sélectionnés aléatoirement et une estimation du déplacement a été réalisée de façon individuelle pour chacun des quadruplets². Des histogrammes ont ensuite été calculés sur chacun des six paramètres des déplacements ainsi estimés, permettant une visualisation du bruit résultant à la fois de la stéréo-vision et des faux appariements.

On remarque que sur la paire d'images choisie, l'ensemble des déplacements individuels s'approche très bien par une distribution normale, autour de la solution donnée

²trois paires de points suffisent théoriquement pour effectuer une estimation mais les cas dégénérés sont alors plus fréquents.

par le GPS (dont le bruit n'est pas représenté sur les figures). Les faux appariements ne sont pas majoritaires, mais semblent malgré tout influencer la solution *globale* trouvée (indiquée « moindre carré » sur les figures) puisque celle-ci se trouve décalée par rapport aux pics des histogrammes.

Il semble raisonnable de penser que la solution la plus proche du déplacement réel se trouve au niveau du pic étant donné que le bruit sur les coordonnées des points est *a priori* sans biais. Il serait donc certainement bénéfique, en termes de précision, de réaliser un filtrage des appariements s'écartant *trop* du pic principal. Pour cela, la construction de ces histogrammes doit être réalisée en ligne, avant chaque estimation. Même si la méthode d'estimation est extrêmement rapide (comparée à la phase de suivi des pixels) cette procédure peut tout de même s'avérer pénalisante vis à vis du temps de calcul.

3.3.2 Précision intrinsèque

La méthode de moindres carrés contraints possède l'inconvénient de ne pas fournir, parallèlement au calcul de la solution la plus vraisemblable, une estimation de la précision de la solution obtenue.

Une méthode envisageable pour obtenir en ligne une estimation de cette précision consiste à calculer une erreur à partir des deux ensembles de points et de la solution trouvée, par exemple en mesurant la distance moyenne entre chaque paire de points 3-D. Le problème est que cette grandeur est scalaire et ne permet pas d'exhiber une précision sur chacun des paramètres estimés.

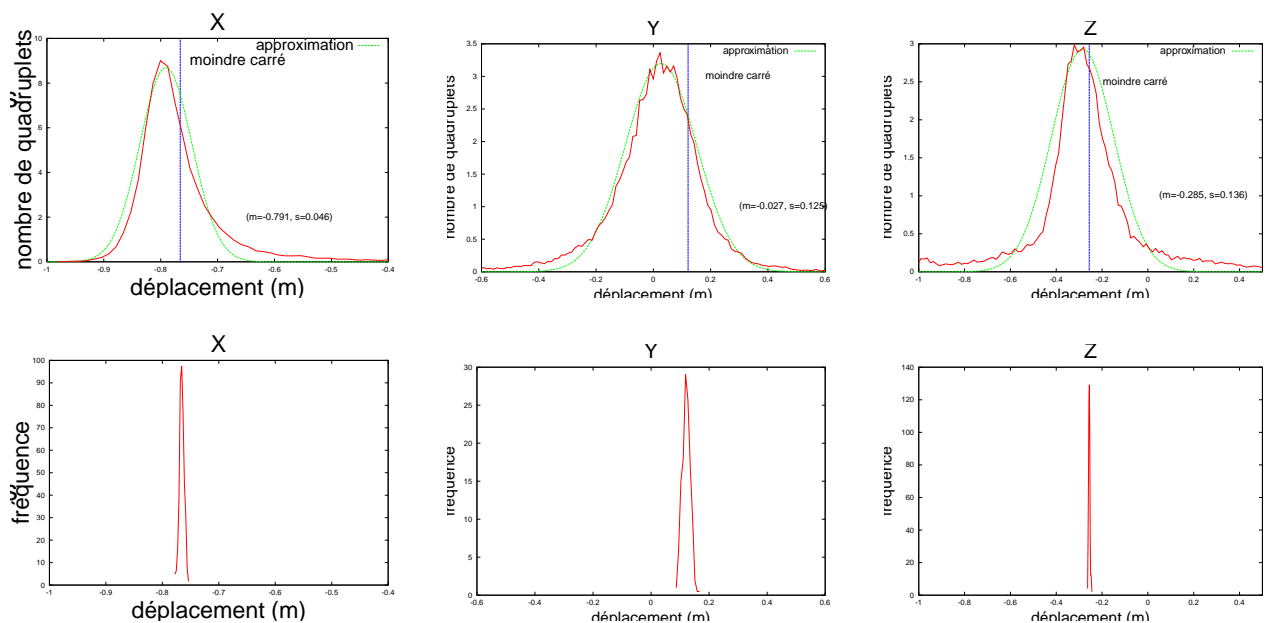


FIG. 3.12 – Comparaison entre le bruit sur les points 3-D et la précision de la solution. La ligne supérieure montre trois histogrammes analogues à ceux de la figure 3.11 et représentant donc le bruit sur les coordonnées 3-D des appariements. La ligne inférieure montre trois autres histogrammes représentant le bruit sur l'estimation d'un déplacement entre deux ensembles d'une vingtaine d'images chacun. On remarque que la précision de l'estimation (ligne du bas) est bien meilleure que ne le laisse penser le bruit sur les appariements (ligne du haut).

Une autre piste serait d'analyser les histogrammes présentés dans le paragraphe précédent pour en déduire, à l'aide d'une approximation gaussienne, une variance associée à la solution. Pour cela, deux séries d'une vingtaine d'images chacune ont été acquises à

deux positions identiques et l'estimation du déplacement a été effectuée entre toutes les paires d'images possibles (soit environ 400 estimations). Les différentes solutions trouvées sont représentées sur les courbes du bas de la figure 3.12. Parallèlement à cela, des histogrammes représentant le bruit sur les appariements 3-D d'une paire particulière d'image ont été calculés et sont représentés sur la ligne du haut de cette même figure.

L'observation de ces figures montre qu'il n'existe pas de corrélation entre la précision de la solution — visualisée par la largeur des histogrammes de la ligne du bas — et le bruit sur les appariements 3-D — visualisé par la largeur des histogrammes de la ligne du haut. Autrement dit, le modèle de l'erreur que l'on pourrait calculer en ligne serait très pessimiste (mais néanmoins vrai) et aboutirait à un manque de confiance vis à vis de la solution trouvée.

3.3.3 Influence de la quantité d'appariements

Le nombre de pixels suivis dans les images est d'une part déterminant pour le temps de calcul nécessaire à l'estimation des déplacements, et d'autre part critique pour la précision de la solution. Suivre le nombre minimal de pixels vis à vis d'une précision souhaitée est donc primordial.

La figure 3.13 montre le résultat d'un grand nombre d'estimations de position, réalisées entre deux couples d'une vingtaine de paires d'images (comme dans la section précédente). Pour chacune de ces estimations, le nombre de pixels utilisés a varié de 50 à 600. On remarque nettement l'augmentation de la précision avec le nombre de points.

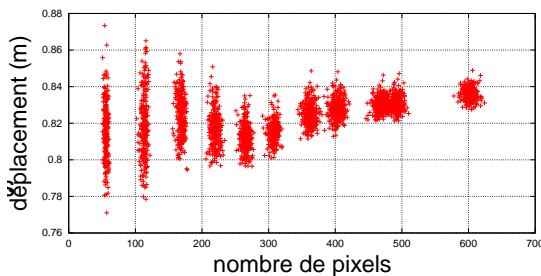


FIG. 3.13 – Influence du nombre d'appariements. Ce graphique a été réalisé à l'aide d'une vingtaine de paires d'images acquises en deux positions identiques. Pour chacun des couples d'images, l'estimation du déplacement a été calculée avec un nombre de points variable. On constate que plus on utilise de points, moins la dispersion de la solution est grande.

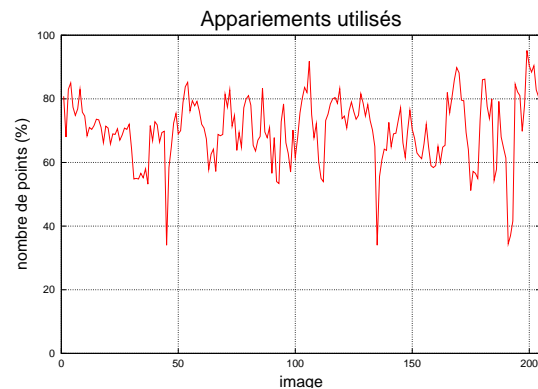


FIG. 3.14 – Quantité d'appariements effectivement utilisés pour le calcul du déplacement : à partir d'une consigne de 500 points à suivre entre chacune des paires d'images prises au cours de l'exécution de la trajectoire de la figure 3.6, le graphique montre que seulement 60 à 80 % peuvent être effectivement utilisés pour l'estimation du déplacement.

La quantité d'appariements actuellement utilisée est de 500 points. Ce nombre, déterminé empiriquement, permet d'obtenir une certaine robustesse dans le calcul et évite les solutions aberrantes parfois trouvées lorsque trop peu d'appariements sont utilisés. La figure 3.14 montre qu'en partant de cette consigne de 500 points, seulement 60 à 80% ont effectivement pu être suivis au cours de la trajectoire de la figure 3.6. Ce taux de pertes élevé provient essentiellement des pixels sortant du champ de vue du robot au cours du déplacement et, dans une moindre mesure, des seuils sur le résultat de la corrélation. Si, d'une part, un filtrage efficace des faux appariements est mis en place

et si, d'autre part, le mouvement estimé par l'odométrie pouvait servir à restreindre la zone dans laquelle les pixels à suivre sont sélectionnés (zone commune aux deux images servant à l'estimation), il devrait être possible de réduire ce taux de pertes, permettant ainsi de suivre moins de pixels.

3.3.4 Influence de la vitesse du robot

La vitesse du robot est également un facteur déterminant pour la précision de la méthode. En effet, la distance séparant deux paires d'images consécutives détermine leur degré de ressemblance et donc la qualité des appariements effectués.

De plus, le modèle d'erreur sur la stéréo-vision indique que le bruit sur les coordonnées des points croît avec le carré de la distance. Plus le robot est rapide, plus les caractéristiques des points $3-D$ mis en correspondance seront différentes.

Les courbes de la figure 3.15 montrent l'influence de cette vitesse : on constate qu'à faible vitesse (3 ou 6 $cm.s^{-1}$) l'estimation de position est bonne, mais qu'au delà elle décroît de façon nette. Le seul moyen d'obtenir une bonne précision à haute vitesse semble donc être la diminution des temps de calculs.

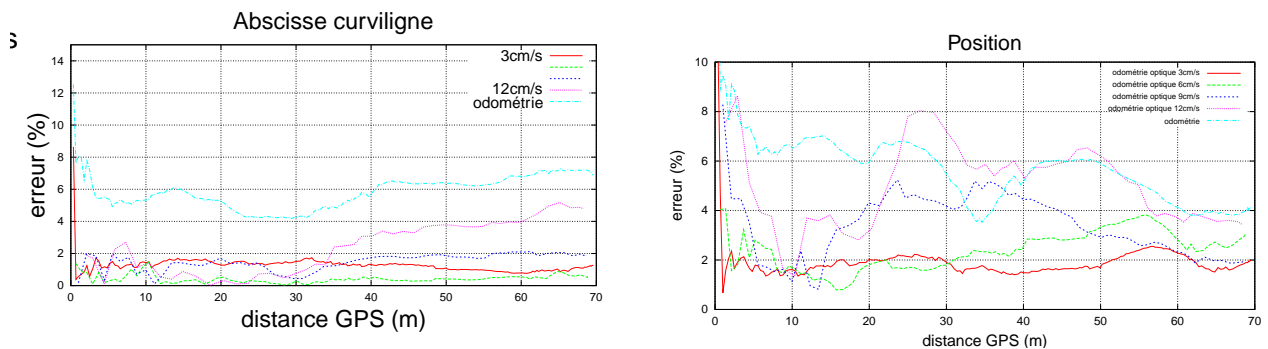


FIG. 3.15 – Influence de la vitesse du robot. On remarque qu'à faible vitesse l'estimation de position se comporte bien : les erreurs à 3 ou 6 $cm.s^{-1}$ sont approximativement équivalentes. En revanche, lorsque la vitesse croît, on observe une nette dégradation des performances.

3.3.5 Autres paramètres

Il existe quantité d'autres paramètres pour lesquels l'influence sur la précision ou la robustesse n'est pas encore établie. Les valeurs actuellement utilisées ont été choisie par « défaut » et « habitude », c'est-à-dire qu'aucune analyse n'a permis d'exhiber des valeurs à préférer, mais que celles-ci ont été choisies en fonction de considérations telles que des temps de calculs ou des volumes de données raisonnables. L'étude avec le CNES, dans le cadre de [Lacroix 2001d], devrait permettre d'obtenir plus d'informations.

Taille des images. La carte d'acquisition présente sur *Lama* (annexe A.4) produit des images vidéo jusqu'à 768×576 pixels. Cependant, pour obtenir un temps de stéréo-corrélation raisonnable (environ 1 seconde), elles sont réduites d'un facteur 3 (256×192 pixels) et les images $3-D$ produites ont donc la même taille. La précision ainsi obtenue sur les images $3-D$ n'est donc pas optimale, mais le volume de données déjà important (environ 1 *Mo* pour une image pleine) permet difficilement d'en augmenter la taille. Une solution envisageable, si la résolution apparaissait comme un facteur critique, serait de ne calculer les points $3-D$ que pour les pixels sélectionnés pour l'estimation de mouvement.

Position des caméras. Un paramètre qui devrait être important est la position des caméras relativement au mouvement. D'après le modèle d'erreur sur la stéréovision et les expérimentations présentées dans ce chapitre, il semble qu'une disposition *perpendiculaire* au mouvement soit préférable : le mouvement semble mieux estimé dans ce plan, et elle permet en tout cas de plus grands déplacements sans modifier l'allure des images vidéo. L'expérience menée avec le ballon dirigeable tend à confirmer cette hypothèse.

Taille des fenêtres de corrélation. Ce paramètre a une influence importante (quadratique) sur le temps de calcul et il s'agit avant tout de trouver un compromis entre précision, robustesse et rapidité (voir annexe ??). Actuellement, une taille de 9×9 pixels est utilisée et donne de bon résultats sur une grande variété de caméras. L'utilisation prochaine de caméras numériques devrait améliorer la qualité des images et donc permettre de réduire cette taille à des fenêtres 7×7 ou même 5×5 .

Taille des zones de recherche. Actuellement, les zones de recherches sont de taille fixe, et relativement grandes (20×20). Il est possible de réduire énormément cette taille en utilisant d'une part l'information *3-D* et la prédiction du mouvement (les pixels les plus éloignés des caméras se déplacent le moins) mais aussi le résultat du suivi sur les premiers pixels. À partir d'une certaine quantité de pixels retrouvés, il est possible de procéder à une première estimation de position, et donc réduire énormément la taille des zones de recherche. Le gain de temps peut être, là aussi, très important.

3.4 Conclusions

La méthode d'odométrie optique proposée s'avère donc être très performante ; à tel point que les résultats paraissent surprenants, au regard du nombre de facteurs entrants en compte dans le calcul des déplacements. Pourtant, de nombreuses améliorations potentielles restent encore à évaluer et implémenter.

Ce travail, commencé dans le cadre de cette thèse, se poursuit donc dans celui du transfert de la méthode vers le CNES, qui s'est immédiatement montré intéressé. Un doctorant du LAAS est d'ores et déjà chargé de l'évaluation des améliorations présentées dans la dernière section de ce chapitre.

Des possibilités liées au contrôle fin des déplacements des plateformes seraient envisageables à court terme, si la méthode pouvait fonctionner à plusieurs *Hz* (voir le paragraphe 1.1.1 du premier chapitre). On pourrait également imaginer des dispositifs *portatifs* de cartographie ou de mesures, promenés à bout de bras et construisant des modèles *3-D* de terrains... Le poids et la miniaturisation des équipements électroniques font, qu'à terme, les applications d'une telle méthode sont nombreuses ; l'estimation de la position est *a priori* possible dès que l'environnement est texturé.

IV

Les cartes d'élévation

La construction de cartes a pour objectif la modélisation de l'environnement grâce à l'analyse de l'ensemble des perceptions du robot. Elle permet de fournir des *représentations* aux couches délibératives et donne ainsi à la machine la capacité de *raisonner* et de *comprendre* le terrain sur lequel elle évolue.

De telles cartes peuvent être fournies par des systèmes extérieurs au robot : dans le cadre d'applications terrestres, des données sont aisément disponibles par imagerie aérienne ou satellite. Ces données initiales sont, cependant, le plus souvent *relativement pauvres* ou acquises à des résolutions insuffisantes pour que le robot soit à même de les utiliser localement, dans l'environnement immédiatement perceptible autour de lui. De plus, le robot doit être capable de réagir face à des modifications ou des événements imprévus et non initialement modélisés. Il est donc *toujours* nécessaire de construire, de façon incrémentale, un modèle à l'échelle du robot. La disponibilité éventuelle de données initiales ne constitue qu'un *plus* pouvant servir, par exemple, à conforter ou enrichir les modélisations locales.

En environnements naturels, les capteurs les plus performants sont sans aucun doute les caméras vidéo et les capteur 3-D de type stéréo-vision ou laser. Ceux-ci fournissent des données *denses* et permettent d'appréhender une grande partie de l'environnement en une seule perception *cohérente*. Cependant, ils sont la plupart du temps incapables de percevoir la totalité de l'environnement visible autour du robot, notamment à cause des occultations dues à la géométrie du terrain. L'utilisation d'images uniques n'est donc pas suffisante et il est nécessaire de fusionner différentes perceptions successives acquises au cours du déplacement du robot.

Les cartes d'élévation sont l'une des nombreuses représentations géométrique de l'environnement et permettent notamment la fusion de plusieurs perception en un modèle global *cohérent*. Elles modélisent la géométrie de l'environnement sous la forme d'une grille cartésienne, généralement définie dans un plan horizontal. En chaque cellule de cette grille est représentée l'altitude moyenne de la zone couverte par la cellule. Dans le contexte des environnements naturels, cette représentation est particulièrement intéressante puisqu'elle fournit un moyen de structurer les données brutes. Les cartes

d'élévation sont généralement utilisées comme support pour d'autres données de type continu (vidéo) ou discret (amers). Un problème majeur de cette technique est que l'environnement se trouve être modélisé par une fonction du type $z = f(x, y)$. Il n'est donc pas possible de représenter des parois verticales, ni même des surplombs.

Les algorithmes pouvant exploiter ces cartes sont nombreux : il existe maintenant des contributions particulièrement performantes en planification de mouvements ou de trajectoires [Bonnafous 2001, Hait 1999] pour des environnements accidentés ou encore dans le domaine de la planification de stratégies d'exploration et de perception.

La construction de telles cartes est étroitement liée au problème de la *localisation* : il est impossible de créer des cartes précises si le robot ne sait pas où il se trouve. On parle ainsi souvent de « *modélisation et localisation simultanées* » (voir section 1.3.3 dans le premier chapitre) et les cartes d'élévation sont à la base d'un large éventail de fonctionnalités dans ce domaine : localisation au cours du déplacement, détection d'amers, localisation par rapport à un modèle initial, ...

Sans toutefois prétendre à l'universalité, les cartes d'élévation sont donc des modèles polyvalents. Leur étude nous a semblée pertinente dans le contexte de ce manuscrit et ce chapitre présente une méthode de construction *incrémentale* de cartes d'élévation, initialement développée sur, et pour, le robot *Lama*. L'environnement est perçu à l'aide d'un banc stéréoscopique : la figure 3.3 montre un exemple d'image *3-D* brute utilisée comme base pour la construction des cartes.

Le premier problème abordé concerne la représentation et la modélisation des incertitudes liées aux données. La section 4.1 propose pour cela une approche basée sur les grilles d'occupation en deux dimensions et montre comment, à partir de l'observation du comportement de celles-ci, il est possible d'en déduire une modélisation simplifiée et implémentable des incertitudes pour une carte *3-D*.

La méthode de construction proprement dite, telle qu'elle est implémentée actuellement, et ses différents aspects concrets sont également présentés : choix de la résolution, nature des informations incluses dans la carte et filtrage des données les plus incertaines.

Afin d'être capable de gérer des modèles de grande taille pour l'exploration et la navigation à long terme, une structure particulière encapsule les cartes construites localement et permet au modèle d'être robuste face à des erreurs de localisation (section 4.2).

Enfin, une fonctionnalité de localisation locale et à long terme est intégrée pour permettre de maintenir la cohérence spatiale des modèles construits (section 4.3).

4.1 Modélisation de l'imprécision et grilles d'occupation

Tous les capteurs *3-D* fournissent des informations imprécises et entachées d'erreurs ; la difficulté majeure rencontrée lors de la construction des cartes d'élévation consiste à représenter fidèlement cette imprécision. L'espace des perceptions est le plus souvent bien représenté en coordonnées polaires et, selon la disposition du capteur, la modélisation des erreurs des points *3-D* dans l'espace cartésien lié à la grille est délicate. Une bonne représentation permettra de manipuler de façon précise et rigoureuse les données contenues dans le modèle et fournira les bases d'une qualification des algorithmes exploitant ces modèles (voir section 4.3). Cette qualification, indispensable, permettra d'établir un modèle des algorithmes et permettra ainsi de propager, le long des chaînes de traitement, la nature incertaine et non-déterministe des perceptions.

La figure 4.1 illustre de façon schématique le problème en deux dimensions lorsque le modèle d'erreur statistique sur les points *3-D* est représenté sous forme d'une distribution de probabilité gaussienne (ce modèle est détaillé en annexe C pour le cas

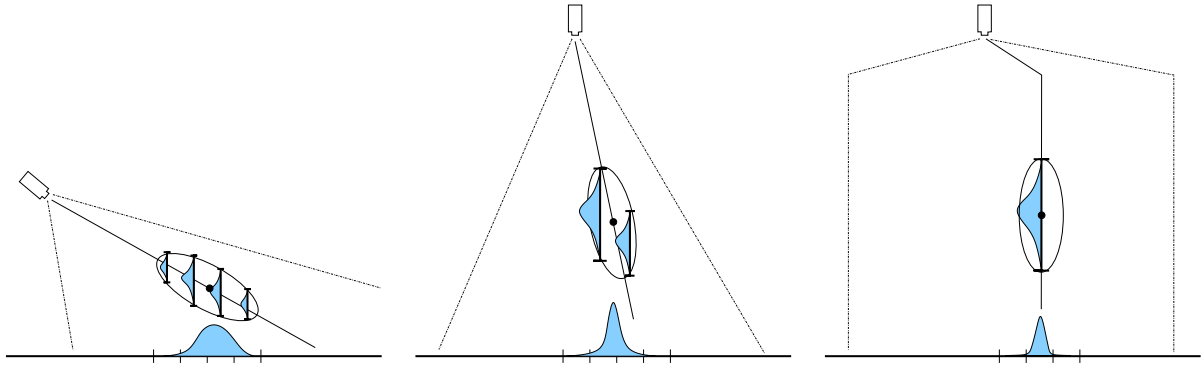


FIG. 4.1 – Projection d'un point 3-D imprécis dans une carte d'élévation : les zones hachurées sont les distributions de probabilité associées aux altitudes des cellules de la carte, selon chacun des axes principaux. La configuration du capteur détermine l'adéquation entre le modèle d'erreur dans le repère du capteur et son interprétation dans l'espace de la carte (de gauche à droite : acquisition rasante, acquisition verticale à faible altitude, acquisition verticale à très haute altitude).

particulier de la stéréo-vision). Selon la configuration du capteur, la structure discrétisée de la grille fait qu'une seule information (fournie par un seul point 3-D) doit être représentée dans plusieurs cellules de la carte. Si cette information est bien représentée par une distribution de probabilités dans le système de coordonnées du capteur, il n'en est plus de même lorsque les caméras perçoivent le terrain de façon *rasante*.

Le schéma de gauche sur la figure 4.1 montre par exemple que les extrémités de l'ellipsoïde d'incertitude se projetteraient en un point 3-D relativement précis si le point mesuré de l'environnement se trouvait effectivement à cet endroit. Cependant, la probabilité que le point se trouve *effectivement* à l'extrémité de l'ellipse est très faible. On observe donc une *décomposition* de la grandeur unique représentant l'incertitude en deux grandeurs : l'*imprécision* d'une part et l'*incertitude* d'autre part.

Le cas où le capteur est globalement orthogonal au plan de construction de la carte (schéma de droite sur la figure 4.1) est beaucoup mieux conditionné : un point 3-D se projette dans, en général, beaucoup moins de cellules et sa représentation sous forme d'une distribution de probabilités est beaucoup plus adaptée.

Il est intéressant d'étudier comment les grilles d'incertitude (ou grilles d'occupation), habituellement utilisées pour modéliser des environnements intérieurs, peuvent apporter un moyen de valider les différentes approches utilisées pour la construction de cartes d'élévation.

En tant que telles, elles ne constituent pas un moyen réaliste de modéliser un environnement tridimensionnel de par le simple fait qu'une discrétisation de l'espace n'est pas, à l'heure actuelle, envisageable en termes de quantité d'information ou de vitesse de traitement. En revanche, il se trouve que les données que l'on cherche à manipuler se trouvent être particulièrement bien représentées par des probabilités d'occupation, et l'étude du comportement de ces grilles peut fournir un moyen de valider des méthodes moins coûteuses et réalistes.

Cette section se propose donc de montrer comment, dans le cas simplifié de deux dimensions, il serait possible de construire un modèle du monde grâce à cette technique. Deux approches classiques sont étudiées (approche bayésienne et de Dempster-Shafer) et une solution pour passer de ces représentations à une carte d'élévation est ensuite proposée.

Pour des raisons de clarté, l'analyse faite ici considère uniquement la reconstruction d'un profil de terrain en deux dimensions de type $z = f(x)$ comme l'illustre la figure 4.2.

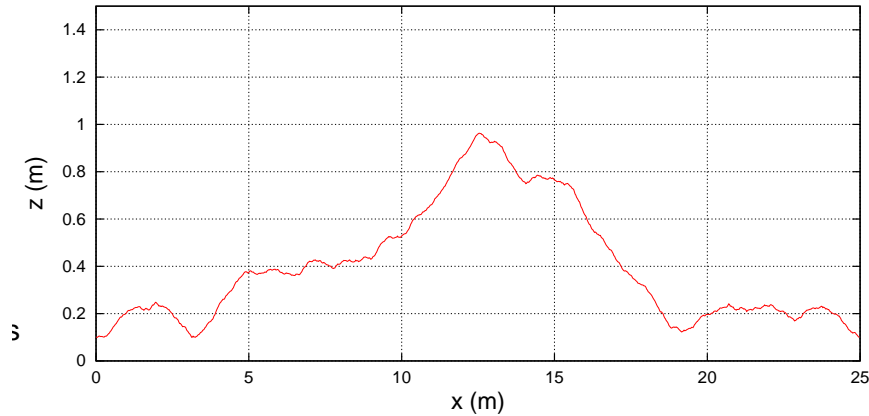


FIG. 4.2 – Profil de terrain 2-D simulé pour la construction d'une grille d'occupation. La résolution horizontale est beaucoup plus grossière que la résolution verticale afin de permettre une comparaison réaliste avec les cartes d'élévation.

Ce profil correspondrait à une coupe verticale de l'environnement 3-D perçue par le robot.

Sur ce terrain, un banc stéréoscopique est simulé, et des perceptions peuvent être effectuées. Les paramètres de ce banc ont été choisis pour correspondre à celui du robot *Lama* : une perception bornée à une distance de 15 m et un champ de vue de 45° comportant 200 pixels pour lesquels un bruit blanc gaussien dont l'écart-type est proportionnel au carré de la distance est ajouté. Le coefficient de proportionnalité choisit est de 0.01, ce qui est supérieur d'un facteur 10 au modèle d'erreur déterminé empiriquement sur le robot *Lama*.

Le plan (z, x) est discrétisé en cellules et une grille d'occupation peut ainsi être construite. Le taille horizontale des cellules est de 2.5 cm, et la résolution verticale a été choisie très fine (3 mm), pour être ensuite capable d'analyser les distributions de probabilité des altitudes sur des colonnes $x = \text{constante}$ et ainsi retrouver un modèle de cartes d'élévation.

4.1.1 Un modèle de capteur pour la stéréo-vision

Afin de modéliser les données perçues et l'imprécision qui les caractérise, il est nécessaire de construire un modèle du capteur utilisé. Le rôle de ce modèle est de décrire la *nature* des données perçues, c'est à dire l'information fournie par chaque perception.

Pour le cas de la stéréo-vision, un tel modèle est difficilement calculable de façon analytique : la compréhension du système physique, électronique et logiciel sous-jacent est un processus trop complexe pour permettre de d'en déduire d'une façon rigoureuse un modèle. Cependant, en combinant expérience et analyse du phénomène physique mesuré, il est possible d'obtenir un modèle réaliste.

Essentiellement, ce capteur perçoit une distance. Cette distance est ensuite interprétée comme étant la signature d'un objet de l'environnement, situé à cette distance. Étant donné un point 3-D et l'écart-type modélisant sa *précision* (la façon dont cet écart-type est calculé est présentée en annexe C), le modèle de son *incertitude* peut alors être représenté comme sur la figure 4.3.

Une grandeur comprise entre -1.0 et 1.0 , que nous appellerons « *confiance* », permet de représenter le caractère « flou » de la perception. Une confiance de 1.0 indique

qu'il est tout à fait probable qu'un point réel de l'environnement se trouve à l'endroit considéré et une confiance de -1.0 indique qu'il est par contre tout à fait probable qu'il n'y ait *pas* de point à cet endroit. Une confiance égale à 0.0 représente l'incertitude maximale.

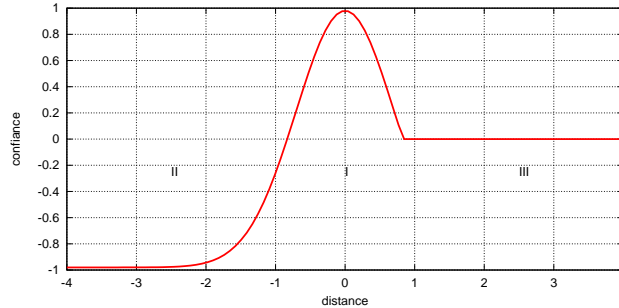


FIG. 4.3 – Modèle de l'incertitude sur les points issus de stéréovision (la distance est centrée sur la position mesurée du point et exprimée en multiple de l'écart-type associé).

Le modèle d'erreur du banc stéréoscopique se compose de trois parties. La première (notée I sur la figure 4.3) modélise le fait qu'il est plutôt *probable* qu'un point de l'environnement se trouve *effectivement* dans cette zone : la confiance est positive. La seconde partie (II) représente quant à elle le fait qu'il n'y a *probablement pas* de points de l'environnement dans cette zone, étant donné qu'une mesure a été faite à une distance supérieure : la confiance est négative. Enfin, la dernière partie (III) indique que l'occultation due au point mesuré interdit toute conclusion quant à la présence ou l'absence de point : la confiance est nulle.

Il est possible d'exprimer analytiquement ce modèle Ω en fonction de la distance d et de l'écart-type σ du point perçu grâce à une gaussienne :

$$\Omega_{d,\sigma}(x) = \begin{cases} 0 & \text{si } x > d + \sigma\sqrt{2\ln 2}, \\ 2\cdot e^{-\frac{1}{2}\left(\frac{x-d}{\sigma}\right)^2} - 1 & \text{sinon.} \end{cases} \quad (4.1)$$

Bien que ce modèle soit assez classique, il n'est en rien issu d'un formalisme quelconque. Il se contente de représenter une *connaissance* sur le système, à l'aide d'une grandeur appelée ici *confiance*. La forme même du modèle est arbitraire : une gaussienne se prête bien au phénomène que nous voulons modéliser, mais il est possible d'imaginer toute autre forme. Seule l'expérience sur des données réelles permet de valider un choix particulier.

Un premier problème se pose donc immédiatement lors de la construction de la carte de l'environnement, lorsqu'il est nécessaire de passer de cette grandeur intuitive — la confiance — à un modèle mathématique de type statistique ou probabiliste, par exemple. Selon le formalisme choisi, l'interprétation du modèle du capteur sera différente : les deux sections suivantes présentent deux façons d'interpréter ce même modèle. Une analyse de ce problème pour l'utilisation de capteur ultra-son est présentée dans [Murphy 2000] et a servi de base à l'élaboration de ces simulations.

4.1.2 Approche bayésienne

La méthode la plus employée pour la construction de grilles d'occupation consiste à traduire le modèle de capteur en probabilités et de combiner ces probabilités grâce

aux règles de Bayes.

Considérons à nouveau la figure 4.3 : moyennant une transformation *adéquate* (la plus triviale étant par exemple une mise à l'échelle entre 0 et 1), il est possible de considérer qu'elle fournit la probabilité P que le banc stéréo retourne la distance d si la cellule de la grille d'occupation située à la distance 0 était réellement occupée. Si l'on note s une mesure particulière fournie par la stéréo-vision, H l'hypothèse qu'une cellule donnée est occupée par un point de l'environnement, le modèle de capteur peut donc être interprété comme la probabilité conditionnelle $P(s|H)$, c'est-à-dire la probabilité que le capteur donne la distance d si la cellule H considérée était effectivement occupée.

Avec les mêmes notations, la probabilité que l'on souhaite calculer pour chaque cellule de la grille d'occupation est $P(H|s)$, c'est-à-dire la probabilité que chaque cellule de la grille soit vide ou occupée par un point, étant donnée une mesure fournie par la stéréo-vision. La règle de Bayes permet de passer du modèle du capteur à la grille d'incertitude :

$$P(H|s) = \frac{P(s|H).P(H)}{P(s|H).P(H) + P(s|\neg H).P(\neg H)} \quad (4.2)$$

$P(H)$ et $P(\neg H)$ sont les probabilités d'occupation *a priori*. Dans le cas d'un environnement initialement inconnu, pour lequel on ne dispose pas d'information particulière, il est d'usage de considérer que $P(H) = P(\neg H) = 0.5$. L'équation 4.2 peut donc s'écrire très simplement :

$$P(H|s) = P(s|H)$$

La règle de Bayes, ainsi que l'indépendance des mesures effectuées au cours des différentes perceptions, permet de combiner deux perceptions s_1 et s_2 au sein de la grille :

$$P(H|s_2, s_1) = \frac{P(s_2|H).P(H|s_1)}{P(s_2|H).P(H|s_1) + P(s_2|\neg H).P(\neg H|s_1)} \quad (4.4)$$

La difficulté consiste à trouver comment exprimer le modèle du capteur en termes de probabilités d'occupation. L'inconvénient majeur du formalisme de Bayes est qu'il ne permet pas de représenter *explicitement* l'inconnu : une cellule de la grille est soit vide, soit occupée, et l'incertitude émerge lorsque cette probabilité se rapproche de 0.5. Ce formalisme fait donc une confusion entre *incertitude* et *inconnu*.

Il est néanmoins possible de construire une grille d'occupation réaliste, au prix de l'utilisation de « recettes » *ad hoc*. L'idée retenue ici permet d'atténuer autour de 0.5 la valeur maximale ou minimale des probabilités issues d'une perception pour modéliser l'*incertitude* liée à l'éloignement des perceptions.

Pour calculer $P(s|H)$ à partir du modèle Ω , nous affirons qu'un point perçu à grande distance doit avoir moins de poids dans les règles de mise à jour qu'un point perçu à faible distance : pour une cellule située à la distance x du capteur on définit

$$P(s|H)(x) = \frac{1}{2} + \lambda_d \cdot \left(p(x) - \frac{1}{2} \right)$$

avec

$$\begin{aligned} p(x) &= P_{max} \cdot \left[\frac{1}{2} (\Omega_{d,\sigma}(x) + 1) \right] \\ \lambda_d &= \frac{d_{max} - d}{d_{max}} \end{aligned}$$

λ_d traduit l'incertitude croissante sur la qualité de la mesure en fonction de la distance et P_{max} représente la probabilité maximale que l'on autorise pour une cellule. Cette valeur permet d'affirmer que toute mesure est potentiellement erronée et donc qu'une probabilité de 1 est impossible. On peut remarquer que cette considération, bien que raisonnable, provient essentiellement du fait que les règles bayésiennes de mise à jour se comportent mal vis à vis de probabilités égales à 0 ou 1. Dans la pratique, $P_{max} = 0.98$ semble un bon choix.

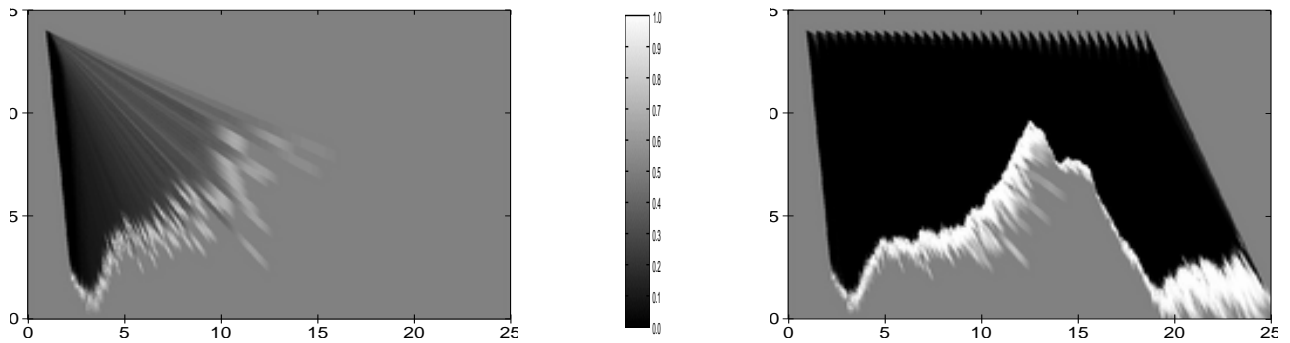


FIG. 4.4 – Construction d'une grille d'occupation avec une approche bayésienne : la figure de gauche montre le résultat de la reconstruction issue d'une perception effectuée sur le terrain montré sur la figure 4.2 ; la figure de droite est le résultat de la fusion de 36 perceptions régulièrement espacées. Les probabilités d'occupation sont représentées sur une échelle de niveaux de gris (figurée au centre) allant de 0 à 1 où 1 représente la présence certaine d'un point de l'environnement.

La figure 4.4 montre le résultat d'une reconstruction après 1 et 36 perceptions simulant le mouvement d'une caméra évoluant sur le terrain virtuel présenté dans la figure 4.2. On peut d'ores et déjà constater que les limites de la surface reconstruite sont relativement floues, et les probabilités semblent être très « binaires ». La figure 4.5 confirme cette impression en montrant un profil extrait le long d'une colonne située en $x = 11$ m.

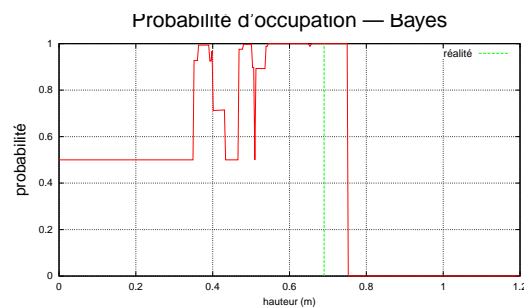


FIG. 4.5 – Exemple de profil reconstruit avec le formalisme de Bayes : le profil représente la probabilité associée à chaque altitude de la grille pour $x = 11$ m. Le trait vertical montre l'altitude réelle du terrain à l'endroit considéré.

On remarque immédiatement que la modélisation n'est pas satisfaisante : le bruit relativement élevé des perceptions et la quantité importante d'informations fusionnées dans la grille font que les probabilités se sont « binarisées ». C'est là une signature caractéristique de ce formalisme : les règles de mise à jour font que la fusion d'informations compatibles renforce énormément la confiance en ces informations et la difficulté liée à la complémentarité des probabilités *vide* et *occupé* rendent la modélisation du problème délicate.

4.1.3 Approche de Dempster-Shafer

Si la majorité des contributions dans le domaine de la construction de grilles d'occupation utilisent le formalisme de Bayes, certains auteurs critiquent — à juste titre — cette approche [Tirumalai 1995]. Pour les raisons évoquées dans le paragraphe précédent, la modélisation du problème de reconstruction n'est pas évidente lorsqu'on utilise ce formalisme. Même si les résultats obtenus sont relativement similaires avec les deux approches, la modélisation apparaît ici beaucoup plus naturelle.

Le formalisme de Dempster diffère de celui de Bayes de par le fait que des valeurs distinctes sont assignées à une proposition et sa négation ; ces valeurs sont alors appelées *croyances*. Dans le cas des grilles d'occupation, l'ensemble des croyances peut être formé par les propositions $C = \{ \text{occupé}, \text{vide}, \text{inconnu} \}$.

Les croyances associées à chaque propositions sont comprises entre 0 et 1 et Dempster propose un mécanisme de fusion pour deux ensembles de croyances C_1 et C_2 en une distribution de croyances $C_{1,2}$ représentant un consensus :

$$C_{1,2}(i) = \frac{1}{1 - \sum_{j \cap k = \emptyset} C_1(j) \cdot C_2(k)} \times \sum_{j \cap k = i} C_1(j) \cdot C_2(k) \quad (4.5)$$

où i, j et k représentent des propositions de C et $C(i)$ la croyance associée à la proposition i .

On remarque donc que ce formalisme permet de façon immédiate de distinguer l'absence de données ($\text{inconnu} = 1$) de l'incertitude maximale ($\text{vide} = \text{occupé} = 0.5$, par exemple). De plus, la gestion des perceptions incompatibles est explicitement écrite, permettant de *remettre en cause* les données déjà présentes dans la grille.

Le calcul de C , pour une perception, à partir du modèle Ω est donc similaire à celui effectué pour la construction Bayésienne, à ceci près que l'incertitude se reporte sur la croyance C_{inconnu} :

$$\begin{aligned} C_{\text{vide}}(x) &= \lambda_d \cdot (1 - p(x)) \\ C_{\text{occupé}}(x) &= \lambda_d \cdot p(x) \\ C_{\text{inconnu}}(x) &= 1 - \lambda_d \end{aligned}$$

avec

$$\begin{aligned} p(x) &= P_{\text{max}} \cdot \left[\frac{1}{2} (\Omega_{d,\sigma}(x) + 1) \right] \\ \lambda_d &= \frac{d_{\text{max}} - d}{d_{\text{max}}} \end{aligned}$$

La figure 4.6 montre le résultat de la reconstruction (seule la croyance associée à *occupé* est représentée). On peut observer une différence nette avec la grille obtenue dans le paragraphe précédent : la surface est beaucoup plus précisément définie.

La reconstruction obtenue est beaucoup plus satisfaisante qu'avec le formalisme bayésien. En effet, il a été possible ici de définir des probabilités d'occupation numériquement faibles *sans pour autant* accorder un crédit élevé à l'hypothèse contraire. On remarque donc que l'on obtient une distribution de probabilités beaucoup fidèle à la réalité et surtout qu'il est possible d'approximer cette distribution par une gaussienne dont la moyenne représentera l'altitude du terrain la plus probable et l'écart-type la précision espérée de cette valeur.

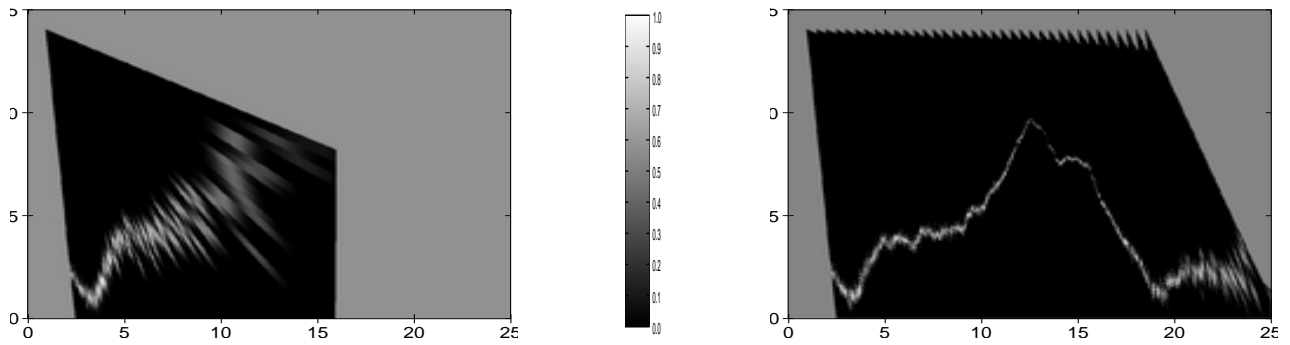


FIG. 4.6 – Construction d'une grille d'occupation avec une approche de Dempster-Shafer : la figure de gauche montre le résultat de la reconstruction issue d'une perception effectuée sur le terrain montré sur la figure 4.2 ; la figure de droite est le résultat de la fusion de 36 perceptions régulièrement espacées. Seule la croyance associée à la proposition « la cellule est occupée » est représentée, sur une échelle allant de 0 à 1 où 1 représente la présence certaine d'un point de l'environnement.

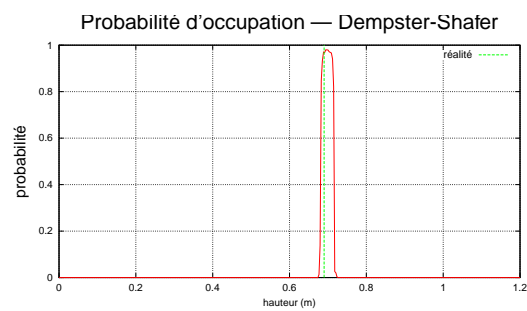


FIG. 4.7 – Exemple de profil reconstruit avec le formalisme de Dempster-Shafer : le profil représente la probabilité associée à chaque altitude de la grille pour $x = 11$ m. Le trait vertical montre l'altitude réelle du terrain à l'endroit considéré.

4.1.4 Des grilles d'occupation aux cartes d'élévation

L'objectif de cette section était d'étudier la construction rigoureuse d'une grille d'occupation, dans le but d'obtenir un moyen de valider des approches plus simples mais réalistes pour la construction de cartes d'élévations 3-D.

Dans un premier temps il est donc nécessaire de déduire du modèle probabiliste construit une représentation sous la forme $z = f(x)$. Le terrain simulé ici ne présentant pas de surfaces verticales ou de surplombs, il est possible d'analyser chaque colonne des grilles d'occupation pour en déduire une altitude z unique et la plus probable. Pour cela, nous utilisons la reconstruction obtenue grâce au formalisme de Dempster-Shafer, en approximant les profils de probabilité par des distributions normales.

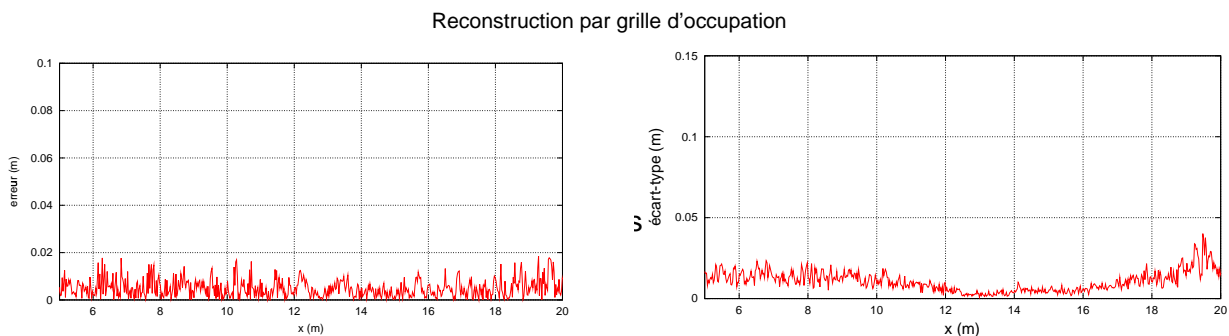


FIG. 4.8 – Reconstruction d'un profil de terrain à partir d'une grille d'occupation. La grille utilisée ici est celle de la figure 4.6, où les distributions de probabilités de chaque colonne $x = \text{cte}$ ont été approximées par une gaussienne. L'erreur de reconstruction est représentée sur le graphique de gauche et la précision (sous forme d'un écart-type) sur celui de droite.

Le résultat de cette reconstruction est présenté sur la figure 4.8. On constate que le profil obtenu est très satisfaisant : l'erreur absolue est toujours inférieure à 2 cm et très souvent de moins de 5 mm. Compte-tenu du bruit créé sur les perceptions (écart-type égal à $0.01 \cdot d^2$, d étant la distance du point perçu), cette méthode semble donc très bien modéliser le problème.

En trois dimensions, l'utilisation d'une grille d'occupation est cependant relativement coûteuse : le volume de données à traiter et la quantité de calculs engendrés sont loin d'être négligeables. Les cartes d'élévation sont donc généralement utilisées et des méthodes de construction plus simples sont mises en place.

La plupart des méthodes de représentation des imprécisions n'utilisent qu'une seule grandeur pour représenter à la fois l'incertitude et l'imprécision, et ces méthodes ne sont donc pas adaptées aux cas des cartes d'élévations : il est nécessaire d'utiliser à la fois une grandeur représentation l'imprécision d'une information et une autre grandeur représentant son incertitude. La méthode que nous présentons ici utilise un écart-type pour représenter l'imprécision, et une probabilité pour représenter l'incertitude.

La plus simple d'entre elles consiste à considérer chaque point 3-D perçu comme exact et simplement moyenner l'altitude dans chaque cellule de la carte. Un écart-type peut également être calculé à partir de l'altitude de tous les points servant au calcul de chaque cellule. La figure 4.9 montre le profil de terrain obtenu par cette méthode.

En comparaison avec le résultat obtenu par la méthode des grilles d'occupation, on remarque que la reconstruction est beaucoup moins fidèle à la réalité, et on observe quelques points pour lesquels une erreur supérieure à 10 cm est commise. D'autre part

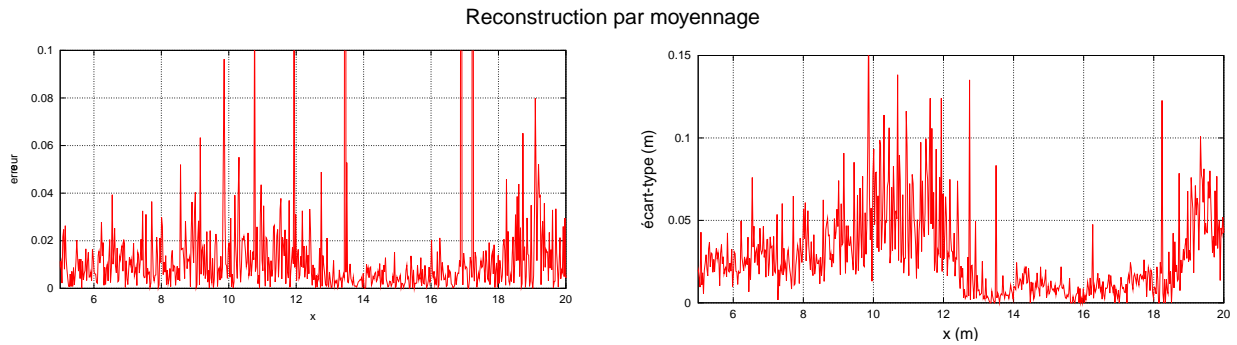


FIG. 4.9 – Reconstruction d'un profil de terrain à partir d'un moyennage. Le profil reconstruit est le même que celui de la figure 4.8 (l'échelle est différente) mais la technique utilisée se contente de moyennner l'altitude dans chaque cellule d'une carte d'élévation.

l'estimation de la précision semble très pessimiste en regard des erreurs effectivement commises.

Une autre méthode permet d'améliorer légèrement ces résultats (figure 4.10). En considérant que les pixels à placer dans la carte représentent une surface définie par quatre pixels voisins, il est possible d'augmenter sensiblement la précision de la reconstruction et, notamment, d'éliminer ces données aberrantes obtenues avec le moyennage simple.

Pour représenter fidèlement l'incertitude associée aux données, il est possible de se reposer sur l'idée que l'environnement va être perçu de nombreuses fois : en moyenne, le modèle s'approche de la réalité. Les points $3-D$ des images de stéréo-vision ne sont pas considérés comme *ponctuels* mais plutôt comme une *surface* (correspondant au pixel vidéo auquel ils sont associés). La résolution variable au sol des images est ainsi représentée. De plus, après avoir considéré un modèle d'erreur sur la stéréo-vision, il semble pertinent de considérer une incertitude $3-D$ proportionnelle à la surface au sol — c'est-à-dire approximativement proportionnelle au carré de la distance — des pixels vidéo. La figure 4.11 montre le résultat de la construction d'une carte à partir d'une image $3-D$ avec ou sans interpolation.

Cette technique présente l'avantage d'être extrêmement rapide, nécessite beaucoup moins de calculs que la méthode d'interpolation proposée par [Kweon 1992] et approxime en moyenne aussi bien l'environnement (ceci en particulier parce que les images $3-D$ du robot *Lama* sont denses).

On observe malgré tout que l'estimation de la précision est toujours trop pessimiste.

4.1.5 Une implémentation sur le robot *Lama*

Choix d'une résolution

La résolution des cartes détermine la précision à laquelle l'environnement va être modélisé. Plus cette précision est grande, plus le modèle sera sensible au bruit des données. Il est donc nécessaire de trouver un compromis entre une modélisation fine mais bruitée et une modélisation plus grossière.

Idéalement, on souhaiterait que les points proches du robots, perçus avec le plus de précision, soient le plus fidèlement représentés, tout en ne laissant pas de trous entre eux. La figure 4.12 montre une analyse effectuée à l'aide d'une image $3-D$ du robot *Lama* : on constate que la densité de ces images est telle qu'une taille de maille de l'ordre de 5 cm permet d'obtenir ce résultat.

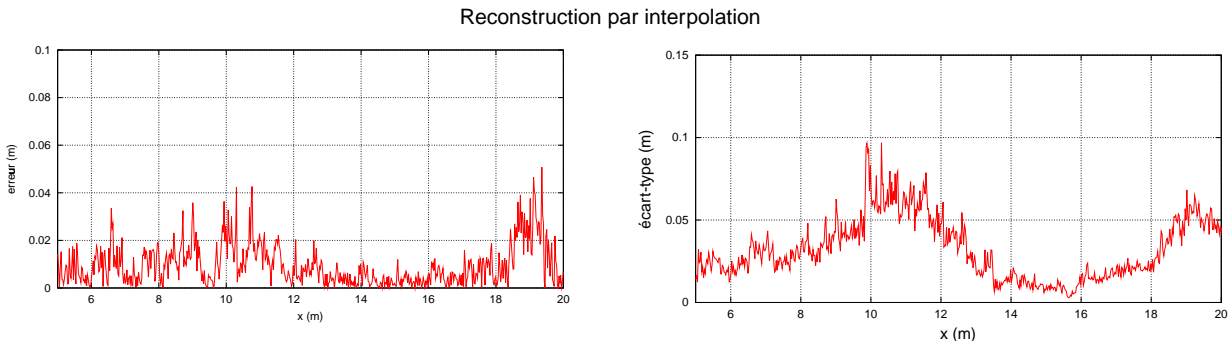


FIG. 4.10 – Reconstruction d'un profil de terrain à partir d'une interpolation. La technique de reconstruction utilisée ici effectue un moyennage de l'altitude dans chaque cellule de la carte d'élévation ; contrairement à la figure 4.9, les pixels perçus sont interpolés dans chaque cellule qu'ils recouvrent et un poids inversement proportionnel à la distance est appliqué.



FIG. 4.11 – Influence de l'interpolation pour le remplissage d'une carte d'élévation : la carte de gauche est construite sans interpolation.

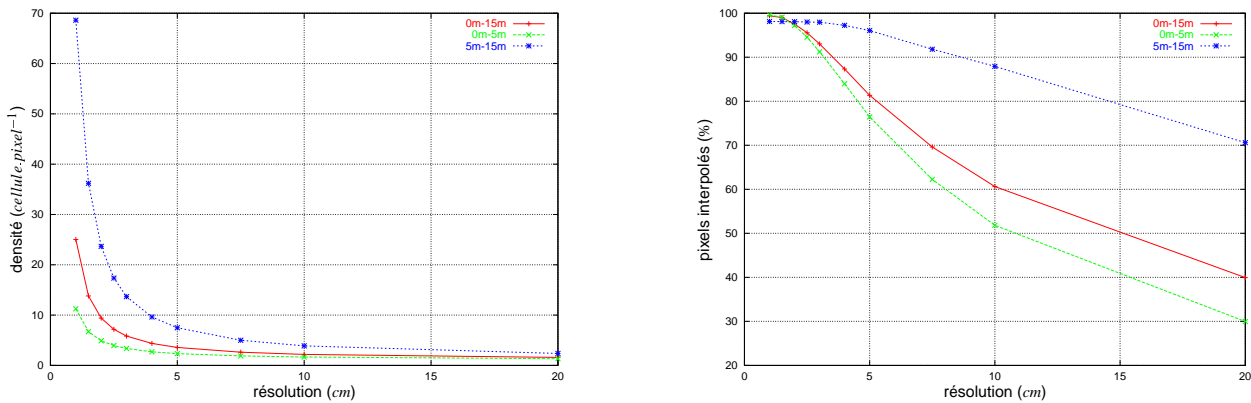


FIG. 4.12 – Densité des images de points 3-D. À gauche le nombre de cellules occupées en moyenne par un point 3-D, sur des tranches de 0 à 15m, de 0 à 5m et de 5 à 15m, en fonction de la taille de la maille de la carte. À droite le pourcentage de pixels nécessitant une interpolation, c'est à dire occupant plus d'une cellule du la carte, sur les mêmes tranches de distance. Les deux courbes ont été produites pour une même image de référence.

Filtrage des données incertaines

Outre le fait de bien modéliser incertitude et imprécision, il est nécessaire de détecter les données aberrantes. Celles-ci sont en effet souvent source de problèmes lorsqu'on tente de les fusionner : le fait que ces données soient aberrantes font qu'elles ne se comportent en général pas selon le modèle empirique d'erreur qui a pu être construit sur les données de façon globale et le problème de la fusion avec d'autres données est alors mal posé.

Dans le cas de la stéréo-corrélation comme sur le robot *Lama*, il est possible de profiter du modèle d'erreur sur le tracking de pixels (annexe ??). Les points que l'on sait être statistiquement mal corrélés, et donc très susceptibles d'être erronés, seront ainsi éliminés grâce à la fonction d'auto-corrélation.

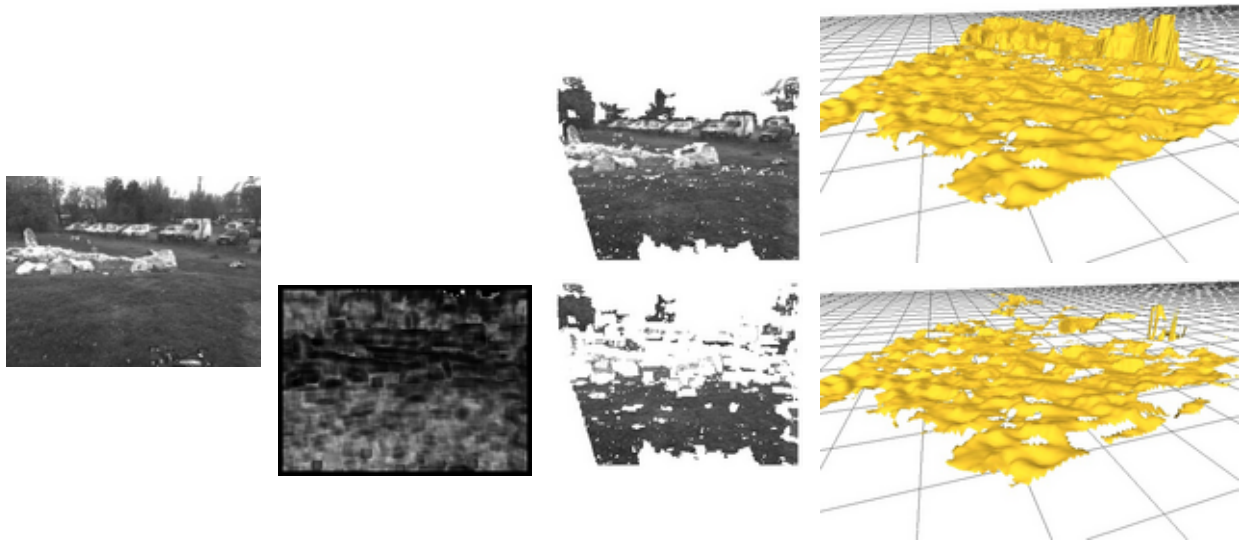


FIG. 4.13 – Filtrage des données les plus incertaines : le calcul d'une image d'auto-corrélation (deuxième image en partant de la gauche) permet la détection des points les plus mal corrélés par la stéréo-vision. Les troisième et quatrième images en partant de la gauche montrent le résultat du filtrage obtenu (en haut sans filtrage et en bas avec filtrage). Le volume des données ainsi perdues est faible et rapidement compensé par l'accumulation des perceptions.

Informations fusionnées dans le modèle

Si l'information tridimensionnelle est la partie la plus importante, il est néanmoins très intéressant d'inclure l'ensemble des perceptions à l'intérieur du modèle construit. En particulier, il est possible d'inclure les informations de luminance (ou de couleur) provenant des images vidéo ayant servi à construire les images $3-D$. D'autres informations peuvent être incluses pour enrichir le modèle : il peut s'agir d'informations de nature continue comme l'élévation ou la couleur, mais aussi d'information discrètes comme des amers ou des étiquettes associées à des lieux particuliers, détectée par d'autres méthodes. On obtient ainsi un modèle sémantiquement plus riche.

Enfin, dans le but d'obtenir une représentation plus fine, il est possible de stocker non seulement la valeur de la grandeur perçue, mais aussi sa dérivée au voisinage du point considéré. Dans le cas des altitudes, on peut ainsi calculer la pente du terrain en chaque point, à l'aide de l'ensemble de l'image $3-D$. Cette information pourrait permettre d'aider la fusion d'information ultérieures.

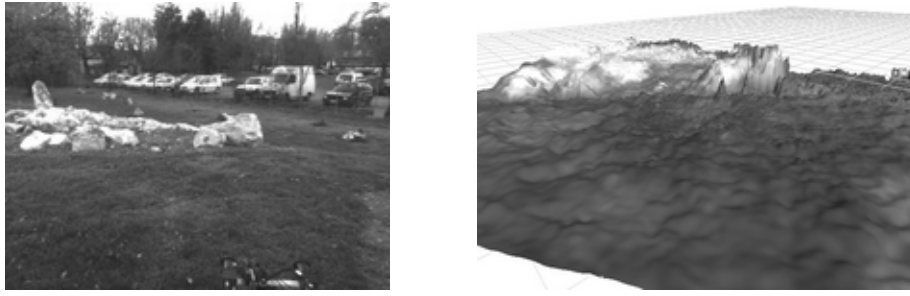


FIG. 4.14 – Carte d'élevation incluant les informations de luminance : l'image de droite représente une carte d'élevation construite à partir de 5 images vidéo prises autour du point de vue montré dans l'image de gauche.

4.2 Gestion du modèle à long terme

Le principal reproche que l'on peut adresser aux représentations sous forme de cartes d'élevation concerne la structure rigide dans laquelle les informations se trouvent fusionnées. Cette structure interdit, d'une part, de mémoriser une zone de taille arbitrairement grande et, d'autre part, de traiter les problèmes d'incohérence de la modélisation, liés à des erreurs de localisation ou à des modifications survenant dans l'environnement. Les méthodes de construction classiques « oublient » en effet que le robot perçoit l'environnement selon une trajectoire, et qu'il existe donc une chronologie reliant les perceptions. Conserver cette chronologie et représenter l'environnement *selon la trajectoire* réalisée par le robot permet de traiter les deux problèmes de la modélisation à long terme et du maintien de la cohérence.

La représentation idéale consiste à mémoriser chaque perception individuelle, pour les fusionner à la demande en fonction des dernières informations de position disponibles. Ainsi chaque position peut être corrigée et un modèle toujours cohérent est obtenu. Cette solution n'est cependant pas envisageable : le volume des données à mémoriser est rédhibitoire.

La structure que nous proposons permet d'obtenir un compromis entre volume des données et ré-actualisation du modèle : des corrections sur les différentes positions du robot sont possibles et le maintien la cohérence spatiale de la carte est réalisé. Grâce à quelques approximations, le modèle obtenu combine les avantages des deux approches « classique » et « idéale » : un volume de données égal (voire inférieur dans certains cas) à celui d'une carte d'élevation classique et une souplesse permettant la déformation locale du modèle, approchant ainsi le cas idéal où chacune des perceptions est stocké.

4.2.1 Carte locale

Afin de pouvoir explorer des zones de taille initialement inconnue, ou tout simplement des zones dont la taille ne permet pas le stockage d'une carte intégrale, le modèle construit est tout d'abord composé d'une carte locale.

Une carte d'élevation de taille fixe est donc attachée à un repère lié au robot et permet de fusionner les différentes perceptions locales (voir figure 4.15). La taille et la position de la carte par rapport au robot doivent être choisies pour être compatibles avec le champ de vue et perdre le moins possible de données lors des fusions.

L'intérêt de la carte locale, par rapport à la mémorisation de l'ensemble des perceptions, est double : la quantité de données à stocker est nettement réduite et le modèle ainsi construit est beaucoup plus robuste vis à vis d'erreurs sur quelques perceptions, grâce à la fusion de plusieurs informations dans chaque cellule de la carte.

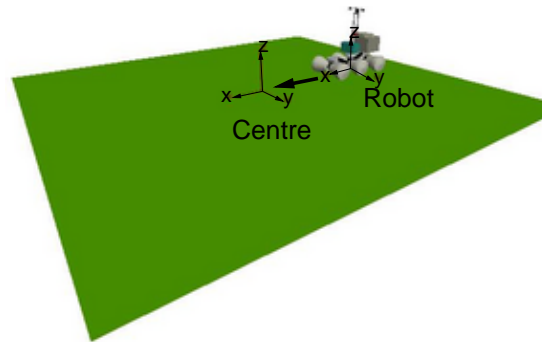


FIG. 4.15 – Position du modèle local par rapport au robot : le centre de la carte locale est rigidement lié à repère fixe par rapport au robot. La position relative robot / carte dépend de la configuration des caméras et de leur champ de vue.

Du point de vue de la cohérence spatiale, il est évident que les problèmes mentionnés en introduction de la section restent présents pour la construction de ce modèle, même s'il est local. Cependant, si l'on suppose que les méthodes de localisation présentes à bord du robot sont *suffisamment* performantes, la dérive en position — c'est-à-dire la distance entre la position estimée et la position véritable — croît alors d'une façon *approximativement* linéaire en fonction de l'abscisse curviligne le long de la trajectoire¹. Autrement dit, l'erreur sur la position estimée du robot ne grandit par de façon brutale en un court laps de temps et reste donc faible devant la résolution de la carte : le modèle est cohérent *localement*.

4.2.2 Modèle global

Au cours de longs déplacements, les erreurs de localisation tendent à grandir. Les méthodes de recalage ou d'estimation de position absolue doivent entrer en jeu pour réduire cette erreur, mais nous avons vu qu'elles ne peuvent pas fournir la position du robot à haute fréquence.

Le robot doit donc être capable de modéliser l'environnement alors que sa position est *potentiellement erronée*, dans l'attente de corrections ou de validation de la position de la part des algorithmes de localisation les plus lents. Les éventuelles corrections sur les positions relatives des perceptions locales doivent donc s'effectuer *rétrospectivement* et *le long* de la trajectoire réalisée. Une modélisation simpliste ne peut pas tolérer de telles corrections : lorsque plusieurs informations *incompatibles* ont été fusionnées, il est impossible de les dissocier. La structure que nous proposons permet ces modifications.

Tout d'abord, remarquons que le modèle local, attaché au robot, ne suffit pas pour mémoriser l'ensemble des données perçues : puisque la carte se déplace avec le robot, des portions précédemment explorées sont amenées à sortir de la carte locale, comme le montre la figure 4.16.

Étant donné que la carte locale n'est mise à jour *que* lorsqu'une perception est effectuée, et que ces perceptions sont *discrètes*, les zones amenées à sortir des limites du modèle local ont une surface non négligeable, correspondant approximativement au déplacement du robot entre deux perceptions. Dans le cas d'un déplacement le long d'une ligne droite parfaite, par exemple, la zone éliminée du modèle local a pour la largeur la largeur de la carte et pour hauteur la longueur du déplacement réalisé entre

¹Le chapitre précédent a montré que c'était vrai aussi bien pour l'odométrie que pour l'odométrie optique.

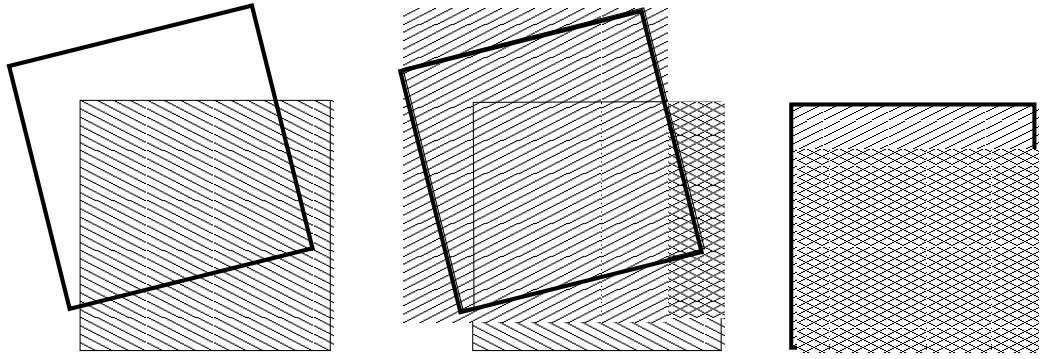


FIG. 4.16 – Mise à jour d'une carte d'élevation locale : lorsque qu'une nouvelle perception est effectuée (rectangle en gras, à gauche), celle-ci est mise en correspondance avec la dernière carte locale construite (rectangle fin, à gauche). Les anciennes élévations ainsi que la nouvelle perception sont fusionnées dans la nouvelle carte (au centre) et cette dernière devient le modèle courant. Les données situées à l'extérieur du nouveau modèle sont éliminées (à droite) et mémorisées à part.

deux prises de vue. Nous appellerons « sous-cartes » ces portions éliminées du modèle local.

Chaque *sous-carte* ainsi construite au cours du mouvement est mémorisée dans une liste et constitue le modèle proprement dit. La figure 4.17 schématise ce processus en ne représentant qu'un nombre réduit de sous-cartes : les déplacements du robot entre les perceptions sont grands et les sous-cartes schématisées ont une surface importante. Dans une situation réelle, un nombre de perception beaucoup plus conséquent crée des sous-cartes de taille beaucoup plus petites.

Le centre de chaque sous-carte est positionné dans l'environnement, grâce à la dernière position connue du robot lors de la création de la sous-carte. Ces portions du modèle sont par la suite *figées* et ne connaissent plus de mises à jour : on considère qu'elles représentent le résultat de la modélisation de l'environnement et qu'elles sont localement cohérentes. La seule modification autorisée est le déplacement, si la position est ré-estimée, et la déformation, comme l'explique le paragraphe suivant.

4.2.3 Déformation du modèle

Si aucune correction n'est apportée sur les positions des sous-cartes dans l'environnement, le modèle est directement exploitable : la simple concaténation des différentes sous-cartes présentes permet d'obtenir une carte globale.

En revanche, lorsque le robot ré-estime des positions, il est nécessaire de déplacer une ou plusieurs parties du modèle. Par exemple, la figure 4.17 illustre le cas d'une trajectoire en boucle, au cours de laquelle un *même* objet est perçu depuis la position R_a en A et depuis la position R_b en B . Une méthode de recalage peut détecter et reconnaître, à la deuxième perception, cet amer et produire une correction sur la position R_b ainsi que sur l'ensemble de la trajectoire reliant R_a à R_b .

La structure de liste de sous-cartes permet de déformer le modèle et prendre en compte ces corrections. Cependant, il ne suffit pas de modifier les positions individuelles des cartes : la notion de continuité, issue de la construction, doit être conservée et la *forme* des sous-cartes modifiée, à la manière d'une bande élastique déformable.

La figure 4.18 illustre le principe retenu pour le cas de deux sous-cartes consécutives : deux cartes R_a et R_b voient la position de leurs centres respectifs modifiée, selon la correction C_a pour la carte a et C_b pour la carte b . C_a et C_b sont *a priori* deux vecteurs de dimension 6, représentant un déplacement dans l'espace, mais puisque les différentes cartes sont horizontales et situées à une altitude constante (car l'élevation fait partie

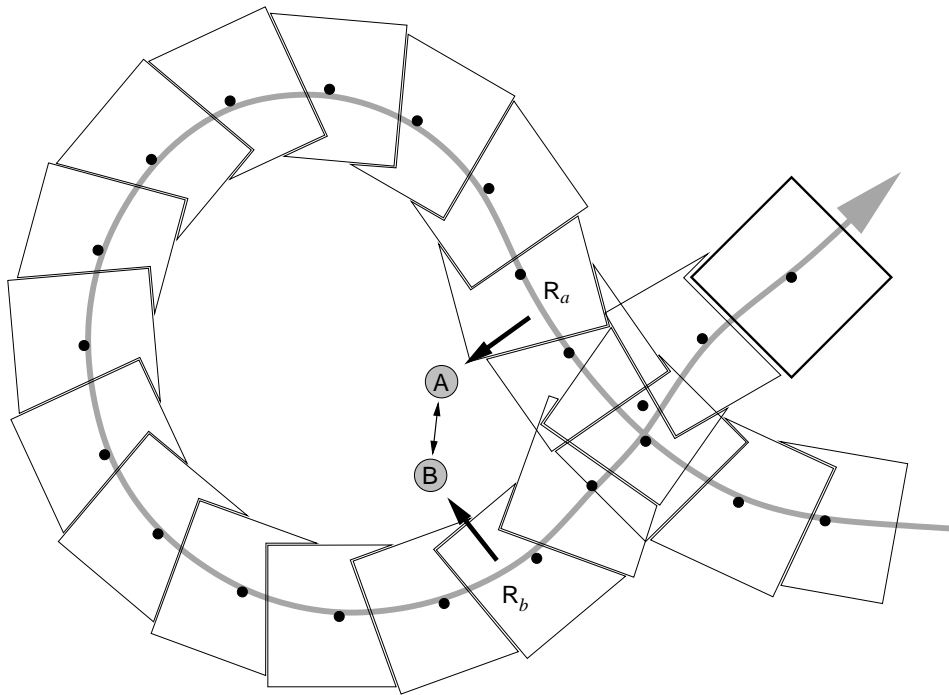


FIG. 4.17 – Représentation schématique du modèle de cartes d'élévations, après une trajectoire fictive comportant un nombre réduit de perceptions (les différentes cartes apparaissent disjointes pour plus de clarté). Les marques noires représentent le centre des différentes cartes, qui est positionné par rapport à l'environnement exploré. La carte dessinée en trait gras (en haut, à droite) est la dernière carte locale construite et n'est donc pas subdivisée. La structure ainsi obtenue permet de déformer l'ensemble du modèle lorsque des positions doivent être corrigées. On remarque par exemple, à l'endroit où la trajectoire boucle sur elle-même, que des zones communes mais perçues à des instants très différents ne sont pas fusionnées : la position exacte de leur intersection peut être ré-estimée à tout instant.

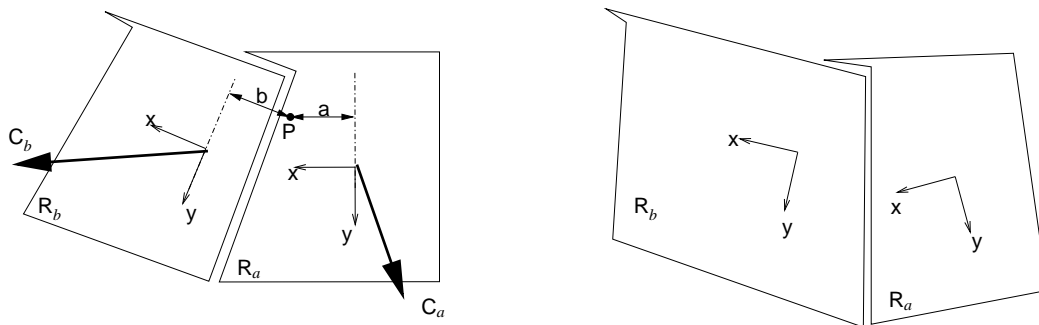


FIG. 4.18 – Déformation du modèle global : elle consiste à interpoler des corrections, notées C_a et C_b (6 paramètres), définies uniquement au centre des cartes, dans le plan 2-D formé par la surface explorée des cartes R_a et R_b . En chaque point P on calcule la correction $C = k.C_a + (1 - k).C_b$ telle que cette déformation soit au minimum continue pour conserver la cohérence du modèle. Ici, $k = b/(a + b)$ où a et b sont respectivement les distances du point P à l'axe y des repères des cartes a et b permet d'assurer cette propriété. La figure de droite montre le résultat de la déformation.

du modèle), le nombre de paramètres se réduit à 3. La déformation du modèle selon C_a et C_b doit tenir compte du fait que le robot a perçu l'environnement de a vers b d'une façon continue et avec une erreur de localisation dont la croissance était *faible* : une interpolation de C_a et C_b en chaque point du plan des sous-carte permet de prendre en compte ces contraintes. La figure 4.18 donne l'exemple le plus simple d'interpolation, où la distance de chaque point à l'axe y du repère de la carte permet une propagation continue des corrections. D'autres techniques d'interpolation sont possibles et doivent encore être étudiées.

4.3 Localisation sur des cartes d'élévation

L'intérêt de la méthode de modélisation proposée est sa capacité à intégrer des corrections sur la position ou la trajectoire. Nous avons donc tenté de développer plusieurs méthodes de localisation permettant de valider cette technique. Malheureusement, les résultats sont encore récents et n'ont pas pu être explorés suffisamment pour obtenir une méthode de localisation efficace sur ces modèles. Ces pistes paraissent néanmoins très prometteuses.

Deux approches différentes sont présentées. Une première méthode, intermédiaire entre estimation de mouvement et recalage, permet d'assurer la cohérence locale du modèle au cours des fusions des perceptions. Plusieurs autres techniques, très prospectives, de recalage basées sur des amers permettraient une localisation à plus long terme.

4.3.1 Corrections locales de la position

Un des aspects importants de la construction des modèles concerne la garantie de la cohérence, lors des fusions d'images $3-D$ locales. En effet, une fusion non contrôlée peut aboutir à des erreurs catastrophiques si la position des perceptions n'est pas suffisamment bonne. De telles erreurs se produisent parfois à bord du robot : une vibration ou un glissement, non mesuré, fait que quelques images sont parfois très mal positionnées. Il est indispensable de détecter ces situations pour éliminer les images ou, mieux, corriger leur position si l'on en est capable.

Nous avons donc exploré une méthode, intermédiaire entre estimation de mouvement et recalage, qui permet à la fois de contrôler ou améliorer la qualité du modèle construit et d'obtenir une estimation de position. Il s'agit d'une technique qui utilise les données $3-D$ sans construction de modèle d'objet : une image $3-D$ locale est positionnée dans le modèle précédemment construit grâce à la minimisation d'une distance entre image et modèle, en fonction des 6 paramètres de position. La distance donnant pour l'instant les meilleurs résultats semble être la somme des carrés des distances euclidiennes (SSD) entre chaque point de l'image $3-D$ et le modèle.

Afin d'optimiser les calculs et obtenir ainsi une méthode *utilisable*, plusieurs simplifications ont été faites et restent à évaluer. D'une part, seul un sous-ensemble des points des images sont utilisés pour le calcul de la distance à minimiser : il s'agit de l'étape la plus coûteuse en temps. D'autre part, l'algorithme de minimisation employé a été choisi pour la simplicité des calculs : il s'agit d'un simplex, qui ne nécessite pas l'évaluation des dérivées partielles de la fonction à minimiser. Enfin, une estimée initiale de la position de l'image, fournie par l'odométrie, a été employée et l'espace de recherche a été fortement restreint au voisinage de cette position (la suite de la section montrera l'influence de ce choix).

Cette méthode estime un déplacement, entre deux images, et l'erreur n'est donc pas bornée. De plus, elle est conçue pour être activée relativement fréquemment et à chaque

perception. Il s'agit donc d'une méthode d'estimation de mouvement. Cependant, une mémorisation des données est effectuée : les perceptions locales sont recalées grâce à un modèle agrégeant une partie des perceptions passées. Il s'agit donc également d'une méthode de recalage.

Les figures 4.19 et 4.20 montrent les résultats obtenus à l'heure actuelle en termes d'estimation de position, sur la série d'images acquises sur le robot *Lama* et utilisées pour qualifier l'odométrie optique.

La première observation que l'on peut faire au regard des courbes de la figure 4.19 est que la position n'est pas vraiment mauvaise jusqu'au milieu de la trajectoire, où une erreur angulaire importante a été commise. La dérive inexorable qui s'ensuit donne une idée pessimiste de la qualité de la position : la précision moyenne reste de l'ordre de celle de l'odométrie — rappelons que l'estimée initiale de la position a été fournie par l'odométrie, et que ce résultat n'est donc pas surprenant.

La figure 4.20 est plus intéressante, en particulier dans la zone située entre 30 et 40 m : là où l'erreur sur l'estimation de la distance parcourue, commise par l'odométrie, croît de façon assez brutale, l'erreur de la méthode de recalage sur les cartes d'élevation reste relativement constante. Pour cette zone particulière la méthode a donc été performante.

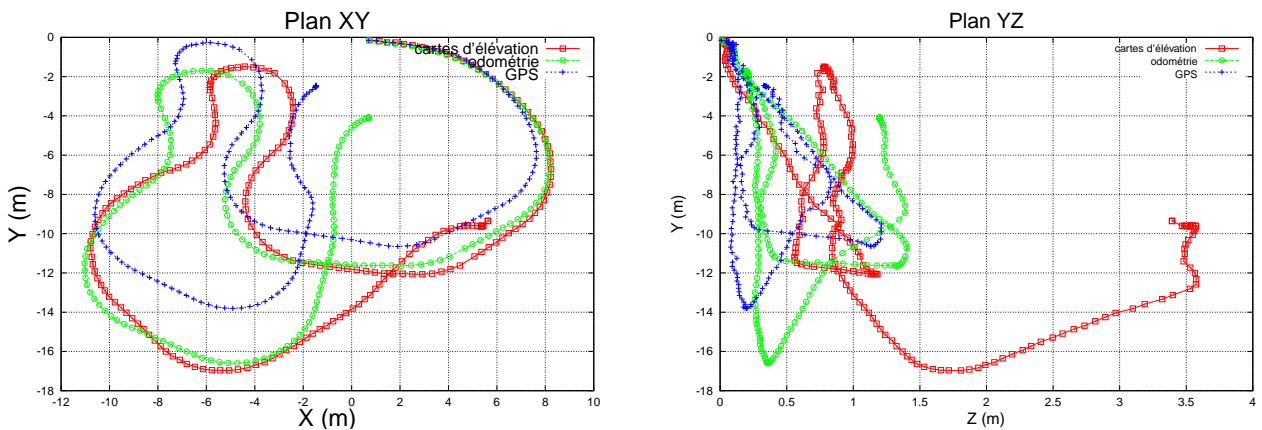


FIG. 4.19 – Trajectoire réelle mesurée par recalage sur cartes d'élevation, odométrie et GPS. Les deux graphiques montrent une vue de dessus (plan XY horizontal, à gauche) et de profil (plan YZ vertical, à droite) d'une trajectoire réalisée à la vitesse de 2 cm.s^{-1} sur une distance d'environ 70 m. La séquence d'images utilisée est la même que celle du chapitre 3 sur la figure 3.6.

Globalement, le résultat du point de vue de la position n'est donc pas très bon, même s'il est encourageant. De nombreux facteurs peuvent être en cause : d'une part les optimisations mentionnées ci-dessus doivent être évaluées et, en particulier, l'algorithme de minimisation ainsi que le choix de la distance pourraient être améliorés ; d'autre part, de nombreuses dimensions, permettant de connaître la position des caméras relativement au repère dans lequel l'odométrie calcule la position, ont été mesurées de façon approximative et aucune méthode de calibrage n'a été mise au point. Contrairement à l'odométrie optique, qui estime la position de la caméra et cumule la position dans ce repère, il y a pour cette méthode un effet de cumul sur des biais éventuels lors du « redressement » de l'image dans un repère vertical.

De façon paradoxale, on observe une augmentation de la qualité du modèle obtenu, notamment sur la précision en termes d'écart-type sur les altitudes, malgré le fait que

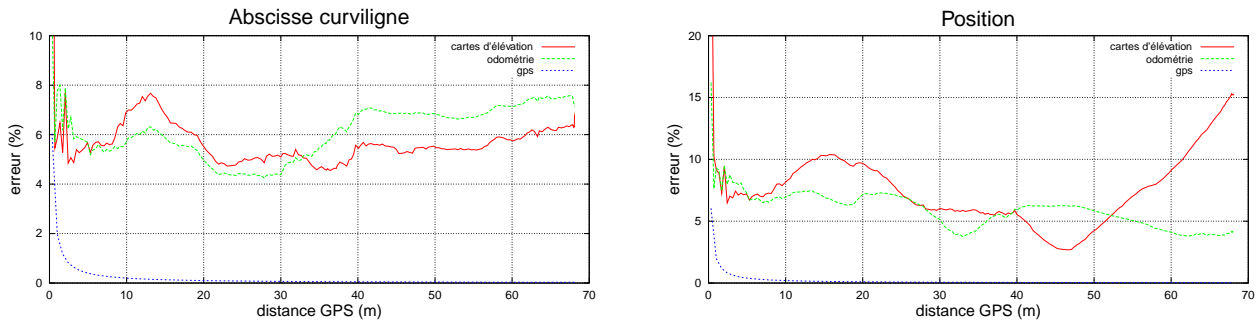


FIG. 4.20 – Comparaison des erreurs en translation entre recalage sur cartes d'élévation et odométrie. Sur le graphique de gauche est représentée l'erreur sur la mesure de la distance parcourue pour la trajectoire de la figure 4.19. Sur le graphique de droite on peut observer la distance relative entre la position mesurée par le GPS et les positions calculées par le recalage sur les cartes d'élévation et l'odométrie, pour cette même trajectoire. La courbe de précision du GPS est superposée, montrant que le bruit sur sa position est rapidement négligeable.

la position calculée soit moins bonne que celle, par exemple, de l'odométrie. Si l'on considère encore une fois qu'il est très probable qu'il y ait des *biais* sur les positions des caméras par rapport au robot, ce phénomène n'est pas surprenant : le modèle construit est *localement cohérent*, mais sa forme globale est faussée, déformée par la présence des biais. La structure déformable choisie prend donc toute sa signification.

4.3.2 Estimation de la position sur la base d'amers

Un amer est un élément stable de l'environnement que le robot est capable d'identifier et éventuellement reconnaître parmi l'ensemble des amers précédemment perçus. Pour permettre une bonne estimation de position, ces éléments doivent être facilement identifiables, géométriquement riches et surtout stables dans l'environnement, quel que soit le point de vue. La présence d'attributs géométrique est cependant facultative, si de nombreux amers sont utilisés : leurs positions relatives sont suffisantes pour effectuer des recalages.

Ce paragraphe présente quelques pistes que nous avons abordées. Certaines ont été abandonnées, mais d'autres semblent prometteuses et pourraient donner naissance à des méthodes de recalage sur amers à court terme. Deux problèmes ont été étudiés : la segmentation, permettant d'isoler les amers du reste de l'environnement et la modélisation, pour effectuer des recalages sur la base d'attributs géométriques.

Segmentation

Plusieurs contributions proposent des techniques de segmentation *3-D* permettant d'extraire des objets saillants de l'environnement [Moutarlier 1991, Betge-Brezetz 1995]. Cependant, les techniques de détection simples ne sont efficaces que dans des cas où l'environnement est composé d'objets épars sur un sol globalement plat ; si l'environnement devient trop accidenté, elles s'avèrent inopérantes. Afin de détecter des objets remarquables, nous avons proposé deux méthodes.

La première se base sur une discrétisation régulière des images *3-D* permettant de prendre en compte la géométrie projective du capteur [Mallet 1998]. Des contours (lignes polygonales) sont extraits sur la base de la densité des points qui se projettent verticalement dans les cellules de la discrétisation, permettant ainsi de détecter les zones les plus verticales de l'environnement (figure 4.21). Un recalage entre deux objets de ce type peut alors être effectué sur la base d'un attribut (par exemple de courbure) calculé le long du contour extrait. Cette méthode a fait l'objet d'une étude pour



FIG. 4.21 – Extraction d'amers sur la base de la densité des points 3-D : des contours polygonaux, correspondant aux zones les plus verticales de l'environnement, peuvent être extraits sur la seule base d'images de points 3-D et fournir la base d'un recalage.

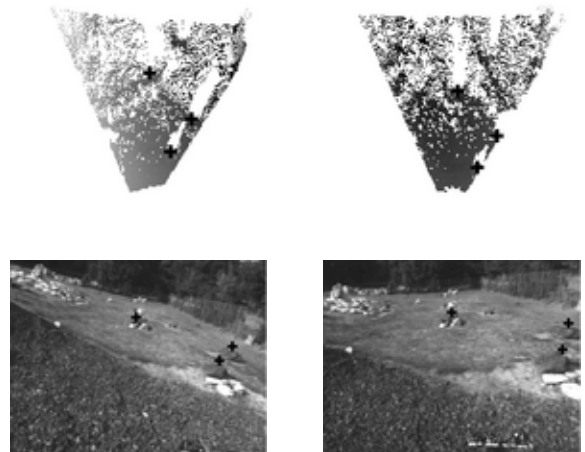


FIG. 4.22 – Extraction d'amers sur la base de cartes d'élévation : grâce au calcul d'un score de ressemblance entre un modèle (ici un paraboloïde) et les cartes d'élévation, des amers stables (les croix noires) peuvent être extraits. Ici, Trois mètres séparent les deux images et les amers sont situés à des distances allant de 3 à 10 mètres.

l'*Aérospatiale* [Lacroix 1997], mais présente l'inconvénient d'être très sensible à la qualité des images 3-D; elle n'a pas connu de nouveaux développements depuis.

Une autre méthode de segmentation plus prometteuse utilise les cartes d'élévation : le calcul d'un score de ressemblance, à différentes échelles, entre ces cartes et un modèle 3-D pré-défini (par exemple un paraboloïde) permet d'exhiber ainsi les zones possédant la forme particulière recherchée. Ainsi, la figure 4.22 montre le résultat de cette méthode sur deux images distantes de quelques mètres. Le résultat de cette détection pourrait être utilisé pour des méthodes abordant le problème de la localisation simultanée des amers et du robot (SLAM) comme par exemple [Marco 2000].

Modélisation

Une méthode développée dans [Mallet 1997] permet d'obtenir un recalage 3-D, sur les 6 paramètres de position, à partir d'une modélisation d'amers. Une hypothèse restrictive, considérant que l'environnement est globalement plat et parsemé d'obstacles, a permis de d'obtenir une méthode de segmentation pour séparer les obstacles du sol par un simple seuillage sur l'altitude des points 3-D.

Le recalage en position repose sur la modélisation des amers par un maillage déformable destiné à approximer au mieux un nuage de points 3-D représentant l'objet. Sur chaque sommet, un ou plusieurs attributs peuvent être calculés (sur la figure 4.23 la courbure locale a été utilisée). Le maillage est ensuite déformé sur une sphère pour s'affranchir de la géométrie de l'objet [Delingette 1993, Higuchi 1995].

La position est obtenue grâce à une technique de corrélation entre deux modèles de ce type, en fonction des deux degrés de liberté autorisés par le modèle sphérique. Des appariements de points 3-D entre deux vues d'un même objet sont réalisés et une estimation de position similaire à celle de l'odométrie optique fournit le déplacement entre les deux vues du même objet.

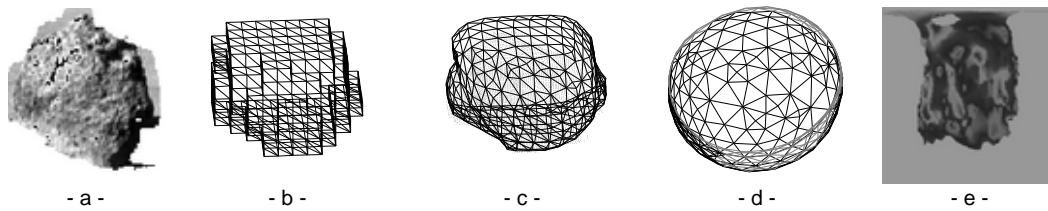


FIG. 4.23 – Processus de modélisation d'amers par maillages déformables : a) Image vidéo d'un obstacle segmenté. b) Maillage par marching cubes. c) Déformation du maillage pour approximer au mieux les points de données. d) Déformation sur une sphère (après calcul de l'attribut de courbure en chaque sommet). e) Modèle de l'amer, représenté en coordonnées polaires.

Cette méthode semblait donner des résultats utilisables, mais s'est avérée excessivement coûteuse en temps de calcul. De plus, les images $3-D$ issues du robot *Lama* semblaient beaucoup trop bruitées pour permettre le calcul d'un attribut géométrique stable. Son développement n'a donc pas été poursuivi.

4.4 Conclusions

Ce chapitre a abordé le problème de la construction de cartes d'élévation intégrant un modèle d'erreur réaliste sur les données $3-D$ et, dans une moindre mesure, le problème du recalage à partir de ces cartes.

L'évaluation de deux méthodes de construction probabilistes semble montrer qu'une approche de type Dempster-Shafer permet une modélisation à la fois plus intuitive et plus réaliste des incertitudes et des erreurs des capteurs $3-D$ de type stéréo-vision. Cette analyse a permis d'obtenir une méthode de validation des techniques plus classiques de construction et devrait guider les développements futurs, permettant ainsi d'obtenir une construction la plus réaliste possible. Néanmoins, la structuration des données, adaptée à la navigation d'un robot réel, permet d'ores et déjà de disposer d'un outil de travail qui permettra d'une part l'étude de fonctions de recalage et, d'autre part, la construction de modèles cohérents à grande échelle.

Le problème de la localisation a été abordé, mais peu de résultats définitifs ont, pour l'instant, été produits. Parmi les objectifs futurs, l'utilisation effective des imprécisions des données $3-D$ est prioritaire : elle permettra de développer des méthodes de localisation capables de qualifier le résultat de leurs estimations.

Enfin, la présence de cartes à grande échelle permettra d'aborder le problème de la localisation absolue, à partir de données initiales, et fournira un moyen de positionner le robot dans son environnement.

V

Intégration logicielle

Les chapitres précédents ont abordé la localisation du point de vue méthodologique : différentes solutions algorithmiques ont été proposées et évaluées dans des situations réelles, à bord d'un robot. Nous avons malheureusement pu constater qu'aucune de ces solutions n'est parfaite, ni générique, et que chacune peut faillir à son rôle d'estimateur de position dans certaines circonstances. Ces échecs sont, la plupart du temps, très difficiles à détecter si l'on ne se place pas dans un contexte plus large que celui de la seule fonctionnalité défaillante.

Obtenir un plus grand degré de robustesse doit, selon nous, passer par l'utilisation d'un vaste ensemble de méthodes possédant des caractéristiques différentes et permettant, entre autres, de se qualifier mutuellement. La clé du problème de la localisation — et, d'une façon générale, de l'autonomie — réside dans l'intégration et l'utilisation *coordonnée* d'un ensemble de fonctionnalités *concurrentes*.

La mise en œuvre d'un ensemble de tâches aussi variées que celles nécessaires à la navigation pose une véritable problématique scientifique concernant la meilleure manière d'organiser et de structurer les processus, la façon la plus efficace de partager les ressources que sont les différents capteurs ou encore les outils de communication pour échanger ces données. Le LAAS a, dans ce but, proposé un formalisme d'intégration au sein duquel nous avons abordé le développement d'un système de localisation générique.

Ce chapitre, assez prospectif dans son ensemble, débute par la description succincte de l'architecture logicielle définie par le LAAS. En particulier, les différents points effectivement instanciés à bord du robot *Lama* sont décrits. La seconde section présente l'intégration des algorithmes de localisation proprement dits et exhibe les différents éléments nouveaux et génériques qui ont permis de réaliser une démonstration utilisant les trois fonctionnalités développées dans ce document ; le contenu de cette démonstration est détaillé dans la section 5.3. Enfin, une réflexion plus générale conclut le chapitre et présente la façon dont ces travaux pourraient se poursuivre.

5.1 Une architecture pour l'autonomie

L'organisation et l'intégration des fonctionnalités au sein d'un système est un point essentiel : quantité de travaux sont menés dans ce sens dans la communauté robotique [Simmons 1994, Borelly 1998]. Le LAAS travaille également sur la définition et le développement d'une architecture logicielle pour l'autonomie des robots [Alami 1998a] et ces travaux ont permis la réalisation de nombreuses démonstrations de robotique [Alami 1998b, Alami 2000].

Cette section présente l'architecture LAAS¹ : l'objectif n'est pas ici d'en faire une description rigoureuse et exhaustive, mais plutôt de la résumer afin d'exhiber les points plus spécifiquement abordés pour l'intégration d'algorithmes de localisation. En particulier, le paragraphe 5.1.2 explique comment sont assurés les transferts de données et souligne différents problèmes que nous avons pu rencontrer. Le paragraphe 5.1.3 présente quant à lui l'état actuel du robot *Lama* du point de vue des parties de l'architecture effectivement instanciées.

5.1.1 Hiérarchie et modularité

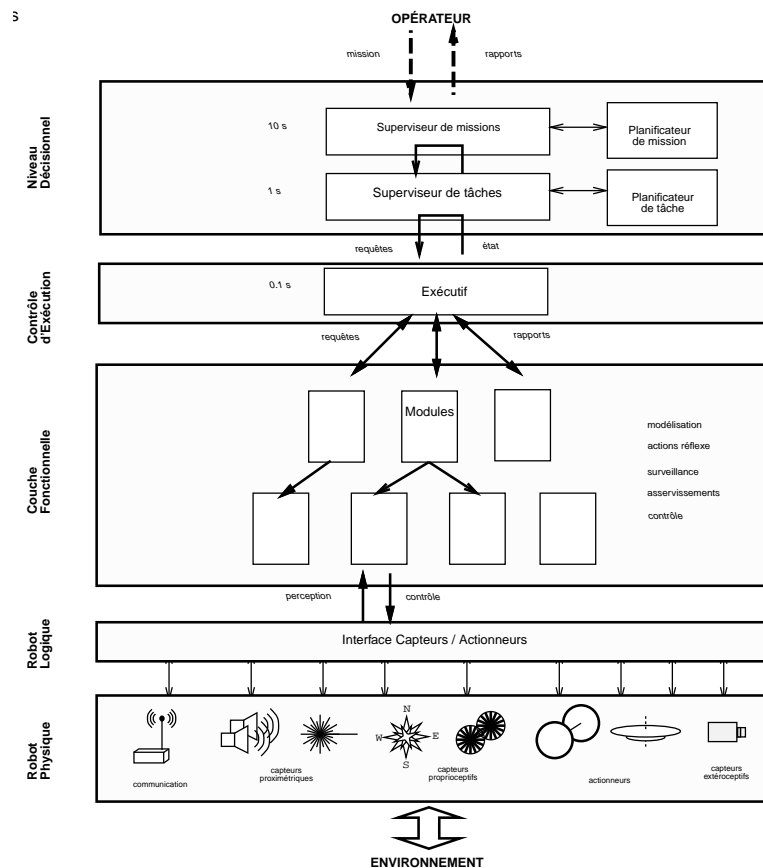


FIG. 5.1 – Architecture logicielle LAAS.

La principale caractéristique de la stratégie d'intégration utilisée à bord des ro-

¹LAAS Architecture for Autonomous Systems. La récursivité de l'acronyme mettrait-elle en avant l'aspect modulaire et hiérarchique de l'architecture, au sein de laquelle l'autonomie doit être présente à tous les niveaux ?

bots du LAAS consiste à définir un système hiérarchique et modulaire permettant de satisfaire les contraintes suivantes :

Programmabilité Les robots ne doivent pas être dédiés à des tâches ou des environnements uniques ni conçus de façon *ad hoc* et figée. Ils doivent être capables de mener à bien différentes missions, décrites à un niveau abstraction élevé.

Adaptation Les robots doivent mener à bien leurs missions en modifiant et affinant leur comportement, en fonction d'un objectif et du contexte courant.

Réactivité Les robots doivent traiter des événements dans des limites de temps compatibles avec la réalisation de leurs objectifs tout en assurant leur propre sécurité.

Cohérence Les réactions des robots doivent être guidées par les objectifs à atteindre et leur comportement doit être le résultat d'une suite logique d'actions permettant d'atteindre ces objectifs.

Robustesse L'organisation des fonctionnalités doit permettre aux robots d'exploiter les redondances fonctionnelles, détecter les situations anormales et réagir de façon cohérente vis à vis de celles-ci.

Quatre couches logicielles coexistent, reliant l'opérateur à la plateforme matérielle et traduisant les missions abstraites en un enchaînement d'actions élémentaires :

Le robot logique permet de rendre les logiciels des couches supérieures aussi indépendants que possible du robot physique sous-jacent — et donc *portables* d'un robot à l'autre.

Le niveau fonctionnel inclut l'ensemble des capacités d'action et de perception données au robot. Ces fonctions de traitement sont encapsulées dans des entités contrôlables et communicantes appelées « *modules* » [Fleury 1996]. Un module peut lire des données exportées par d'autres modules et exporter lui-même ses propres structures de données, résultats de ses traitements.

L'organisation des modules n'est pas *figée* et leur agencement dépend des missions que l'on souhaite confier au robot. Les interactions entre modules sont elles aussi dynamiques et dépendent de l'environnement ou de la tâche en cours d'exécution. Cette stratégie permet d'obtenir un comportement *flexible* et *reconfigurable*.

Les modules sont interfacés avec les capteurs et actionneurs à travers la couche logicielle du *robot logique*.

L'exécutif contrôle et coordonne l'exécution des fonctions distribuées dans les modules, en fonction des besoins de la tâche courante. Il agit comme un pivot entre le niveau décisionnel (présenté ci-après) et la couche fonctionnelle, créant le lien entre les *décisions* et les *actions*.

L'exécutif est un système purement *réactif*, dépourvu de capacités de planification. Il reçoit du niveau décisionnel une séquence d'actions à exécuter puis choisit, paramètre et synchronise de façon dynamique les fonctions adéquates, présentes dans la couche fonctionnelle. Il gère à la fois le flux de contrôle et le flux de données et résout les conflits potentiels grâce à l'usage de priorités.

Le niveau décisionnel comprend l'ensemble des capacités de planification de tâches et de supervision de mission présentes à bord du robot. Il réagit aux événements produits par l'exécutif. Cette couche peut être divisée en plusieurs entités basées sur la

même conception mais utilisant différentes représentations ou différents outils algorithmiques. Son instanciation dépend avant tout du contexte considéré et des problèmes à traiter.

5.1.2 Flux de contrôle, flux de données

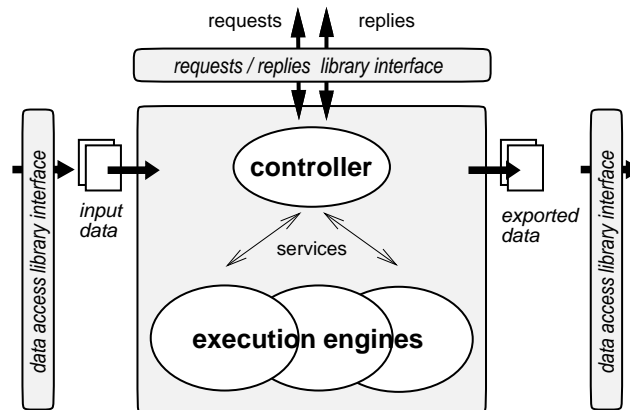


FIG. 5.2 – Description des modules.

Afin de mener à bien les tâches et activités demandées par le niveau « exécutif », les modules doivent à la fois communiquer entre eux et avec les couches logicielles formées par l'exécutif et le robot logique. Ces communications se divisent en deux catégories principales : le flux de *contrôle* et le flux de *données*.

Flux de contrôle

Ce flux est actuellement composé de l'ensemble des requêtes et répliques échangées entre les modules ou avec l'exécutif. Les requêtes permettent le déclenchement d'activités élémentaires contenues au sein des modules, tandis que les répliques sont destinées à acquitter les requêtes.

On peut remarquer que dans l'architecture LAAS, aucun formalisme n'est clairement défini quant à la structure de ces requêtes — bien qu'une implémentation de celles-ci existe au sein du logiciel GenoM [Fleury 1996].

L'implémentation actuelle des requêtes permet à un module (ou à l'exécutif) de s'adresser à un autre module via le nom de ce dernier. Ce mécanisme est très efficace lorsque la couche fonctionnelle est bien structurée ; il s'avère néanmoins pénalisant lorsque l'ensemble des modules disponibles n'est pas connu à l'avance ou encore que *plusieurs* modules pourraient répondre au besoin exprimé par une requête.

Ainsi, il serait certainement bénéfique du point de vue de la *programmabilité* et de la *robustesse* de définir une notion de *type de service*, permettant à un module d'émettre une requête pour obtenir un service sans s'adresser de façon nominative à un module en particulier. Par exemple, si un module fournissant normalement le service demandé ne peut répondre (pour cause de défaillance ou de surcharge), un *autre* module, capable de fournir ce même service, peut ainsi prendre le relais. De plus, un mécanisme de communication général pourrait ainsi être mis en place, de façon indépendante du robot (ainsi, une image *3-D* peut être fournie par un module pilotant un banc stéréo ou

une caméra 3-D sans modifier le système). Ce mécanisme devrait vraisemblablement passer par l'exécutif, chargé de la gestion des ressources formées par l'ensemble des *services* disponibles.

L'absence de la notion d'évènement au sein de la couche fonctionnelle est également remarquable. En effet, si les requêtes permettent d'exprimer un *besoin*, il n'existe actuellement pas de moyen d'exprimer la *présence* d'une information nouvelle. De tels *évènements* pourraient permettre de mettre en place des actions *réflexes*² de façon très simple et, encore une fois, *programmable*. Des considérations sur la *réactivité* d'un tel système doivent toutefois être considérées.

Flux de données

Le flux de données permet, comme son nom l'indique, d'échanger des données entre les modules, les capteurs, ou l'exécutif. Il est actuellement réalisé grâce à l'utilisation de structures de données nommées *posters*.

Les posters sont des structures de données *partagées, statiques*³ et accessibles de façon *globale* et *nominative*. Chaque poster est la propriété d'un seul module, possédant seul le droit d'écriture. D'autres modules peuvent lire ces données, mais les lectures concurrentes sont impossibles (*i.e.* une seule lecture est autorisée à chaque instant).

Ici encore, aucun mécanisme n'est fourni pour la définition du flux de données : comme pour les requêtes, les posters sont accédés de façon nominative et l'obtention de structures de données passe donc par la connaissance de l'ensemble des posters présents dans le système.

De plus, il est de la responsabilité de chaque client de s'assurer que les données qu'il souhaite lire *correspondent* effectivement à ce qu'il attend, et rien ne permet une *définition commune* des structures échangées : un module chargé de la production d'un certain type de donnée est *également* chargé de la définition de la *structure* de celle-ci. Il en résulte que les échanges ont actuellement tendance à être décrits de façon *statique* à l'intérieur des modules, compromettant la *modularité* du système. Le développement de fonctionnalités destinées à échanger des informations doit donc se faire de façon *concertée* et ne résulte pas d'un comportement *automatique* du système. Dans la pratique, il s'en suit une forte incompatibilité entre des modules n'ayant pas été explicitement prévus pour communiquer ensemble, ainsi qu'une *duplication* excessive des types de données circulants.

La mise en place d'un système de typage des communications (à mettre en parallèle avec la notion de *type de service* présentée plus haut) pourrait permettre, en premier lieu, de valider de façon automatique les transferts de données effectués. Ensuite, la définition du flux de données au niveau de *l'exécutif* pourrait être réalisée de façon beaucoup plus souple et/ou automatisable : un module s'attendant à recevoir, par exemple, une image 3-D pourrait être connecté de façon immédiate avec un autre module produisant cette même donnée.

La limitation des accès en lecture, ou la nécessité d'avoir *un seul* propriétaire par poster, sont également pénalisantes. Par exemple, un module accédant en lecture à une structure de donnée volumineuse peut *bloquer* l'ensemble du système, *y compris* le propriétaire du poster, pendant toute la durée de son accès. Une mauvaise programmation peut donc avoir des conséquences catastrophiques pour le robot.

²« réflexe » ne signifiant ici pas *forcément* « dépourvu de réflexion ».

³Leur taille est fixe et ils sont créés une unique fois, pour toute la durée de la mission.

Un partage efficace devrait permettre d'optimiser les accès simultanés aux données en proposant, par exemple, un mécanisme de recopie automatique en cas de conflits. Selon ce principe, et pour être efficace, la création des posters doit être dynamique. C'est un point délicat puisque la garantie de fonctionnement du système est perdue (il est en effet possible de consommer toute la mémoire disponible et d'empêcher la création d'un nouveau poster). Il est donc indispensable d'installer, pour chaque module, un comportement approprié en cas d'échec lors d'une tentative d'écriture dans un poster.

Une autre limitation que l'on peut évoquer concerne le caractère éphémère et instantané des données produites dans les posters. L'espace réservé à un module pour la production de ses données est statique, et ne souffre actuellement aucune modification dans le temps (en termes de taille ou de localisation physique). Le système ne possède donc aucune notion de *mémoire* et une donnée produite est alors nécessairement consommée, ou perdue (remplacée par la version plus récente de celle-ci). Il n'est pas possible, de façon *explicite*, de conserver une donnée pour un traitement ultérieur.

Enfin, la section suivante présentant l'intégration d'algorithmes de localisation montrera qu'il existe un besoin d'association entre données (la section suivante nomme ce besoin « étiquetage »). Cette association permet de mémoriser le *contexte* de production des données et permet aux différents algorithmes de disposer de l'ensemble des informations nécessaires à leurs traitements. Ainsi, on remarque par exemple que les images vidéo ayant servi à la construction d'une image *3-D* par stéréo-vision sont *quasi-systématiquement* utilisées simultanément. De même, la *date* de production des données est fondamentale dès que l'on met en œuvres des algorithmes dont le temps d'exécution est long.

Pour réaliser ces associations de façon générique, un poster doit donc pouvoir se comporter comme un client vis à vis d'un autre poster, c'est à dire avoir la possibilité de référencer ou déréférencer d'autres données. Bien sûr, le poster n'étant qu'une donnée inerte, ce mécanisme devra être mis en place par l'intermédiaire d'une action au niveau des modules, de l'exécutif ou du superviseur. Mais l'association devrait être *indépendante* du client l'ayant créée : l'information doit donc être stockée au niveau de la structure même du poster.

5.1.3 Cas particulier du robot *Lama*

L'ensemble des logiciels présents à bord de *Lama* sont développés selon le formalisme présenté dans la figure 5.1. À l'heure actuelle, cependant, les fonctionnalités présentes ne requièrent pas de capacités évoluées de planification au niveau tâche et les superviseurs proposant ces possibilités n'ont donc pas été employés.

L'ensemble des démonstrations réalisées étaient composées de boucles figées, effectuées pour la plupart des traitements en boucle répétitives, à la fréquence la plus élevée permise par les temps de traitements. La définition et l'exécution de ces boucles est normalement en charge de l'exécutif, selon la terminologie présentée dans le paragraphe 5.1.1.

Afin de pouvoir programmer et tester les différentes fonctionnalités, ainsi que l'assemblage de celles-ci, nous avons développé un logiciel générique permettant de s'interfacer avec les modules de la couche fonctionnelle et de déclencher l'exécution d'activités au sein de celle-ci. L'ensemble se présente sous la forme d'un serveur, chargé de la communication avec les modules, et d'un nombre quelconque de clients, émettant des requêtes correspondants aux services offerts par les modules présents et écrites dans le langage Tcl.

Ce logiciel a permis le développement de l'ensemble des travaux présenté dans ce document. Il permet notamment l'écriture très simple d'outils de visualisation (grâce au paquetage Tk) des données présentes à bord.

5.2 Intégration d'algorithmes de localisation

L'utilisation de différents algorithmes de localisation nécessite la définition d'un contexte dans lequel ces algorithmes vont s'intégrer. Afin d'aborder ce problème de façon générique, configurable et adaptative nous avons proposé un module particulier, chargé de la collecte des différentes positions produites et de leur fusion en une position unique [Lacroix 2000c, Lacroix 2001c]. Ce module permet de répondre à quatre besoins : une gestion centralisée de la géométrie et de la configuration des capteurs (paragraphe 5.2.1), une gestion des contextes d'acquisition des données (paragraphe 5.2.2), une gestion cohérente des informations (paragraphe 5.2.3) et enfin une fusion et une détection des fautes éventuellement commises par les divers estimateurs (paragraphe 5.2.4).

5.2.1 Graphe géométrique du robot

Tous les capteurs utilisés à bord d'un robot sont spatialement distribués sur le châssis. Lors de l'assemblage de diverses fonctionnalités de perception, l'échange de données passe inévitablement par l'utilisation des positions relatives et des configurations courantes de ces capteurs. La géométrie du robot est donc un aspect important qu'il est nécessaire de considérer.

Dans un but de *réutilisabilité* et de *généricité*, il n'est pas judicieux de résoudre ces problèmes de façon *ad-hoc* : distribuer les informations et calculs de nature géométrique dans les divers modules aboutit non seulement à la duplication et la dispersion de données mais rend également les modules *dépendants* du robot sur lequel ils fonctionnent. Il apparaît donc souhaitable d'utiliser un système centralisé dédié à la gestion des aspects géométriques.

L'objectif d'un tel système est de parvenir à inclure *toutes* les informations dont les divers modules peuvent avoir besoin, de regrouper cette information dans une structure générique et d'obtenir ensuite une description abstraite du robot physique sous-jacent. Les modules utilisant cette information sont ainsi rendus indépendants du robot.



FIG. 5.3 – Robots ayant des configurations géométriques particulièrement différentes (de gauche à droite, Hilare IIbis, Lama et Obelix) : leur point commun est qu'ils peuvent tous se décrire grâce à un graphe où les nœuds représentent des repères particuliers (capteurs, actionneurs ou articulations). Les branches connectant les nœuds correspondent alors soit à une partie rigide du châssis, soit à une configuration d'un actionneur.

La figure 5.3, sur laquelle sont représentés quelques robots, montre que la géométrie peut varier énormément d'une plateforme à l'autre. Ainsi, l'utilisation d'une image issue

d'une caméra nécessitera, en fonction du robot, l'accès à des ressources très différentes pour obtenir sa position relativement au robot ou à un repère fixe.

Pour résoudre ces problèmes, nous avons proposé un module définissant un modèle géométrique du robot, qui peut ensuite être utilisé par les autres modules de façon *générique* et transparente. Ce module est donc la seule partie dépendante du robot. Les aspects techniques liés à la gestion du modèle et aux structures de données adoptées sont présentés en annexe B.

Dans ce modèle, la géométrie d'un robot est décrite par un graphe dont les noeuds sont des repères particuliers et les branches des changements de repère reliant ces noeuds. Les liens entre les repères sont soit *rigides*, soit *mobiles*. Les liens rigides ne peuvent pas être modifiés au cours de l'exécution et correspondent normalement à une partie mécanique du châssis du robot. Les liens mobiles sont par contre dépendant de la configuration courante d'un actionneur ou d'une partie du châssis. Les liens mobiles sont mis à jour périodiquement, à une fréquence pré-définie. Ces mises à jour se font grâce à la définition de fonctions, associées au graphe, qui sont chargées d'obtenir les paramètres pertinents et de calculer la configuration courante du lien et qui sont bien-sûr dépendantes du robot (l'annexe B décrit le cas du robot *Lama*).

Un repère particulier, dit « principal », est défini : il permet d'homogénéiser les transferts de données et chaque module chargé de l'acquisition de données métriques (stéréo-vision, par exemple) peut ainsi produire et exprimer ses données dans ce repère.

Ce module est situé, au sein de l'architecture, dans la couche fonctionnelle. Les données exportées (voir figure 5.4) sont une liste de repères, utilisables par les autres modules. Ces repères correspondent le plus souvent à des capteurs mais ils peuvent également s'avérer être un moyen efficace de séparer des liens qui auraient été plus complexes à définir autrement. La position courante de chaque repère est exprimée par rapport au repère principal qui est lui-même positionné dans un repère fixe de l'environnement grâce à la position du robot (cette position est le résultat de la fusion des différents estimateurs présents, voir paragraphes 5.2.3 et suivants). Enfin, l'ensemble des valeurs associées à ces données sont datées.

Date
Main to Base Main to Origin
Camera (top, left) to Main
Camera (top, right) to Main
...
GPS to Main

FIG. 5.4 – Exemple de données exportées par le module « Géométrie » : l'ensemble des repères définis dans le modèle géométrique sont exportés et calculés par rapport à un repère particulier, appelé « Main ». Ce dernier repère est positionné dans l'environnement et permet ainsi de connaître la configuration complète du robot.

5.2.2 Étiquetage et contexte d'acquisition des données

Le contexte dans lequel les données utilisées par les algorithmes de localisation ont été acquises est primordial. La configuration des capteurs doit être connue, ainsi que

les dates d'acquisition des données, pour permettre un calcul cohérent de la position du robot.

Le modèle géométrique, présenté dans la section précédente, a été défini dans ce but et une stratégie d'acquisition a pu être mise en place : tous les modules d'acquisition lisent, dans le modèle géométrique, la configuration courante du capteur auquel ils sont rattachés, ainsi que la date à laquelle les données sont récupérées et associent ces données aux perceptions proprement dites.

Cette stratégie est appelée « étiquetage », et fournit deux propriétés importantes :

- Les modules d'acquisition ne connaissent que le nom d'un repère auquel ils sont rattachés, et sont ainsi rendus indépendant du robot sur lequel ils s'exécutent.
- Les clients des modules d'acquisition n'ont *pas besoin* de connaître la provenance des données qu'ils manipulent étant donné que toutes les informations nécessaires sont contenues dans les « étiquettes » associées à toutes les données.

Les données contenues dans les « étiquettes » sont au nombre de quatre (figure 5.5). Le premier champ est la date à laquelle les informations ont été exportées. Les deux champs suivants sont des positions : la transformation du repère *Main* vers le repère *Origin* est la position *absolue* du robot, relative à un repère fixe de l'environnement (voir paragraphe suivant). Ce repère fixe est généralement la position du robot au début de la mission mais il peut être positionné n'importe où par un utilisateur. La transformation du repère *Main* vers le repère *Base* est une position particulière, dont l'intérêt est décrit dans le paragraphe suivant.

Date
Main to Base Main to Origin
Frame to Main

FIG. 5.5 – Données utilisées pour l'étiquetage : actuellement au nombre de quatre, ces données permettent de fournir l'ensemble des informations nécessaires à la localisation. On y trouve la configuration du capteur par rapport au repère principal du robot, ainsi que la position de ce dernier repère dans l'environnement. La date permet d'assurer la cohérence de la position calculée.

5.2.3 Gestion centralisée des positions

Nous avons montré qu'il est nécessaire de disposer de plusieurs méthodes de localisation, chacune répondant à des besoins différents. La présence de ces différentes méthodes induit donc la présence de *plusieurs* estimées de positions au sein du système, chacune correspondant à la même position du robot. Un nombre *a priori* inconnu d'estimateurs fournissent donc des résultats de façon *asynchrone* et avec un retard égal au temps de calcul nécessaire pour estimer les positions. Bien évidemment, les nombreux autres algorithmes utilisant la position du robot doivent pouvoir accéder à une position *unique*, représentant la meilleure estimation courante.

Afin de pouvoir gérer cette situation, nous avons proposé un schéma d'intégration dans lequel un module particulier est chargé du regroupement des différents estimateurs de position présents dans le système. Ce module est l'endroit où les différentes positions calculées sont centralisées, fusionnées et rendues disponibles pour les modules qui souhaitent les utiliser.

Actuellement, le rôle majeur de ce module est de maintenir une cohérence temporelle entre les estimations de position, compte-tenu des retards dans la production de celles-

ci. Il permet également d'insérer à la demande de nouveaux estimateurs, de manière transparente et automatique pour le reste des fonctionnalités présentes.

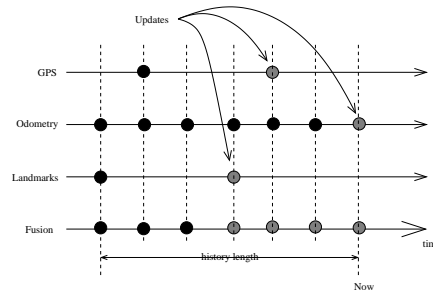


FIG. 5.6 – Gestion temporelle de la production des positions. Les différents estimateurs présentés ici ne sont que quelques exemples permettant d'illustrer la structure choisie pour gérer la production asynchrone de positions avec des retards. Les marques noires indiquent les positions précédemment stockées et marques grises les nouvelles positions produites à l'instant marqué « présent ».

La stratégie de centralisation des positions est pour l'instant élémentaire. À tout instant, un estimateur de position peut se déclarer comme producteur d'une position auprès du module de gestion des positions. Cette opération n'est réalisée qu'une fois, pour chaque estimateur. Par la suite, le module centralisateur vérifie périodiquement si un ou plusieurs producteurs ont calculé une nouvelle position, et récupère celle-ci le cas échéant.

Un diagramme temporel est associé à chaque estimateur de position : il permet de mémoriser un historique pour chaque producteur et autorise ainsi le stockage des positions produites (voir figure 5.6). La stratégie d'étiquetage permet de connaître la date pour laquelle les positions ont été calculées et celles-ci sont placées dans le diagramme correspondant à l'estimateur de position ayant fourni la nouvelle donnée.

Deux classes d'estimateurs sont distinguées : les estimateurs absolus et les estimateurs relatifs ; les premiers produisent une position dans un repère fixe de l'environnement (on y trouvera typiquement les méthodes de localisation *absolue* de notre typologie) tandis que les seconds calculent un *déplacement* à partir de la position précédente (ce sont les méthodes d'*estimation de mouvement* et de *recalage* de notre typologie). L'historique des positions permet de gérer très simplement ces deux classes : les positions absolues sont simplement placées telles quelles dans le diagramme, tandis que les positions relatives sont cumulées avec la dernière position de l'estimateur correspondant.

Enfin, la meilleure position du robot est calculée grâce à une stratégie présentée dans le paragraphe suivant.

5.2.4 Vers une fusion et une détection de fautes

À chaque période, les nouvelles positions sont récupérées et insérées dans le diagramme. Toutes les positions entre la date de mise à jour la plus ancienne et la date courante sont alors marquées pour la réestimation.

À proprement parler, il n'y a actuellement pas d'algorithme de fusion, au sens où des positions issues d'estimateurs différents ne sont jamais « mélangées » (par moyennage ou filtrage, par exemple). En effet, nous n'avons pour l'instant pas de modèles d'erreur suffisamment élaborés pour les divers estimateurs que nous avons développés et un développement rigoureux de méthodes de fusion n'est pas encore possible. De plus,

nous avons vu qu'il est toujours possible qu'un estimateur (typiquement, l'odométrie) produise une position *fausse* sans détecter cette erreur. Le résultat de la fusion pourrait alors être catastrophique.

Nous avons néanmoins proposé une stratégie très simple, qui permet de mettre en place les mécanismes futurs : à chaque estimateur de position est attribué un score de confiance, réel compris entre 0 et 1 et indiquant un degré de fiabilité. Ainsi, l'odométrie sera déclarée *moins fiable* que l'odométrie optique et le GPS sera affecté du score maximal (lorsqu'il fonctionne en différence de phase). Étant données les caractéristiques individuelles de chaque position produite à un instant donné, le meilleur estimateur est choisi et permet de calculer la position courante du robot la plus réaliste.

Il est important de remarquer qu'il n'est pas possible de simplement stocker la position courante du robot telle quelle : la présence de plusieurs estimateurs fait que celle-ci peut être amené à faire des « sauts » virtuels lors de corrections importantes (par des méthodes de recalage, par exemple). Étant donné que certains modules peuvent avoir besoin d'une continuité dans l'évolution de la position (un module d'asservissement sur une trajectoire par exemple), ces sauts doivent être évités.

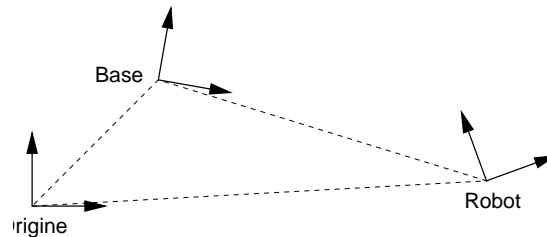


FIG. 5.7 – Repère virtuel permettant d'éviter des « sauts » de position.

Pour éviter ce problème, un repère virtuel, appelé « Base », est défini. Un estimateur de position, choisi comme référence locale (ce sera typiquement un *estimateur de mouvement*, comme l'odométrie ou l'odométrie optique), permet de calculer la position de ce repère par rapport au repère fixe de l'environnement, appelé « Origin ». La position fournie par l'estimateur de référence est *toujours* la transformation $Robot \rightarrow Base$ et la meilleure position est toujours $Robot \rightarrow Origin$. Ainsi, la transformation $Base \rightarrow Origin$ peut être vue comme une *correction* apportée à l'estimateur de référence, mais définie à l'*origine* du repère fixe de l'environnement : il est alors possible de continuer à cumuler les estimations de mouvement dans le repère *Base* tout en bénéficiant de la dernière correction disponible. Cette correction peut également être vue comme un indicateur de la *dérive* des estimateurs locaux.

La position $Robot \rightarrow Origin$ est exportée telle quelle et peut-être utilisée par les modules qui souhaitent des positions absolues. La position $Robot \rightarrow Base$ ne peut être utilisée que localement, c'est à dire pour calculer un petit déplacement à partir de deux de ces positions.

Ce schéma d'intégration pourra, à terme, permettre la détection de fautes, grâce à la comparaison de l'ensemble des positions disponibles. Il est par exemple d'ores et déjà possible de définir un mécanisme de rejet simple, lorsqu'au moins *trois* estimateurs sont disponibles et que seuls deux de ces trois estimateurs s'accordent sur une même position. Il est évident qu'un modèle d'erreur sur chaque algorithme pourra permettre la mise en place d'un mécanisme plus sophistiqué.

5.3 Une démonstration de navigation autonome

La démonstration présentée ici a permis de réaliser plusieurs tâches d'exploration, sur des distances d'une centaine de mètres environ. Un but à atteindre est sélectionné par l'opérateur dans une image vidéo et le robot décide d'un chemin permettant de rejoindre cet objectif. Au cours du déplacement, l'environnement est modélisé et plusieurs algorithmes de localisation sont mis en œuvre. Lorsque le robot a atteint son objectif, il est possible de lui demander de retourner à sa position initiale.

Cette section présente succinctement l'ensemble des fonctions permettant de réaliser les diverses tâches impliquées dans la réalisation de cette mission.

5.3.1 Couche fonctionnelle

Modélisation de l'environnement

Deux fonctionnalités différentes coexistent à bord du robot et servent chacune des objectifs distincts : un modèle qualitatif permet la génération de trajectoires et la planification de chemins sur des terrains simples et un modèle numérique permet la planification de trajectoires en terrains accidentés.

Le modèle qualitatif fournit une description de l'environnement en termes de *classes de navigabilité*, à partir d'images 3-D issues de stéréo-vision [Lacroix 1995a]. Les images sont discrétisées en *cellules* régulières dans le plan des caméras et à chaque cellule est attribuée une probabilité d'appartenance à une classe particulière de terrain, grâce à un classificateur bayésien.

Le modèle numérique est obtenu grâce à la méthode présentée dans le chapitre 4.

Génération des déplacements

Il existe trois méthodes, adaptées à trois situations différentes.

Lorsque le terrain est relativement plan et dégagé d'obstacles, le robot se déplace grâce au modèle qualitatif. Des arcs de cercles sont évalués en termes de traversabilité et une heuristique permet de combiner risque et attraction vers le but à atteindre [Haddad 1998b]. Le risque est défini comme étant la probabilité de rencontrer un obstacle le long de chaque arc.

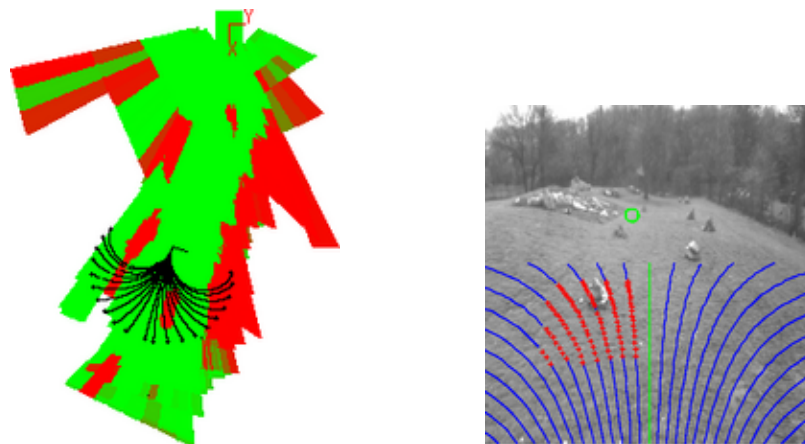


FIG. 5.8 – Génération de déplacement élémentaires.

Si le terrain est plus accidenté, la notion d'obstacle devient dépendante de la structure de locomotion utilisée. Un planificateur permet d'évaluer des trajectoires grâce à

l'analyse d'un ensemble de positions discrètes et un placement du robot sur les cartes d'élévation construites [Bonnafous 2001].

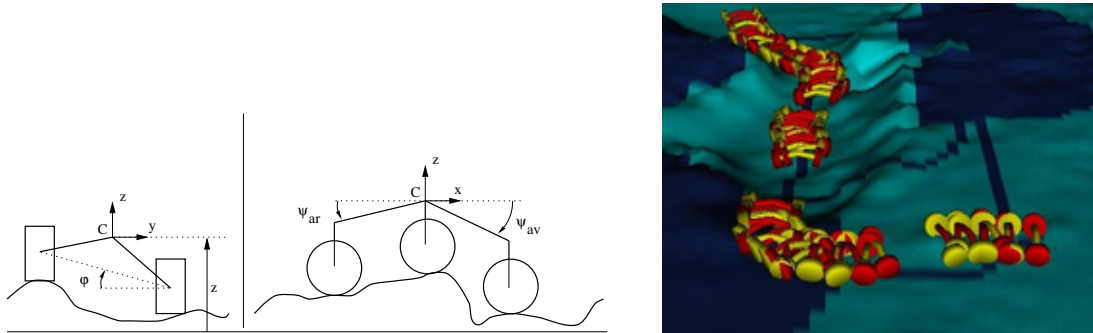


FIG. 5.9 – Les angles de configuration permettant le placement sur les cartes d'élévations.

Enfin, en présence d'impasses, il est nécessaire de mettre en place une stratégie plus globale. Pour cela, le modèle qualitatif est agrégé en régions de même nature qui définissent un graphe [Lacroix 1995b]. L'exploration de ce graphe permet de déterminer un chemin permettant de rallier l'objectif et un sous-but est ensuite calculé, dans les limites des zones déjà explorées. Les algorithmes locaux sont alors à nouveau activés et permettent de rejoindre ce sous-but.

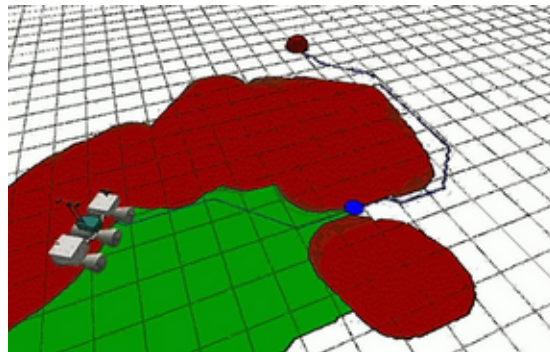


FIG. 5.10 – Planification et sélection d'un sous-but.

Localisation

Deux méthodes sont utilisées : l'odométrie présentée dans le chapitre 2 et l'odométrie optique présentée dans le chapitre 3. Elles sont toutes les deux activées systématiquement, intégrées dans le schéma présenté dans la section précédente.

L'odométrie optique requérant un temps de calcul relativement long (de l'ordre de 5 secondes), l'odométrie permet d'avoir une position à plus haute fréquence et, grâce au mécanisme de fusion, bénéficiant de la dernière correction apportée par l'odométrie optique.

L'absence de méthode de recalage fait néanmoins cruellement défaut, en particulier pour les démonstrations les plus longues.

Suivi visuel de l'objectif

Un module que nous avons développé permet le maintien dans le champ de vue des caméras de l'objectif à atteindre. Son principe est des plus simples et sa présence ne sert qu'à démontrer l'ensemble des autres fonctionnalités.

L'objectif est constitué d'un rectangle de petite taille, extrait d'une image d'une des caméras du robot. Une corrélation de pixels dans les images suivantes (monoculaires) permet de retrouver la nouvelle position de l'objectif et une commande en vitesse, proportionnelle au déplacement de l'objectif, de la tourelle 2-axes supportant les caméras assure le recentrage la cible. Cette stratégie s'avère donner de bons résultats étant donné la faible vitesse de déplacement du robot, mais il est évident que les cas d'échec sont nombreux.

Autres fonctionnalités

On trouve bien évidemment dans la couche fonctionnelle l'ensemble des modules fournissant des services *bas-niveau* tels que l'acquisition d'image, la stéréo-vision (voir annexe C), le contrôle des déplacements et de la vitesse du robot, ainsi que l'ensemble des modules de lecture des instruments de bord (en particulier, le GPS, qui est utilisé comme validation des estimateurs de position).

5.3.2 Contrôle d'exécution

L'ensemble des fonctionnalités sont contrôlées grâce au logiciel de contrôle interactif des modules que nous avons développé, par l'intermédiaire de scripts écrits dans le langage Tcl.

La stratégie de navigation utilisée consiste tout d'abord à calculer systématiquement les deux modèles de l'environnement (modèle qualitatif et cartes d'élévation). L'odométrie et l'odométrie optique fonctionnent également de façon permanente. Les déplacements sont ensuite générés grâce à l'algorithme de navigation en terrains plats, qui requiert un temps de calcul négligeable. Si aucun arcs élémentaire permettant de se rapprocher du but n'est trouvé, l'algorithme de planification sur les cartes d'élévation est employé. Ce dernier génère des déplacements jusqu'à ce que l'algorithme pour les terrains plats fonctionne à nouveau ou qu'aucune trajectoire ne soit trouvée sur les cartes d'élévation. Dans ce cas, la méthode de planification globale est employée, un sous but est calculé, et l'ensemble de la stratégie est de nouveau appliqué.

5.4 Conclusions

Nous avons insisté sur le fait qu'une navigation autonome à long terme ne pouvait être possible que grâce à l'utilisation d'un vaste ensemble de fonctionnalités. La gestion de cette diversité d'algorithmes amène inévitablement au développement d'un système d'intégration complexe, ainsi qu'au développement d'outils permettant cette intégration et facilitant le développement.

Au cours de la réalisation de démonstrations appliquées à une plateforme réelle, nous avons pu exhiber des limitations dans les outils d'intégration disponibles au LAAS. Loin de remettre en cause ces derniers, elles ont, au contraire, permis de définir les directions des développements futurs et des ébauches de solutions ont pu être proposées.

En particulier, nous avons défini un schéma d'intégration destiné à gérer un ensemble quelconque d'algorithmes de localisation, permettant de d'obtenir un système capable des les intégrer et de les déclencher de façon modulaire.

Les résultats obtenus sont encourageants. Le robot *Lama* est maintenant capable de se mouvoir de façon autonome, sur une centaine de mètres, sur divers types de terrains, et les mouvements sont générés grâce à une perception active de l'environnement, au cours des déplacements.

Conclusions

Cette thèse s'est inscrite dans le cadre du projet d'Expérimentations de Déplacement en Environnements Naturels (EDEN) du LAAS-CNRS, grâce à un co-financement CNRS et Aérospatiale. Ce travail a donné lieu à 10 publications dans des conférences internationales et 2 contrats avec l'Aérospatiale et le CNES. Les principales contributions en sont :

Chapitre 1 — Navigation autonome et localisation

Une typologie des méthodes de localisation a été définie. Trois classes d'algorithmes ont été exhibées : les méthodes d'estimation de mouvement, les méthodes de recalage et les méthodes de localisation absolue. Le critère principal permettant de les distinguer est la *portée* de la position produite, en termes d'erreur relativement à la distance parcourue. Au moins un algorithme de chaque classe est nécessaire pour le développement de robots navigants de façon autonome.

Chapitre 2 — L'odométrie

Une synthèse sur la commande des châssis de type « Marsokhod » a permis de montrer que l'odométrie utilisée sans capteurs supplémentaires ne permet ni le contrôle fin des déplacements, ni une estimation correcte de la position sur ces plateformes. L'ajout d'un compas, puis d'un gyromètre, a été analysé et les améliorations obtenues ont été présentées. L'utilisation d'inclinomètres a permis de développer le calcul de la position $3-D$ du robot. Une étude de quelques données comme la vitesse des roues ou les courants consommés selon le type de terrain a montré qu'une qualification en ligne ou une détection des fautes de l'odométrie semblait possible.

Chapitre 3 — L'odométrie optique

Une méthode originale d'estimation visuelle de mouvement permettant de calculer une position $3-D$ de façon incrémentale a été présentée. Son intégration à bord du robot *Lama* a été détaillée des résultats obtenus sur ce robot, ainsi que des

résultats préliminaires obtenus sur un ballon dirigeable, ont été détaillés. Une analyse statistique des erreurs commises a été effectuée, et l'influence des paramètres les plus importants a été analysée. La méthode a été jugée très performante et le logiciel développé fera l'objet d'un transfert vers le CNES.

Chapitre 4 — Les cartes d'élévation

Une méthode de construction incrémentale de cartes d'élévation a été développée pour le robot *Lama*. En particulier, des techniques de construction probabilistes ont été évaluées et ont permis de qualifier des méthodes de construction plus simples couramment employées. Une structure tolérante des corrections *a posteriori* sur les différentes positions par lesquelles le robot est passé a été proposée pour le stockage des modèles construits. Une méthode de recalage basée sur la minimisation d'une distance entre une image 3-D local et le modèle incrémental a été évaluée, et diverses techniques d'extraction, de modélisation ou de recalage sur des amers ont été proposées, notamment dans le cadre d'un contrat avec l'Aérospatiale.

Chapitre 5 — Intégration logicielle

Une intégration de l'ensemble des contributions a pu être faite au sein de l'architecture définie par le LAAS. Ces travaux ont abouti à une réflexion générale sur la définition du flux de contrôle et de données. En particulier, un outil générique, permettant le contrôle interactif des fonctionnalités présentes à bord des robots, a été développé. Des solutions permettant de résoudre des difficultés liées à l'intégration d'algorithmes de localisation ont été proposées : un mécanisme de gestion centralisée de la géométrie et un module de gestion d'un nombre quelconque d'estimateurs de position ont pu faciliter l'intégration des méthodes présentées et permettront le développement de nouvelles méthodes. Une démonstration de navigation autonome sur une centaine de mètres a été mise en place et évaluée de nombreuses fois, en particulier à la Cité de l'Espace à Toulouse pour une exposition destinée au grand public.

Ces travaux ne présentent pas un aboutissement. En particulier, la qualification plus précise des algorithmes développés représente la prochaine étape à réaliser. Cette qualification permettra d'améliorer encore la qualité de la position calculée et augmentera la robustesse vis à vis des erreurs de localisation par l'intermédiaire d'une *fusion* effective des différentes positions, ainsi qu'une détection de fautes.

L'étude de méthodes de recalage, grâce aux pistes proposées, devra également être accomplie. Ainsi, le robot pourra se déplacer sur plusieurs centaines de mètres et construire des modèles conséquents de l'environnement. Lorsqu'une zone suffisamment grande pourra être explorée de façon autonome, les méthodes de localisation absolue pourront alors être abordées.

Du point de vue de l'implémentation, les réflexions présentées sur le flux de donnée et de contrôle pourront être approfondies, permettant ainsi une intégration plus facile, plus modulaire et plus rapide de nouvelles méthodes de localisation.

Enfin, lorsque ces étapes seront maîtrisées, la réalisation de missions effectives sera possible. Les robots autonomes pourront alors remplir leur rôle d'exploration et effectuer des tâches utiles dans les zones où l'homme ne peut intervenir directement.

Annexes

A

Le robot *Lama*

Lorsque le robot Lama est arrivé au laboratoire au début de l'année 1996, il avait été partiellement équipé par la société Alcatel Space Industries (figure A.1). Outre les capteurs permettant de connaître la configuration du châssis, il était muni d'un accéléromètre deux axes et d'un compas magnétique. Deux cartes 68040 assuraient l'acquisition des données de ces capteurs et le contrôle de la locomotion, grâce à des logiciels développés par Alcatel. La commande des moteurs était de type *bang-bang*, le système de contrôle des moteurs fourni par le VNII-Transmach, concepteur et constructeur du châssis, étant basé sur une série de relais.

Depuis, cet équipement de base a été complètement redéfini, et de nouveaux instruments sont venus s'ajouter. Il s'agit d'un travail continu et important, essentiellement réalisé par les ingénieurs et techniciens du laboratoire, et auquel nous avons contribué, notamment en développant les pilotes de différents instruments. Les grandes étapes de ces travaux sont les suivantes :

- Remplacement des relais de contrôle de la puissance par des servo-amplificateurs, permettant ainsi un contrôle fin des trajectoires. Pour cela, les capteurs magnétiques de tour de roue qui équipaient le robot ont été remplacés par des codeurs optiques plus précis.
- Installation de deux bancs stéréoscopiques orientables, l'un monté sur un mat solidaire de l'essieu central, l'autre étant solidaire de l'essieu avant.
- Installation de convertisseurs d'énergie permettant l'alimentation à partir du secteur.

Bien entendu, l'architecture informatique embarquée a évolué sans cesse durant ces travaux : de ce point de vue, Lama est aujourd'hui très bien instrumenté (figure A.1), et continue toujours à évoluer.

Dans cette annexe, nous présentons les caractéristiques des différents instruments dont les données ont été considérées dans différents chapitres du manuscrit.



FIG. A.1 – Évolution du robot Lama. De gauche à droite : lors de sa livraison à Alcatel Space Industries en 1995, tel qu'il est arrivé au laboratoire début 1996, et enfin dans son état actuel (fin de l'année 2000).

A.1 Géométrie du châssis

A.2 Capteurs proprioceptifs

A.2.1 Les capteurs de configuration

Ces capteurs permettent de connaître les trois angles qui définissent la configuration du châssis (figure A.2). Il s'agit de potentiomètres, dont la mesure de la tension aux bornes permet, après numérisation, d'obtenir une mesure d'angle avec une résolution de l'ordre du 1/100 de degré. Ils ont été calibrés lors de la construction du châssis (détermination des offsets et facteurs d'échelle).

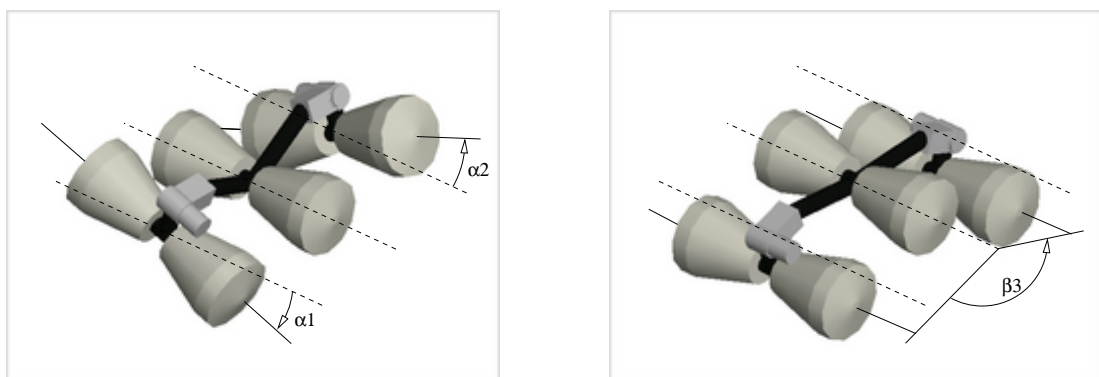


FIG. A.2 – Angles de configuration interne du châssis de Lama : au nombre de trois, ils sont mesurés grâce à des potentiomètres.

Ces trois angles correspondent aux articulations *passives* du châssis : deux autres potentiomètres permettent de connaître les angles formés par les bras du robot, qui sont activement contrôlés lors de l'application du mode de locomotion péristaltique (que nous n'avons pas considéré durant nos travaux). Enfin, l'atteinte de valeurs extrémales pour chacun des trois articulations passives est détectée par des contacteurs, dont le déclenchement provoque un arrêt d'urgence au niveau de la couche de locomotion.

A.2.2 Les codeurs odométriques

Il s'agit de codeurs incrémentaux qui délivrent 500 impulsions par tour. Ils sont montés sur l'arbre rapide du moteur (avant réduction) : le rapport des réducteurs étant de 1/503, il délivrent donc 251500 impulsions par tour de roue¹ : ils sont donc extrêmement précis.

¹Soit une impulsion pour un déplacement linéaire de 5.610^{-5} m !

A.2.3 Le compas magnétique

Remarque : Stricto sensu, les seuls capteurs proprioceptifs du robot sont les capteurs de configuration et les odomètres. Néanmoins, par abus de langage et habitude, sont classés dans la catégorie des capteurs proprioceptifs les capteurs qui permettent de déterminer la position du robot sans percevoir d'éléments particulier de l'environnement (capteurs inertiels, compas magnétiques et inclinomètres).

Le compas installé à bord de Lama est un instrument de marque Navico (modèle HS8000), qui équipe habituellement des bateaux de plaisance. Il s'agit d'un détecteur de flux magnétique monté sur un système de compensation d'attitude ("gimballed"), qui fournit une mesure du cap par rapport au nord magnétique sur une liaison série à la fréquence de 5 Hertz. Sa précision annoncée est de $\pm 1.0^\circ$.

Calibration : Comme pour tout compas à flux magnétique, les mesures sont biaisées par la présence de masses métalliques ou de champs magnétiques parasites dans son environnement immédiat. Le modèle installé à bord de Lama comprend une procédure de calibrage interne, mais qui est trop peu précise pour être vraiment utile (la table de correction des valeurs n'est calculée que pour huit valeurs d'angles).

Nous avons donc implémenté une procédure de calibrage : la figure A.3 montre l'importance des perturbations, produisant pour certaines orientations des erreurs allant jusqu'à 35° .

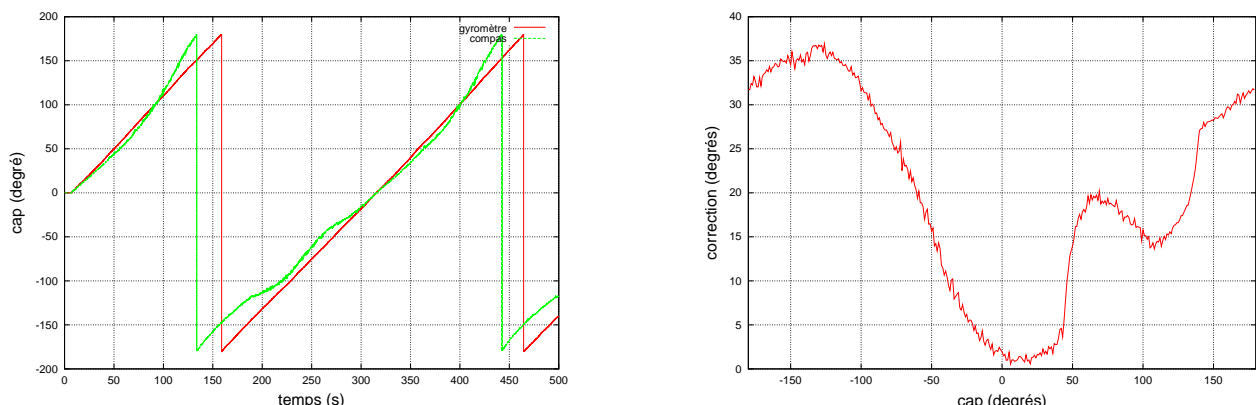


FIG. A.3 – Calibration du compas. En intégrant la vitesse angulaire mesurée par le gyromètre, il est possible d'obtenir une estimation du cap du robot. La différence avec le cap mesuré par le compas est flagrante (graphique de gauche) ; elle provient des perturbations électromagnétiques créées par le châssis du robot. Cette comparaison permet de construire une table de calibrage du compas (graphique de droite) qui, en fonction de l'angle donné, donne la correction à effectuer. Cette table est bien-sûr dépendante du robot, mais aussi — très certainement — de l'intensité des courants circulant dans les moteurs des roues.

L'obtention d'une table de calibrage est très facile lorsqu'on dispose d'une autre référence pour mesurer l'angle de cap. Sur le robot *Lama*, il est possible d'utiliser le gyromètre en intégrant sa vitesse angulaire, sur une période de temps suffisamment courte pour que la dérive soit négligeable : la courbe de calibrage est alors déduite de la différence entre les deux angles, comme sur la figure A.3.

Lorsqu'une autre référence pour le cap n'est pas disponible, il est néanmoins possible d'obtenir une estimation de la table en commandant le robot pour qu'il effectue des rotations à vitesse angulaire constante. Si le contrôle du mouvement est suffisamment bon, l'angle théorique de cap se déduit de la consigne, par intégration. En revanche, la précision de la table est liée à la précision de la commande, et si le sol n'est pas

régulier cette opération est bien-sûr beaucoup moins fiable. Cette façon de procéder a été utilisée sur *Lama* avant l'installation du gyromètre.

A.2.4 Le gyromètre

Lama est équipé d'un gyroscope à fibre optique (modèle KVH E-Core 1000), qui fournit une estimée de la vitesse angulaire du robot, dont on peut déduire une valeur de la position en cap par intégration dans le temps. Les données sont produites à 10Hertz avec une résolution de légèrement supérieure à 0.001 deg.s^{-1} .

Il existe un biais dans la mesure de la vitesse angulaire, qui depend essentiellement de la température du capteur. Ce biais est estimé par la mesure de la vitesse angulaire du gyromètre à l'arrêt complet, pendant un laps de temps plus ou moins long (en pratique le fabriquant conseille 30 secondes). Il suffit alors de le retrancher de toutes les mesures ultérieures.

Modèle d'erreur. Nous avons fait quelques mesures à l'arrêt, afin d'estimer les erreurs sur les données produites : elle suivent une répartition gaussienne avec un écart type σ_ω de l'ordre de 0.015 deg.s^{-1} . Nous pouvons déduire de cette valeur une estimée de l'erreur sur le cap obtenu par intégration des vitesses angulaires dans le temps :

$$\sigma_\theta = \tau\sigma_\omega\sqrt{n}$$

où n est le nombre d'échantillons considérés et τ la fréquence d'échantillonnage².

En intégrant l'ensemble des données fournies à 10Hertz , cela donne une estimée du cap avec un écart type de l'ordre de 0.3° au bout d'une heure de fonctionnement : l'excellente qualité du cap ainsi obtenue est très bien vérifiée expérimentalement (figure A.4).

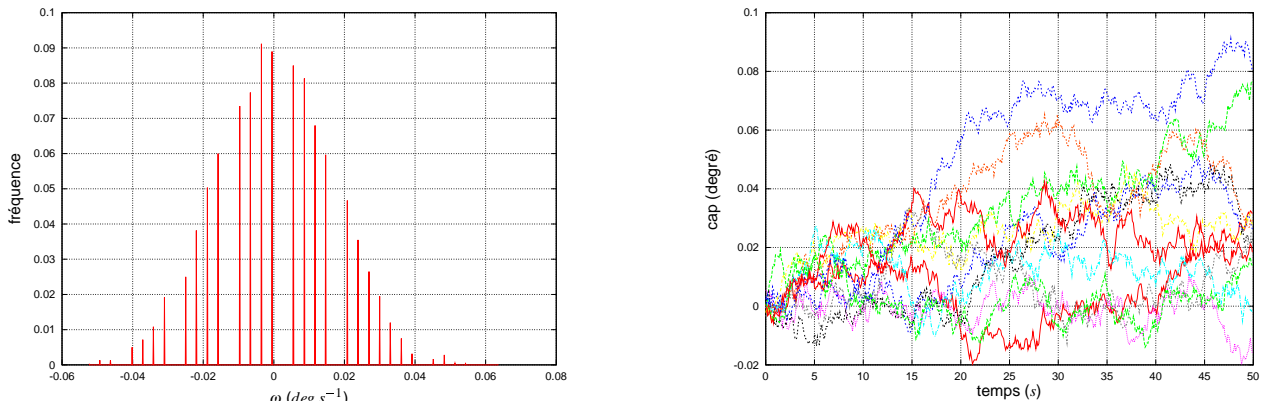


FIG. A.4 – Dérive du gyromètre : un histogramme des vitesses angulaires mesurées à l'arrêt, après élimination du biais, montre que les erreurs sur les valeurs suivent une gaussienne dont l'écart-type est de l'ordre de 0.015 deg.s^{-1} (graphique de gauche). Ce bruit se traduit par une dérive de l'angle de cap obtenu par intégration : le graphique de droite montre une dizaine d'estimations.

²Cette équation se déduit du théorème central de la limite, qui dit que l'écart type d'une somme de n valeurs aléatoires d'une distribution d'écart type σ vaut $\sigma\sqrt{n}$: dans notre cas, l'intégration $\theta = \sum_0^n \tau\omega$ correspond à la somme de n variables $\tau\omega$, dont l'écart type est de $\tau\sigma_\omega$.

A.2.5 Le GPS

Nos collègues du CNES nous ont prêté un système de localisation GPS différentiel à mesure de phase (modèle Leica MC1000), qui fournit une estimée de position à 10 *Hertz*, dont la précision annoncée est de 1cm à $\pm\sigma$ sur les valeurs X et Y , et de 2cm à $\pm\sigma$ sur Z (altitude). Ces spécifications sont très bien vérifiées expérimentalement (figure A.5).

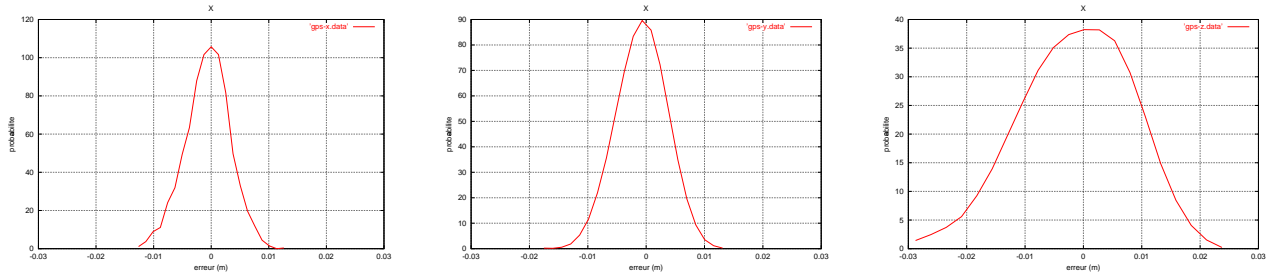


FIG. A.5 – *Densité de probabilité des erreurs commises par le GPS à l’arrêt : la répartition des erreurs suit bien une gaussienne, dont l’écart type correspond à la précision annoncée.*

La précision de ce système en fait une référence idéale pour valider des algorithmes de localisation. Notons cependant que le masquage des satellites par des bâtiments ou de la végétation fait ”décrocher” le système, qui a besoin de 30 secondes à 2 minutes à l’arrêt pour pouvoir fournir une nouvelle estimée de position lorsqu’un nombre suffisant de satellites est en à nouveau vue (les systèmes plus récents sont un peu plus rapides). Enfin, la technologie des récepteurs à différence de phase fait que la disparition de la ”selective availability” le 1er Mai 2000 n’a en rien amélioré les performances de ce système³.

A.3 Capteurs extéroceptifs

Les seuls capteurs extéroceptifs dont est actuellement équipé Lama sont deux bancs stéréoscopiques, montés sur une tourelle orientable suivant deux axes concourants (modèle ”Directed Perception”, qui est très répandu dans les laboratoires de robotique).

Le banc qui est situé à l’avant du robot a pour vocation de percevoir le terrain situé juste devant le robot (fonctions de détection d’obstacles et de modélisation), et son orientation n’est pas contrôlée de manière active. Ses caméras (modèle Digital Vision VHR 2000, matrices CCD d’un tiers de pouce, de 752×582 pixels), sont donc équipées d’objectifs grand angle (focale 2.6mm, ce qui donne un champ de vue horizontal de xxx° et vertical de xxx° (figure A.6).

Le banc situé en haut du mat central est plutôt dédié au suivi de but et à la modélisation d’amers : son champ de vue est plus étroit, et son orientation est activement contrôlée durant l’exécution des déplacements. Il est constitué de deux caméras couleur tri-CCD SONY XC-003⁴, dont les matrices CCD d’un tiers de pouce comprennent 752×582 pixels tiers de pouce, et d’objectifs de longueur focale 4mm, ce qui donne un champ de vue horizontal de xxx° et vertical de xxx° (figure A.7).

Les images d’un banc sont acquises de manière synchrone et numérisées par une carte d’acquisition en 768×568 pixels.

³Nous avons cependant été victime du ”bug de la semaine 1024” en Août 1999 - qui a été résolu par une mise à jour le logiciel du système.

⁴Seul le signal d’intensité a été utilisé durant tous les travaux de ce manuscrit.



FIG. A.6 – Une image brute du banc stéréoscopique du bas, et après correction de la distorsion (au centre). Le champ de vue est représenté sur la figure de droite.



FIG. A.7 – Vue de la même scène que la figure A.6 par le banc du haut - on distingue le second banc dans le bas de l'image. La correction de la distorsion est à peine perceptible, mais reste nécessaire lors de la rectification des images.

A.4 Architecture informatique

Les cartes de traitements de Lama sont montées dans un rack VME, et opérées par VxWorks. Deux cartes 68040 ont pour rôle d'acquérir les données fournies par les différents capteurs (hors caméras) et de contrôler l'ensemble des actionneurs (torelles orientables et locomotion). Les traitements plus lourds (vision, modélisation, génération de trajectoires) sont effectués sur deux cartes Power PC.

B

Geométrie sur le robot

Lama

B.1 Conventions de notations

Une matrice de changement repère notée $A \rightarrow B$ permet de faire passer les coordonnées d'un point P exprimées dans le repère A en coordonnées exprimées dans le repère B . Elles sont exprimées sous la forme de matrices homogènes :

$$P_B = A \rightarrow B.P_A$$

Le vecteur translation de $A \rightarrow B$ donne donc les coordonnées de l'origine du repère A exprimées dans le repère B .

Avec cette notation, la composition de repères s'écrit :

$$A \rightarrow C = A \rightarrow B \circ B \rightarrow C$$

qui se calcule :

$$A \rightarrow C = B \rightarrow C.A \rightarrow B$$

Enfin, une rotation d'angle θ autour d'un axe X est notée $\underline{X}(\theta)$. Les angles d'Euler utilisés ici correspondent à $\underline{Z}(\theta)$, $\underline{Y}(\phi)$, $\underline{X}(\psi)$.

B.2 Définition des différents repères

B.2.1 Les repères liés au banc stéréoscopique

On peut énumérer trois repères liés au banc stéréoscopique :

- Le repère lié à la caméra droite (**CamD**), dont l'origine est située au centre optique de la caméra.

- Le repère lié à la caméra gauche (**CamG**), dont l'origine est située au centre optique de la caméra (c'est dans ce repère que sont exprimés les points 3D déterminés par le module de stéréo-corrélation).
- Le repère lié à la platine (**Platine**), dont l'origine est à l'intersection des axes de rotation site et azimut¹, et qui tourne avec les axes de la platine (il est solidaire des deux repères caméras).

Ces trois repères sont orientés *approximativement* de la même façon (la rotation entre eux n'est pas rigoureusement nulle) : l'axe des X correspond aux axes de visées des caméras, l'axe Z est orienté vers le haut (donc l'axe Y est orienté vers la gauche), et ils sont rigidement liés entre eux.

B.2.2 Les repères liés au robot

Dans le cadre de cette thèse, nous n'avons considéré que les repères suivants :

- Le repère **Mat**, dont l'origine est située à l'intersection des axes de rotation de la platine, mais qui est solidaire du mat (et donc de l'essieu central du robot).
- Le repère **Robot**, dont l'origine est située sur la projection au sol du centre de l'essieu central suivant l'axe du mat, et dont les axes sont solidaires de l'essieu central (X orienté vers l'avant, Y dans l'axe de l'essieu, vers la gauche, et Z suivant l'axe du mat, vers le haut).
- Le repère **RobotRed** (« Robot Redressé »), dont l'origine est située au même endroit que **Robot**, mais dont l'axe des Z est vertical.

Les deux repères **Mat** et **Robot** sont orientés *exactement* de la même façon, et sont rigidement liés.

Deux repères supplémentaires ont été définis, mais ne sont pas considérés actuellement (car non nécessaires) : il s'agit des repères liés aux essieux avant et arrière **RobotAvant** et **RobotArriere**.

B.2.3 Le repère global

Un repère global, lié au terrain (noté **Global**) est également défini. Pour être utilisé en combinaison avec le GPS, il a été défini avec les conventions suivantes : axe OZ orienté verticalement vers le haut, axe OX orienté horizontalement vers le nord (et donc axe OY orienté vers l'ouest).

B.3 Les paramètres perçus permettant de déterminer les repères

B.3.1 Les paramètres directement lus sur les instruments du robot

- Les paramètres d'orientation de la platine : pan θ_p et tilt ϕ_p .
Ces deux angles sont nuls à l'initialisation de la platine. θ_p est positif lorsque la platine tourne vers la gauche et ϕ_p est positif lorsque la platine penche vers le haut.
- Les angles fournis par les inclinomètres (solidaires de l'essieu central) : site ϕ_r et gîte ψ_r .
Ces deux angles sont nuls lorsque le mat est vertical. Le site ϕ_r est positif lorsque le mat penche vers l'arrière et le gîte ψ_r est positif lorsque le mat penche vers la gauche.

¹On suppose que les deux axes de la platine s'intersectent, ce qui n'est pas explicite dans les spécifications techniques de la platine de *Lama*

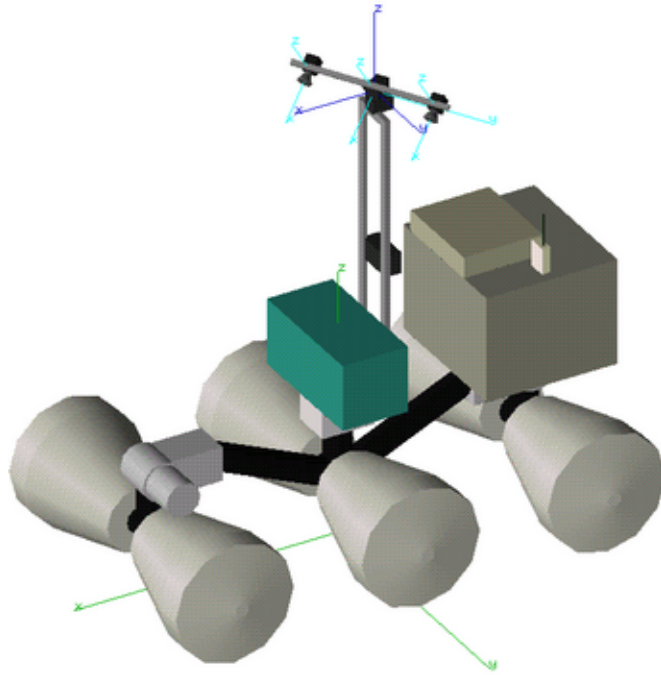


FIG. B.1 – Les repères liés à Lama.

- Le cap θ_r fourni par le compas. Il donne l'orientation du compas par rapport au Nord magnétique, positif vers l'ouest.
- Les angles donnant la configuration du robot, fournis par les codeurs du châssis (voir figure B.2) :
 - α_1 (α_2) est l'angle que fait l'essieu avant (respectivement arrière) avec l'essieu central. Ces deux angles sont nuls lorsque les axes sont parallèles. α_1 est positif lorsque l'axe avant penche vers la droite et α_2 est positif lorsque l'axe arrière penche vers la droite.
 - β_1 (β_2 et β_3) donne l'angle de l'articulation avant (respectivement arrière et centrale) du robot. β_3 donne le demi angle formé entre les deux bras (donc l'angle formé entre le mat et l'un des deux bras) : le mat est toujours situé sur la bissectrice des deux bras. Sa valeur lorsque le robot est à plat est d'environ 83° . β_1 (β_2) est nul lorsque l'essieu avant (respectivement arrière) est en position intermédiaire (sur un sol plan, le bras de l'essieu est alors approximativement vertical). β_1 est positif lorsque l'axe avant est reculé (*i.e.* "plié") et β_2 est positif lorsque l'axe arrière est avancé (*i.e.* "plié").

B.3.2 Les paramètres calculés par différents modules

La totalité des paramètres suivants sont disponibles, et systématiquement calculés sur *Lama* :

- La position odométrique $3D$ du robot, déduite des informations des codeurs des roues centrales et du gyromètre : $(x_o, y_o, z_o, \theta_o)$. Elle est actuellement déterminée par le module *lloco* relativement à une origine arbitrairement choisie.
- La position obtenue par odométrie optique, en utilisant le banc stéréoscopique

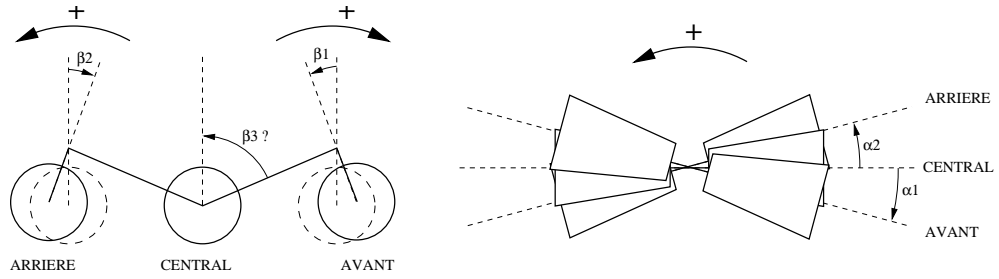


FIG. B.2 – Les angles du châssis de Lama - vues de droite et de face.

	CamD	CamG	Platine	Mat	Robot	RobotRed
CamD		$T = C^{ste}$ $R = C^{ste}$	$T = C^{ste}$ $R = C^{ste}$			
CamG	$T = C^{ste}$ $R = C^{ste}$		$T = C^{ste}$ $R = C^{ste}$			
Platine	$T = C^{ste}$ $R = C^{ste}$	$T = C^{ste}$ $R = C^{ste}$		$T = 0$ $R = f(\theta_p, \phi_p)$		
Mat			$T = 0$ $R = f(\theta_p, \phi_p)$		$T = C^{ste}$ $R = 0$	
Robot				$T = C^{ste}$ $R = 0$		$T = 0$ $R = f(\phi_r, \psi_r)$
RobotRed					$T = 0$ $R = f(\phi_r, \psi_r)$	

TAB. B.1 – Transformations entre les différents repères.

- supérieur : $(x_s, y_s, z_s, \theta_s, \phi_s, \psi_s)$.
- La position GPS, dans le repère global : (x_g, y_g, z_g) .

B.4 Détermination des différentes transformations

Le tableau B.4 montre les paramètres qui interviennent dans la détermination des principaux changements de repères du robot.

B.4.1 Les transformations fixes

- La transformation $CamD \rightarrow CamG$ entre ces deux repères est rigide, et est déterminée par la calibration du banc stéréoscopique. Il existe pour l’instant une estimation de $CamG \rightarrow Platine$ sous la forme d’une translation pure

$$T_{CamG \rightarrow Platine} = (T_x, T_y, T_z) = (0.0, 0.22, 0.0)$$

- La transformation $Mat \rightarrow Robot$ est elle une translation pure, qu’il reste à déterminer précisément. Il existe pour l’instant une estimation de la $Mat \rightarrow Robot$:

$$T_{Mat \rightarrow Robot} = (T_x, T_y, T_z) = (-0.15, 0.0, 1.78)$$

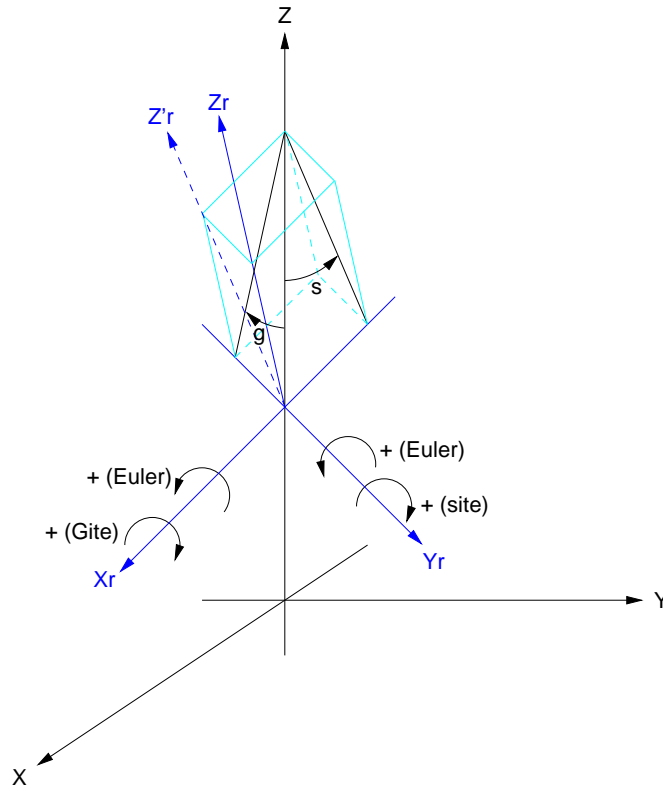


FIG. B.3 – Les angles donnés par les inclinomètres.

B.4.2 Platine vers Mat

Il suffit de composer les deux rotations données par la platine : on effectue une première rotation de $\underline{Z}(\theta_p)$, puis une seconde $\underline{X}_1(-\phi_p)$

B.4.3 Robot vers Robot_Red

Les angles données par les inclinomètres ne sont pas des angles d'Euler (voir figure B.3). On effectue donc une première rotation $\underline{X}_r(\alpha)$, qui amène l'axe Z_r sur l'axe Z'_r , puis une rotation $\underline{Y}_r(\beta)$.

On a (voir la figure) $\alpha = \text{ArcSin}\left(\frac{\sin(\psi_r)}{\cos(\phi_r)}\right)$ et $\beta = -\phi_r$.

B.5 Graphe géométrique

Un fichier de configuration dont la grammaire est donnée sur la figure B.4 permet de décrire la structure de graphe présentée dans la section précédente. Étant donné que les graphes destinés à être manipulés ne sont pas censés être excessivement complexes, la grammaire choisie est très simple. Le principe général, illustré par l'exemple de la figure B.5, est de déclarer les repères que l'on souhaite définir, puis de les relier ensuite par le mot-clé **link** en indiquant si le lien ainsi créé est statique ou mobile. Les quelques autres mot-clés présentés ci-après permettent de définir les attributs des liens (**update**, **translate**, *etc.*) ou des repères (**main**).

```

file ← declaration | declaration file
declaration ← require | frame | main | update | link
require ← "require" string ";"
frame ← "frame" framelist
framelist ← string | string "," framelist
main ← "main" string ";"
update ← "update" string ";"
link ← "link" "{" linkdecl "}"
linkdecl ← linkelt ";" | linkelt ";" linkdecl
linkelt ← "from" string
          | "to" string
          | "translate" double double double
          | "rotate" double double double
          | "update" string
string ← "A"|"Z" | "a"|"z" | digit
double ← ["-"] digit ["."] digit
digit ← "0"|"9"

```

FIG. B.4 – Grammaire du fichier de description géométrique d'un robot.

```

frame A,B,C,D;
main A;

link {
  from B; to A;
  update U2;
}

link {
  from C; to B;
  translate -1.0 1.0 0.0;
}

link {
  from D; to B;
  translate -1.0 -1.0 0.0;
}

link {
  from D; to C;
  update U1;
}

```

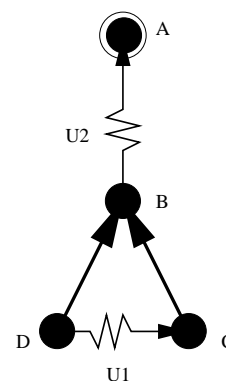


FIG. B.5 – Exemple de graphe géométrique et description textuelle associée.

- ▷ **require** Contient le nom d'une bibliothèque partagée contenant la définition des fonctions permettant la mise à jour des liens mobiles. Cette bibliothèque est chargée par *InSitu* au moment de la lecture du fichier de configuration.
- ▷ **frame** Déclare le nom d'un nouveau repère dont la position doit être calculée par *InSitu*. A l'heure actuelle, aucune contrainte n'est imposée sur ce nom. Il peut sembler avantageux de définir un ensemble de noms utilisables, dans le but d'uniformiser les descriptions des différents robots. Cependant, ces listes prédéfinies consistent plus souvent un handicap car il est difficile d'envisager toutes les situations futures. La figure ref-fig :insitu :robots montre déjà, pour le seul cas des caméras, qu'il est difficile de choisir une description compatible avec les trois types de robots présentés : le robot *Hilare Ibis* possède qu'une caméra, à l'extrémité de son bras ; le robot *Lama* possède quatre caméras réparties en deux bancs stéréo que l'on nomme couramment « haut » et « bas » ; le dirigeable possède quant à lui un banc stéréo dont les caméras sont avantageusement nommées « avant » et « arriere ».
- ▷ **main** Définit le nom du repère principal.
- ▷ **link** Crée un lien entre 2 repères. Les mots-clef **from** et **to** permettent de définir l'orientation du lien — et donc sa signification.
- ▷ **update** Placé à l'intérieur de la définition d'un lien, **update** permet de définir le nom d'une fonction chargée de calculer la transformation courante associée au lien. Placé à l'extérieur de toute définition de lien, **update** définit une fonction invoquée immédiatement avant la mise à jour de chaque lien. Cette fonction est utilisée pour factoriser la lecture des différents paramètres utilisés dans les liens du modèle.

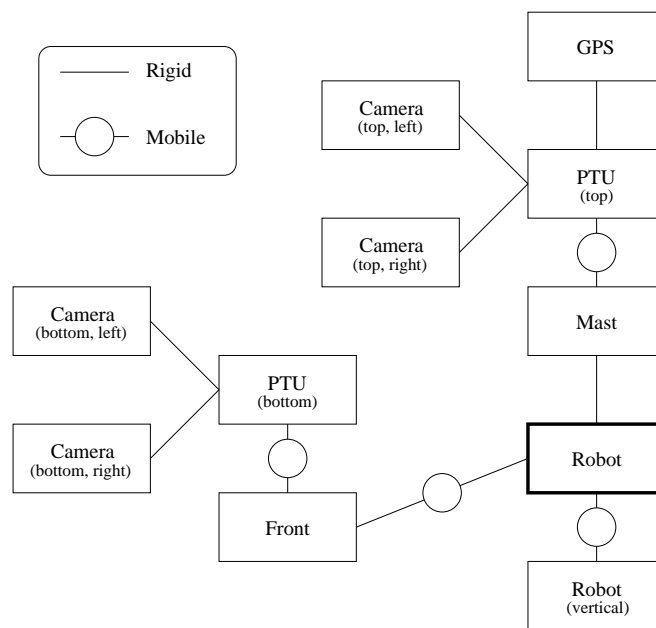


FIG. B.6 – Le graphe géométrique du robot Lama : les rectangles sont les nœuds et représentent les repères calculés et exportés. Les rectangles sont les repères dont les configurations sont exportées dans le système. Le rectangle en gras représente le repère principal. Chaque repère mobile est associé à une fonction chargée du calcul de la configuration courante.

C

Modélisation des erreurs de la stéréo-vision

Nous présentons ici une analyse empirique de l'algorithme de stéréo-vision que nous avons utilisé. Cette analyse a permis d'ébaucher un modèle d'erreur du capteur « stéréo-vision », qui permet d'estimer *en ligne* la précision des coordonnées des points 3D calculés.

C.1 Principe de l'algorithme de stéréo-vision

Le principe de l'algorithme de stéréo-vision par corrélation de pixels que nous utilisons est très classique : il se base sur la connaissance précise des positions respectives des caméras (obtenues par calibration) et sur la détermination d'appariements entre les pixels des images gauche et droite. Ses différentes étapes sont résumées sur la figure C.1 : il s'agit d'un algorithme désormais bien connu dans la littérature [Krotkov 1995, Faugeras 1993], qui donne d'excellents résultats (en terme de nombre de points 3D) dès que la scène perçue exhibe suffisamment de texture. Cet algorithme est utilisé au laboratoire depuis plusieurs années, et quelques améliorations y ont été apportées [Lasserre 1995, Haddad 1998a].

L'opération de rectification est une transformation des images qui permet d'assurer que l'épipolaire d'un point d'une image est une ligne horizontale dans l'autre image : cette opération n'est pas indispensable pour calculer une images de points 3D, mais elle permet d'effectuer de très grandes optimisations de calculs lors de la phase de mise en correspondance [Faugeras 1993].

C'est la phase de mise en correspondance qui est la plus importante : elle consiste à déterminer le correspondant d'un pixel d'une image sur la ligne épipolaire de l'autre image. Elle est réalisée par le calcul de ressemblance entre une fenêtre définie autour du pixel et des fenêtres définies autour des pixels candidats (figure C.2). De nombreux moyens de calculer cette ressemblance sont définis dans la littérature [Martin 1995], et après différentes évaluations nous avons opté pour le score de corrélation croisé norma-

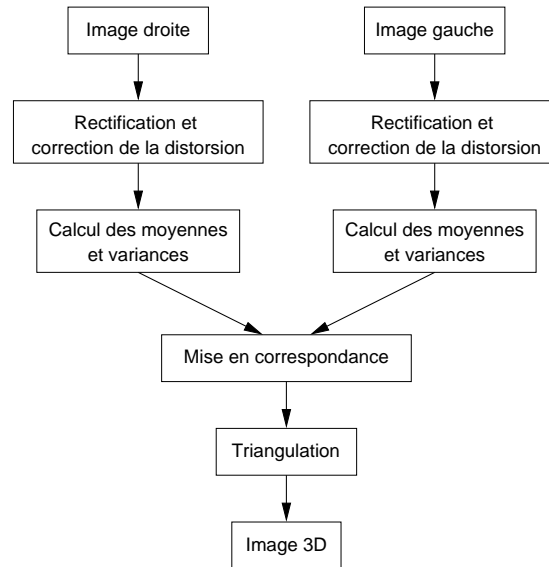


FIG. C.1 – Les différentes étapes de l’algorithme de stéréo-vision par corrélation de pixels.

lisé (ZNCC¹) :

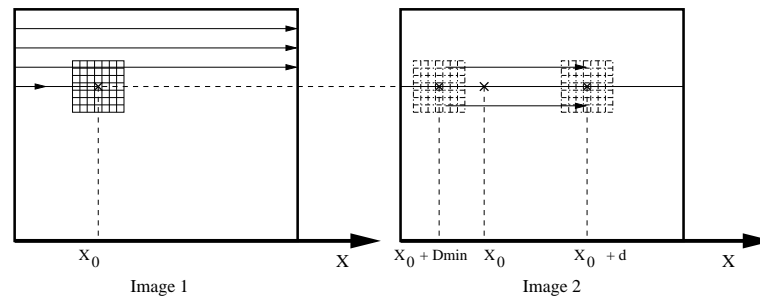


FIG. C.2 – Principe de l’étape de mise en correspondance.

$$ZNCC(x, y, d) = \frac{\frac{1}{N} \sum_{(x,y) \in F} (I_g(x+d, y) I_d(x, y)) - \frac{1}{N^2} \sum_{(x,y) \in F} (I_g(x+d, y)) \sum_{(x,y) \in F} (I_d(x, y))}{\sigma_g \sigma_d}$$

où F est la fenêtre définie autour des pixels, et d est la *disparité* entre deux pixels considérés, c’est à dire le décalage en nombre de colonnes. L’avantage de ce score de corrélation est son indépendance par rapport à toute transformation affine sur les niveaux de gris des pixels : il n’est donc pas sensible aux différences d’illumination et de dynamique de l’image qui apparaissent toujours entre deux caméras.

Un autre critère de ressemblance qui est extrêmement performant est le critère dit *Census*. Introduit dans [Zabih 1994], il très original et séduisant : il se base sur une transformation préalable des images, où à chaque pixel est associé un champ de n bits qui code la relation d’ordre entre le pixel et ses n voisins. La ressemblance entre deux fenêtres déterminées sur ces images est alors donnée par la distance de Hamming, qui énumère le nombre de bits qui diffèrent entre l’ensemble des champs de bits des fenêtres. Des optimisations similaires au calcul du critère ZNCC s’appliquent, et les performances en temps de calcul et en nombre d’appariements produits sont équivalentes.

¹Zero-mean Normalized Cross Correlation

Le seul calcul de ces scores ne permet cependant pas toujours d'établir des appariements valides : une phase dite de "corrélacion inverse", qui vérifie que le pixel apparié dans la seconde image s'apparie avec le pixel d'origine, permet d'éliminer un grand nombre de mauvais appariements, et différents tests sur les courbes de corrélacion permettent d'éliminer les faux appariements restants (figure C.3).

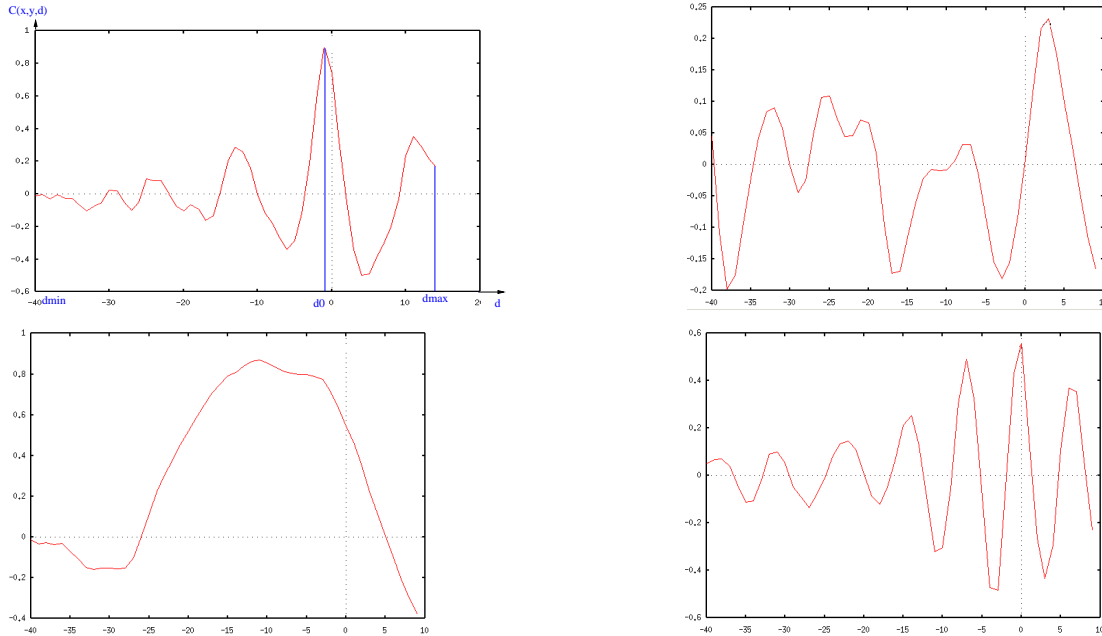


FIG. C.3 – Quelques courbes de scores de corrélacion. En haut à droite, un appariement sûr : la courbe exhibe un pic bien marqué, avec une valeur proche de 1 (maximum théorique du score ZNCC). Les trois autres courbes correspondent à de faux appariements : en haut à droite, la valeur du score au niveau du maximum est trop faible, en bas à gauche, le pic n'est pas bien marqué, ce qui correspond à une zone très faiblement texturée, et en bas à droite, des pics de valeur élevée sont répétés, ce qui correspond à la présence de motifs répétitifs dans la scène.

L'analyse de quelques images suffit à déterminer les différents seuils qui permettent d'éliminer tous les mauvais appariements, sans pour autant trop réduire le nombre de pixels effectivement corrélés. Ces valeurs dépendent de la qualité des images disponibles et du type de scènes perçues : avec les images des caméras Sony qui composent le banc stéréoscopique supérieur de Lama, aucun de ces tests n'est en fait nécessaire, la phase de corrélacion inverse suffisant à éliminer les mauvais appariements (figure C.4).

Une fois le pic de corrélacion déterminé sur l'ensemble des disparités entières considérées, une interpolation quadratique est appliquée à partir de la valeur du pic de corrélacion et des ses deux voisins. Cela permet d'affiner la détermination de la disparité, mais absolument rien ne justifie le calcul de l'interpolation de cette manière : nous allons voir que cela a des effets "pervers" sur la précision des données fournies.

Enfin, l'étape de triangulation permet d'obtenir les coordonnées des points 3D à partir des valeurs de disparité et des matrices de calibration du banc stéréoscopique.

La performance en temps de calcul de la stéréo-vision décroît avec le cube du taux de sous-échantillonnage des images. À bord du robot Lama, les images sont sous-échantillonnées d'un facteur 3 (taille 256×192 pixels), et l'ensemble de l'algorithme prend une seconde. Notons que grâce à la présence de plus en plus fréquente d'instructions vectorielles ou SIMD sur les processeurs (extension MMX des Pentium ou Activec des G4), ce temps de calcul peut être réduit par un facteur d'au moins 10.

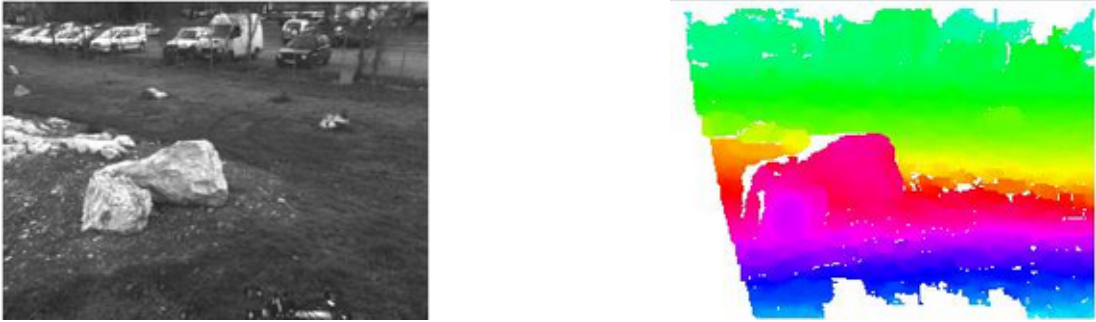


FIG. C.4 – Un résultat de stéréo-vision à partir des images acquises par le banc stéréoscopique supérieur de Lama. Les niveaux de gris de l'image de disparité (à droite) sont inversement proportionnels à la profondeur des pixels. La zone où des pixels ne sont pas appariés correspond au banc stéréoscopique inférieur : ce dernier étant mobile, la disparité maximum jusqu'à laquelle sont effectués les calculs de ressemblance est définie de façon à ne pas couvrir ce volume.

C.2 Analyse empirique de l'algorithme

C.2.1 Méthode

Afin d'avoir une estimation des erreurs non systématiques, nous avons étudié les variations de disparités calculées lors de l'application de l'algorithme à une centaine d'images d'une même scène (une étude analogue a été effectuée dans [Matthies 1992] et [Xiong 1997]).



FIG. C.5 – Les deux scènes sur lesquelles nous avons effectué l'analyse statistique, acquises avec le banc stéréoscopique supérieur de Lama à gauche, et avec le même banc équipé de caméras VHR 2000 à droite (à l'époque de l'acquisition des images de droite, les images étaient numérisées en 512×512 pixels).

C.2.2 Vers un modèle d'erreur

Pour chacun des pixels des images, il apparaît que les variations sur la disparité se répartissent selon une gaussienne (figure C.6) dont l'écart type croit lorsque la disparité diminue. Ceci s'explique par le fait que la scène est de moins en moins texturée dans l'image au fur et à mesure que l'on s'éloigne du capteur (les caméras étant parfaitement alignées, la disparité tend vers 0 à l'infini).

Un aspect plus intéressant est la corrélation qui existe entre la forme du pic de corrélation et l'écart-type sur la disparité : ce dernier tend bien sûr à grandir lorsque le pic s'aplatit. La figure C.7 montre cette corrélation : la forme du pic est ici exprimée par la moyenne des différences de score entre le maximum et les deux disparités voisines (δ). Cette corrélation est assez bien exprimée par l'équation suivante :

$$\sigma_{disp} = 0.007/(\delta + 0.05)$$

Cette fonction constitue donc le modèle d'erreur du capteur. En supposant (ce qui est assez bien vérifié) que cette erreur sur l'estimation de la disparité est une gaussienne, on arrive à l'estimation suivante de l'écart type sur la profondeur après reconstruction de l'image 3D :

$$\sigma_x = \alpha x^2$$

où α est le coefficient de l'équation de triangulation ($x = \alpha/disp$).

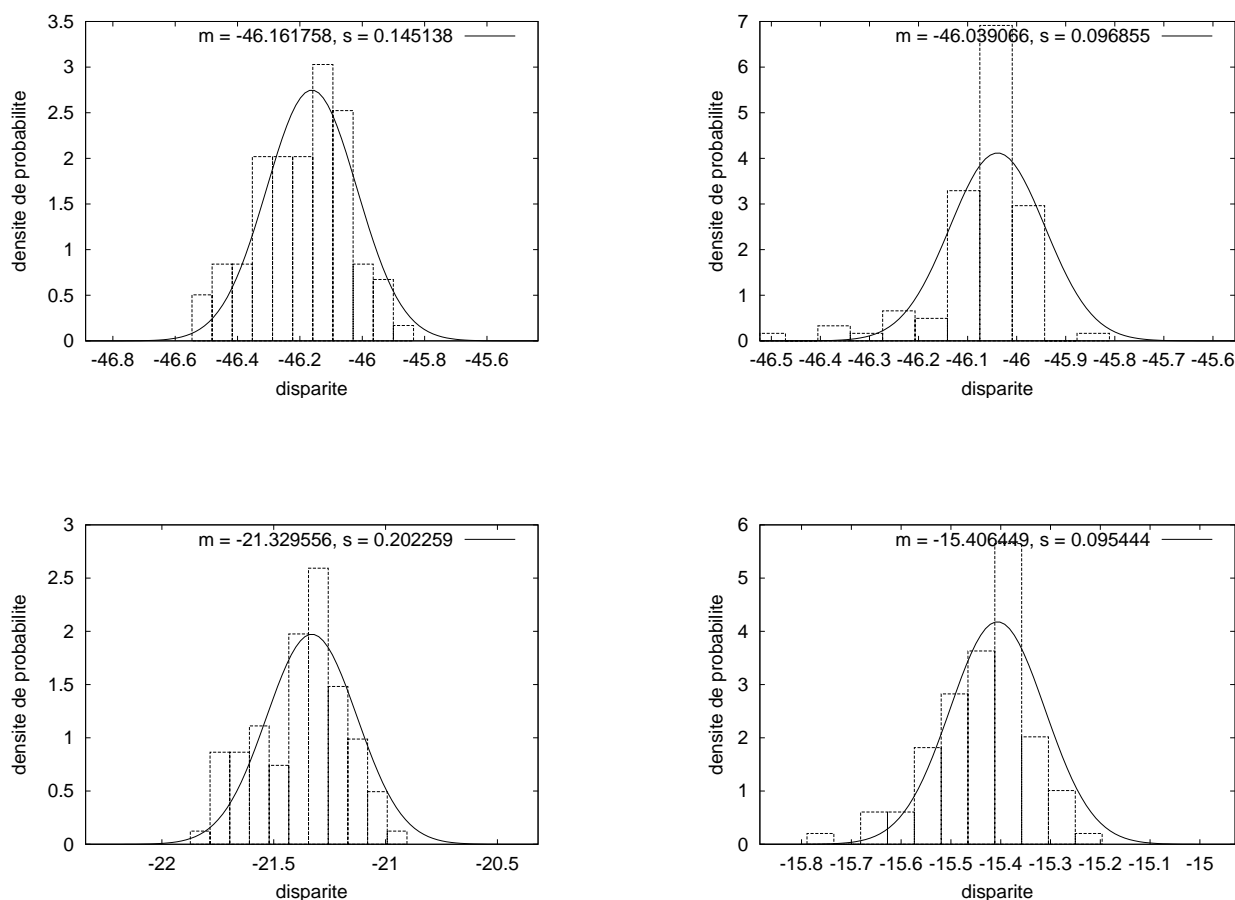


FIG. C.6 – Quelques répartitions des disparités calculées sur une centaine d'images de la scène de gauche de la figure C.5 : la répartition des disparités suit assez fidèlement une gaussienne, dont l'écart type est toujours compris entre 0.05 à 0.3 pixels (noter le changement d'échelle dans les 4 courbes présentées ici). Ces courbes ont été produites en exécutant l'algorithme de stéréovision avec des fenêtres de corrélation de 9×9 pixels.

C.2.3 Problèmes non résolus

Plusieurs problèmes restent cependant à résoudre, avant de disposer d'un réel modèle de la stéréo-corrélation. Nous avons d'ores et déjà identifié deux d'entre eux :

- L'interpolation de la courbe des scores autour du maximum est un facteur déterminant pour la précision des coordonnées 3-D calculées. En effet, le maximum

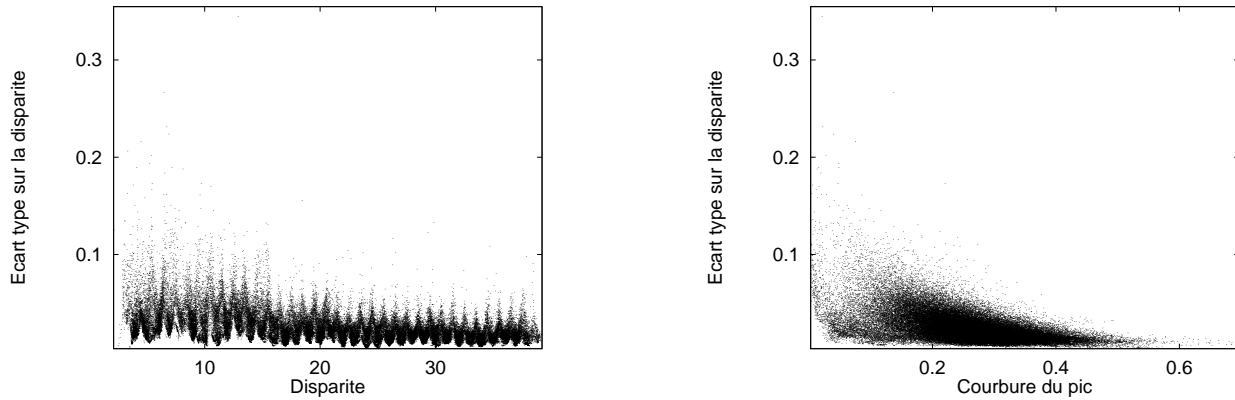


FIG. C.7 – Écarts types mesurés sur l'estimation de la disparité en fonction de la disparité moyenne (gauche), et évolution de ces écarts type en fonction de la courbure mesurée au sommet de la courbe des scores de ressemblance (droite)

réel de corrélation n'a, la plupart du temps, aucune raison de correspondre à une disparité entière. Nous utilisons donc une interpolation parabolique, en fonction du maximum obtenu pour une disparité entière, et les deux disparités voisines. Le choix de la fonction d'interpolation est cependant — *a priori* — libre. Une étude précise du comportement de la fonction de corrélation pour des disparités non-entière permettrait de préciser le choix de la fonction d'interpolation, et donc d'obtenir un modèle pour la précision des points 3-D.

- Un autre problème connu est celui des discontinuités de profondeur (« occluding contours » en anglais [Sara 1997]). Il a été résolu grâce à la fonction d'auto-corrélation lors de la sélection des pixels pour l'odométrie optique, mais en revanche rien n'est fait lors du calcul d'une image 3-D pleine. Les contributions que l'on trouve pour résoudre ce problème mettent souvent en œuvre des méthodes très lourdes, et en tout cas incompatibles avec un robot en mouvement et une fréquence d'acquisition élevée.



FIG. C.8 – Erreurs de l'algorithme de stéréo-vision le long des lignes de gradient de profondeur. L'image de gauche est un grossissement d'une des images originales de la paire stéréoscopique. Sur l'image de disparité (à droite), on voit que les disparités estimées juste à l'extérieur de l'objet correspondent à la profondeur de l'objet.

Bibliographie

- [AB 1998] Special issue on biologically inspired models of navigation. MIT Press, Winter/Spring 1998.
- [Alami 1998a] R. Alami, R. Chatila, S. Fleury, M. Ghallab & F. Ingrand. *An architecture for autonomy*. International Journal of Robotics Research, vol. 17, n° 4, pages 315–337, avril 1998.
- [Alami 1998b] R. Alami, S. Fleury, M. Herrb, F. Ingrand & F. Robert. *Multi Robot Cooperation in the Martha Project*. IEEE Robotics and Automation Magazine, vol. 5, n° 1, mars 1998.
- [Alami 2000] R. Alami, R. Chatila, S. Fleury, M. Herrb, F. Ingrand, M. Khatib, B. Morisset, P. Moutarlier & T. Simeon. *Around the Lab in 40 days...* Dans International Conference on Robotics and Automation, pages 88–94. San Francisco, CA (USA), avril 2000. IEEE.
- [Aloimonos 1997] Y. Aloimonos, editeur. *Visual navigation : from biological systems to unmanned ground vehicles*. Lawrence Erlbaum Associates, 1997.
- [Andrade-Barosso 1998] G. Andrade-Barosso, F. Ben-Amar, P. Bidaud & R. Chatila. *Modeling robot-soil interaction for planetary rover motion control*. Dans International Conference on Intelligent Robots and Systems, pages 576–581. Victoria (Canada), octobre 1998. IEEE.
- [Andrade-Barosso 2000] G. E. Andrade-Barosso. *Modélisation et adaptation du mouvement de robots tout-terrain*. Thèse de Doctorat, Université Paris VI, septembre 2000.
- [Arun 1987] K. S. Arun, T. S. Huang & S. D. Blostein. *Least-Square Fitting of Two 3-D Point Sets*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, n° 5, pages 698–700, septembre 1987.
- [Asada 1988] M. Asada. *Building a 3D World Model for a Mobile Robot from Sensory Data*. Dans International Conference on Robotics

- and Automation, pages 918–923. Philadelphia, PA (USA), 1988. IEEE.
- [Ayala-Ramirez 1998] V. Ayala-Ramirez, R. Murrieta-Cid & M. Briot. *A Fourier-transform based method for estimation of 2-D translation and rotation : an application to outdoor mobile robotics*. Dans 6th Symposium on Intelligent Robotic Systems. Edinburgh (Great Britain), juillet 1998.
- [Bajcsy 1988] R. Bajcsy. *Active Perception*. Proceedings of the IEEE, vol. 76, n° 8, août 1988.
- [Bares 1989] J Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons & W. Witthacker. *Ambler : An Autonomous Rover for Planetary Exploration*. IEEE Computer, pages 18–26, juin 1989.
- [Barshan 1995] B. Barshan & H. Durrant-Whyte. *Inertial Navigation Systems for Mobile Robots*. IEEE Transactions on Robotics and Automation, vol. 11, n° 3, pages 328–342, 1995.
- [Betge-Brezetz 1995] S. Betge-Brezetz, R. Chatila & M.Devy. *Object-Based Modelling and Localization in Natural Environments*. Dans International Conference on Robotics and Automation, pages 2920–2927. Nagoya (Japan), mai 1995. IEEE.
- [Betge-Brezetz 1996] S. Betge-Brezetz, P. Hebert, R. Chatila & M.Devy. *Uncertain map making in natural environments*. Dans International Conference on Robotics and Automation, pages 1048–1053. Minneapolis, MN (USA), avril 1996. IEEE.
- [Boissier 1995] L. Boissier & L. Marechal. *Rover Demonstrator for Moon Exploration*. Autonomous Robots Journal, vol. 2, n° 4, pages 353–362, 1995.
- [Bonnafous 2001] D. Bonnafous, S. Lacroix & T. Siméon. *Motion generation for a rover on rough terrains*. Soumis dans International Conference on Intelligent Robotics and Systems. Maui, HI (USA), octobre 2001. IEEE.
- [Bonnifait 1997] P. Bonnifait & G. Garcia. *6 DOF Dynamic Localization of a Mobile Robot using Odometry, Inclinometers and a Goniometric System*. Dans Fifth International Symposium on Robot Control, pages 173–178. Nantes (France), septembre 1997.
- [Borelly 1998] J.J. Borelly, E. Coste-Manière, B. Espiau, K. Kapellos, R. Pissard-Gibollet & D. Simon. *The Orccad Architecture*. International Journal of Robotics Research, vol. 17, n° 4, pages 338–359, avril 1998.
- [Borenstein 1995] J. Borenstein & L. Feng. *Correction of Systematic Odometry Errors in Mobile Robots*. Dans International Conference on Robotics and Automation, pages 569–574. Nagoya (Japan), mai 1995. IEEE.
- [Borenstein 1996] J. Borenstein, H. R. Everett & L. Feng. *Navigating mobile robots : Sensors and techniques*. A. K. Peters, Ltd., 1996.
- [Brumitt 1998] B. Brumitt & M. Hebert. *Experiments in Autonomous Driving with Concurrent Goals and Multiple Vehicles*. Dans International Conference on Robotics and Automation, pages 1895–1902. Leuven (Belgium), mai 1998. IEEE.

- [Bulata 1996] H. Bulata. *Modélisation d'un environnement structuré et localisation sur amers pour la navigation d'un robot mobile autonome*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, mai 1996.
- [Carre 1998] F. Carre, L. Gallo, B. Mazar, F. Megel & B. Serra. *MONAI : An autonomous Navigation System for Mobile Robot*. Dans International Conference on Robotics and Automation. Leuven (Belgium), mai 1998. IEEE.
- [Chateau 2000] T. Chateau, L. Trassoudaine, F. Collange, C. Debain & P. Botton. *Fusion d'informations incertaines : application d'aide au guidage d'engins agricole*. *Traitement du Signal*, vol. 17, n° 3, pages 249–262, 2000.
- [Chatila 1985] R. Chatila & J-P. Laumond. *Position Referencing and Consistent World Modeling for Mobile Robots*. Dans International Conference on Robotics and Automation, pages 138–145. St Louis, MI (USA), 1985. IEEE.
- [Chatila 1993] R. Chatila, R. Alami, S. Lacroix, J. Perret & C. Proust. *Planet Exploration by robots : from Mission Planning to Autonomous Navigation*. Dans 6th International Conference on Advanced Robotics. Tokyo (Japan), novembre 1993.
- [Chatila 1995] R. Chatila, S. Lacroix, T. Siméon & M. Herrb. *Planetary Exploration by a Mobile Robot : Mission Teleprogramming and Autonomous Navigation*. *Autonomous Robots Journal*, vol. 2, n° 4, pages 333–344, 1995.
- [Chatila 1999] R. Chatila, G. Andrade, S. Lacroix & A. Mallet. *Motion Control for a Planetary Rover*. Dans 2nd International Conference on Field and Service Robotics. Pittsburgh, PA (USA), août 1999.
- [Chaudhuri 1991] S. Chaudhuri & S. Chatterjee. *Performance Analysis of Total Least Squares Methods in Three-Dimensional Motion Estimation*. *IEEE Transactions on Robotics and Automation*, vol. 7, n° 5, pages 707–714, octobre 1991.
- [Cozman 1997] F. Cozman & E. Krotkov. *Automatic Mountain Detection and Pose Estimation for Teleoperation of Lunar Rovers*. Dans 5th International Symposium on Experimental Robotics, pages 164–172. Barcelona (Spain), juin 1997.
- [Cozman 1998] F. Cozman, E. Krotkov & C. Guestrin. *Outdoor Visual Position Estimation for Planetary Rovers*. Dans International Conference on Robotics and Automation. Leuven (Belgium), mai 1998. IEEE.
- [Crowley 1985] J. L. Crowley. *Navigation for an Intelligent mobile robot*. *IEEE Transactions on Robotics and Automation*, vol. 1, n° 1, pages 31–41, mars 1985.
- [Daily 1988] M. Daily, J. Harris, D. Kreiskey, K. Olion, D. Payton, K. Reseir, J. Rosenblatt, D. Tseng & V. Wong. *Autonomous Cross-Country Navigation With the ALV*. Dans International Conference on Robotics and Automation, pages 718–726. Philadelphia, PA (USA), 1988. IEEE.
- [Deans 2000] M. Deans & M. Herbert. *Experimental Comparison of Techniques for Localization and Mapping using a Bearings Only Sensor*. Dans

- 7th International Symposium on Experimental Robotics. Honolulu, HI (USA), 2000.
- [DeBolt 1997] C. DeBolt, C. O'Donnell, C. Freed & T. Nguyen. *The BUGS 'Basic UXO Gathering System' Project for UXO Clearance and Mine Countermeasures*. Dans International Conference on Robotics and Automation, pages 329–334. Albuquerque, NM (USA), avril 1997. IEEE.
- [Delingette 1993] H. Delingette, M. Hebert & K. Ikeuchi. *A Spherical Representation for the Recognition of Curved Objects*. Dans International Conference on Computer Vision, pages 103–112, mai 1993.
- [Demining 1998] International Advanced Robotics Programme. First Workshop on Robotics for Humantarian Demining, septembre 1998.
- [Devy 1994] M. Devy, R. Chatila, P. Fillatreau, S. Lacroix & F. Nashashibi. *On Autonomous Navigation in a Natural Environment*. jras, vol. 16, pages 5–16, 1994.
- [Devy 1998] M. Devy & C. Parra. *3D scene modelling and curve-based localization in natural environments*. Dans International Conference on Robotics and Automation, pages 3091–3096. Leuven (Belgium), mai 1998. IEEE.
- [Devy 1999] M. Devy & C. Parra. *Topological modelling for outdoor mobile robots*. Dans 7th Symposium on Intelligent Robotic Systems, pages 99–108. Coimbra (Portugal), juillet 1999.
- [Dissanayake 1999] M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark & M. Csorba. *A Solution to the Simultaneous Localization and Map Building (SLAM) Problem*. Rapport technique n° ACFR-TR-01-1999, Australian Centre for Field Robotics, mars 1999.
- [Dudek 1993] G. Dudek, M. Jenkin, E. Miliot & D. Wilkes. *Map Validation and Self-Localisation in a Graph-Like World*. Dans 13th International Joint Conference on Artificial Intelligence. Chambéry (France), 1993.
- [Everett 1995] H.R. Everett. *Sensors for mobile robots*. A K Peters, 1995.
- [Faugeras 1993] O. Faugeras, T. Vieville, E. Theron, J. Vuillemin, B. Hotz, Z. Zhang, L. Moll, P. Bertin, H. Mathieu, P. Fua, G. Berry & C. Proy. *Real-time correlation-based stereo : algorithm, implementations and application*. Rapport de recherche n° 2013, INRIA, août 1993.
- [Ferrand 1991] A. Ferrand. *Conception et mise en œuvre d'un système de capteurs proprioceptifs destiné à la localisation relative des robots mobiles*. Thèse de Doctorat, Institut National des Sciences Appliquées, Toulouse, mai 1991.
- [Fillatreau 1993] P. Fillatreau & M. Devy. *Localization of an autonomous mobile robot from 3D depth images using heterogeneous features*. Dans International Conference on Intelligent Robots and Systems, pages 1881–1888. Yokohama (Japan), juillet 1993. IEEE.
- [Fleury 1996] S. Fleury. *Architecture de contrôle distribuée pour robot mobile autonome : principes, conception et applications*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, février 1996.

- [Fontaine 2000] B. Fontaine, D. Termont, L. Steinike, M. Pollefeys, M. Vergauwen, R. Moreas, F. Xu, K. Landzettel, M. Steinmetz, B. Brunner, H. Michaelis, T. Behnke, R. Dewuqker, P. Degezelle, R. Bertrand & G. Vinsentin. *Autonomous Operation of a Micro-Rover for Geo-Science on Mars*. Dans 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk (The Netherlands), décembre 2000.
- [Fox 1998] D. Fox, W. Burgard & S. Thrun. *Active Markov Localization for Mobile Robots*. Robotics and Autonomous Systems, vol. 25, n° 3/4, pages 195–207, 1998.
- [Fox 1999] D. Fox, W. Burgard & S. Thrun. *Markov Localization for Mobile Robotics in Dynamic Environments*. Journal of Artificial Intelligence Research, pages 391–427, novembre 1999.
- [Fuke 1996] Y. Fuke & E. Krotkov. *Dead Reckoning for a Lunar Rover on Uneven Terrain*. Dans International Conference on Robotics and Automation, pages 411–416. Minneapolis, MN (USA), avril 1996. IEEE.
- [Gat 1990] E. Gat, M.G. Slack, D.P. Miller & R.J. Firby. *Path Planning and Execution Monitoring for a Planetary Rover*. Dans International Conference on Robotics and Automation, pages 20–25. Cincinnati, OH (USA), 1990. IEEE.
- [Gat 1994] E. Gat, R. Desai, R. Ivlev, J. Loch & D.P. Miller. *Behavior Control for Robotic Exploration of Planetary Surfaces*. IEEE Transactions on Robotics and Automation, vol. 10, n° 4, pages 490–503, août 1994.
- [Gennery 1989] D. B. Gennery. *Visual Terrain Matching for a Mars Rover*. Dans International Conference on Computer Vision and Pattern Recognition, pages 483–491. San Diego, CA (USA), 1989. IEEE.
- [Giralt 1992] G. Giralt & L. Boissier. *The French Planetary Rover VAP : Concept and Current Developments*. Dans International Conference on Intelligent Robots and Systems. Raleigh, NC (USA), juillet 1992. IEEE.
- [Gonzales 2000] J. Gonzales & S. Lacroix. *Landmark Recognition to Localize a Cross-Country Mobile Robot*. Dans International Symposium on Robotics and Automation. Monterey, NL (Mexico), novembre 2000.
- [Green 1994] D.N. Green, J.Z. Sasiadek & G.S. Vukovich. *Path Tracking, Obstacle Avoidance and Position Estimation by an Autonomous, Wheeled Planetary Rover*. Dans International Conference on Robotics and Automation, pages 1300–1305. San Diego, CA (USA), mai 1994. IEEE.
- [Gutmann 1998] J.-S. Gutmann, W. Burgard, D. Fox & K. Konolige. *An Experimental Comparison of Localization Methods*. Dans International Conference on Intelligent Robots and Systems, pages 736–743. Victoria (Canada), octobre 1998. IEEE.
- [Haddad 1998a] H Haddad. *Contrôle par vision du mouvement d'un orbot mobile en environnement naturel*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, novembre 1998.

- [Haddad 1998b] H. Haddad, M. Khatib, S. Lacroix & R. Chatila. *Reactive Navigation in Outdoor Environments using Potential Fields*. Dans International Conference on Robotics and Automation, pages 1232–1237. Leuven (Belgium), mai 1998. IEEE.
- [Hait 1997] A. Hait, T. Simeon & M. Taix. *A landmark-based motion planner for rough terrain navigation*. Dans 5th International Symposium on Experimental Robotics. Barcelona (Spain), juin 1997.
- [Hait 1999] A. Hait, T. Simeon & M. Taix. *Robust motion planning for rough terrain navigation*. Dans International Conference on Intelligent Robotics and Systems, pages 11–16. Kyongju (Korea), octobre 1999. IEEE.
- [Haralick 1989] R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaiday & M. B. Kim. *Pose Estimation from Corresponding Point Data*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, n° 6, pages 1426–1446, novembre 1989.
- [Hebert 1989] M. Hebert, C. Caillas, E. Krotkov, I. S. Kweon & T. Kanade. *Terrain Mapping for a Roving Planetary Explorer*. Dans International Conference on Robotics and Automation, pages 997–1002. Scottsdale, AZ (USA), 1989. IEEE.
- [Hebert 1992] M. Hebert. *3D Landmark Recognition from Range Images*. Dans Conference on Computer Vision and Pattern Recognition, pages 360–365. Champaign, IL (USA), 1992. IEEE.
- [Herrmann 1999] J.M. Herrmann, K. Pawelsik & T. Geisel. *Self-Localization of Autonomous Robots by Hidden Representations*. Autonomous Robots Journal, vol. 7, n° 1, pages 31–40, juillet 1999.
- [Higuchi 1995] K. Higuchi, M. Hebert & K. Ikeuchi. *Building 3-D Models from Unregistered Ranges Images*. CVGIP-GMIP, vol. 57, n° 4, pages 315–333, juillet 1995.
- [Huang 1994] T. S. Huang & A. N. Netravali. *Motion and Structure from Feature Correspondances : A Review*. Proceedings of the IEEE, vol. 82, n° 2, pages 252–258, février 1994.
- [Huber 1999] D. F. Huber & M. Hebert. *A New Approach to 3-D Terrain Mapping*. Dans International Conference on Intelligent Robotics and Systems, pages 1121–1127. Kyongju (Korea), octobre 1999. IEEE.
- [Hygounenc 2001] E. Hygounenc, P. Soueres & S. Lacroix. *Developments on autonomous airship control at LAAS/CNRS*. Dans 14th AIAA Lighter-Than-Air Systems Convention and Exhibition. Akron, Ohio (USA), juillet 2001.
- [IROS 2000] International Conference on Intelligent Robotics and Systems, Takamatsu (Japon), novembre 2000. IEEE.
- [Jung 2001] I-K. Jung & S. Lacroix. *A robust Interest Point Matching Algorithm*. Dans International Conference on Computer Vision. Vancouver (Canada), juillet 2001.
- [Kemurdjian 1992] A. Kemurdjian, V. Gromov, V. Mishkinyuk, V. Kucherenko & P. Sologub. *Small Marsokhod Configuration*. Dans International Conference on Robotics and Automation, pages 165–168. Nice (France), mai 1992. IEEE.

- [Kieffer 2000] M. Kieffer, L. Jaulin, E. Walter & D. Meizel. *Localisation et suivi robuste d'un robot mobile grâce à l'analyse par intervalles*. *Traitement du Signal*, vol. 17, n° 3, pages 207–219, 2000.
- [Krotkov 1992] E. Krotkov & R. Simmons. *Performance of a Six-Legged Planetary Rover : Power, Positioning and Autonomous Walking*. Dans *International Conference on Robotics and Automation*, pages 169–174. Nice (France), mai 1992. IEEE.
- [Krotkov 1995] E. Krotkov, M. Hebert & R. Simmons. *Stereo Perception and Dead Reckoning for a Prototype Lunar Rover*. *Autonomous Robots Journal*, vol. 2, n° 4, pages 313–331, décembre 1995.
- [Krotkov 1999] E. Krotkov & J. Bitch. *The Defense Advanced Research Projects Agency (DARPA) Tactical Mobile Robot Program*. *International Journal of Robotics Research*, vol. 18, n° 7, pages 769–776, 1999.
- [Kubota 1998] T. Kubota, I. Natakani & Y. Kuroda. *Izu-Ohshima Field Tests for Autonomous Planetary Rover*. Dans *International Conference on Intelligent Robots and Systems*, pages 588–593. Victoria (Canada), octobre 1998. IEEE.
- [Kuipers 1988] B. J. Kuipers & Y. T. Byun. *A Robust, Qualitative Method for Robot Spatial Learning*. Dans *7th National Conference on Artificial Intelligence*, pages 774–779. Saint Paul, MN (USA), 1988. Uni. of Texas at Austin.
- [Kweon 1991] I.S. Kweon & T. Kanade. *Extracting Topographic Features for Outdoor Mobile Robots*. Dans *International Conference on Robotics and Automation*, pages 1992–1997. Sacramento, CA (USA), avril 1991. IEEE.
- [Kweon 1992] I. S. Kweon & T. Kanade. *High-Resolution Terrain Map from Multiple Sensor Data*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, n° 2, pages 278–292, février 1992.
- [Lacroix 1994] S. Lacroix, R. Chatila, S. Fleury, M. Herrb & T. Siméon. *Autonomous Navigation in Outdoor Environment : Adaptive Approach and Experiments*. Dans *International Conference on Robotics and Automation*. San Diego, CA (USA), mai 1994. IEEE.
- [Lacroix 1995a] S. Lacroix. *Stratégies de perception et de déplacement pour la navigation d'un robot mobile autonome en environnement naturel*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, avril 1995.
- [Lacroix 1995b] S. Lacroix & R. Chatila. *Motion and Perception Strategies for Outdoor Mobile Robot Navigation in Unknown Environments*. Dans *4th International Symposium on Experimental Robotics*. Stanford, CA (USA), juillet 1995.
- [Lacroix 1997] S. Lacroix & A. Mallet. *Fusion de Cartes de Traversabilité pour la Détection d'Obstacles lors du Déplacement d'un Robot Mobile*. *Contrat Aérospatiale n° 4500003199*, LAAS-CNRS, décembre 1997.
- [Lacroix 1999] S. Lacroix, A. Mallet & R. Chatila. *Rover self localization in planetary-like environments*. Dans *5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*. Noordwijk (The Netherlands), juin 1999.

- [Lacroix 2000a] S. Lacroix. *Toward autonomous airships : research and developments at LAAS/CNRS*. Dans 3rd International Airship Convention and Exhibition. Friedrichshafen (Germany), juillet 2000.
- [Lacroix 2000b] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb & R. Chatila. *Autonomous Rover Navigation on Unknown Terrains : Functions and Integration*. Dans 7th International Symposium on Experimental Robotics. Honolulu, HI (USA), 2000.
- [Lacroix 2000c] S. Lacroix, A. Mallet, D. Bonnafous, S. Fleury, M. Di Marco & R. Chatila. *Autonomous Long Range Navigation in Planetary-Like Environments*. Dans 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk (The Netherlands), décembre 2000.
- [Lacroix 2001a] S. Lacroix, I-K. Jung & A. Mallet. *Digital Elevation Map Building with Low Altitude Stereo Imagery*. Dans 9th Symposium on Intelligent Robotic Systems. Toulouse (France), juillet 2001.
- [Lacroix 2001b] S. Lacroix, I-K. Jung, A. Mallet & R. Chatila. *Towards cooperative air/ground robotics : issues related to environment modeling*. Dans 10th International Conference on Advanced Robotics. Budapest (Hungary), août 2001.
- [Lacroix 2001c] S. Lacroix & A. Mallet. *Integration of concurrent localization algorithms for a planetary rover*. Dans 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space, juin 2001.
- [Lacroix 2001d] S. Lacroix, A. Mallet, I.K. Jung & G. Scot. *Estimation visuelle du mouvement d'un rover*. Commande CNES n° 713/2/00/0074, LAAS-CNRS, janvier 2001.
- [Lacroix 2002] S. Lacroix, R. Chatila & A. Mallet. Images de profondeur, chapitre Images de profondeur et navigation autonome en environnement naturel, pages 215–242. Hermes, 2002.
- [Lasserre 1995] P. Lasserre & P. Grandjean. *Stereo vision improvements*. Dans 7th International Conference on Advanced Robotics, pages 679–685. Guixols (Spain), septembre 1995.
- [Latombe 1991] J. C. Latombe, A. Lazanas & S. Shekhar. *Robot Motion Planning with Uncertainties in Control and Sensing*. Artificial Intelligence, vol. 52, n° 1, 1991.
- [Laurette 1991] R. Laurette, A. De Saint-Vincent, R. Alami, R. Chatila & V. Perébaskine. *Supervision and control of the AMR intervention robot*. Dans 5th International Conference on Advanced Robotics. Pisa (Italy), juin 1991.
- [Lazanas 1992] A. Lazanas & J.C. Latombe. *Landmark-Based Robot Navigation*. Dans 10th National Conference on Artificial Intelligence, pages 816–822. San Jose, CA (USA), 1992.
- [Leonard 1991] J. J. Leonard & H. F. Durrant-Whyte. *Simultaneous map building and localization for an autonomous mobile robot*. Dans International Workshop on Intelligent Robots and Systems. Osaka (Japan), novembre 1991. IEEE.

- [Leonard 1999] J. J. Leonard & H. J. S. Feder. *Decoupled Stochastic Mapping*. Technical Memorandum n° 99-1, MIT Marine Robotics Laboratory, mars 1999.
- [Levitt 1990] T. S. Levitt & D. T. Lawton. *Qualitative Navigation for Mobile Robots*. Artificial Intelligence, vol. 44, n° 3, pages 305–360, août 1990.
- [Linden 1986] T. A. Linden, J. P. Marsh & D. L. Dove. *Architecture and Early Experience with Planning for the ALV*. Dans International Conference on Robotics and Automation, pages 2035–2042. San Francisco, CA (USA), 1986. IEEE.
- [Lucido 1998] L. Lucido & R. Deriche. *Mise en correspondance de profils bathymétriques pour la navigation des engins sous-marins*. Dans Reconnaissance des Formes et Intelligence Artificielle, vol. III, pages 293–301. Clermont-Ferrand (France), janvier 1998.
- [Maaref 1999] H. Maaref, M. Oussalah & C. Barret. *Fusion de données capteurs en vue de la localisation absolue d'un robot mobile par une méthode basée sur la théorie des possibilités. Comparaison avec le filtre de Kalman*. Traitement du Signal, vol. 16, n° 5, pages 345–359, 1999.
- [Maeyama 1997] S. Maeyama, A. Ohya & S. Yuta. *Long Distance outdoor Navigation of an Autonomous Mobile Robot by Playback of Perceived Route Map*. Dans 5th International Symposium on Experimental Robotics, pages 141–150. Barcelona (Spain), juin 1997.
- [Majumder 2000] S. Majumder, J. Rosenblatt, S. Scheduling & H. Durrant-Whyte. *Map Building and Location for Underwater Navigation*. Dans 7th International Symposium on Experimental Robotics. Honolulu, HI (USA), 2000.
- [Mallet 1997] A. Mallet. *Modélisation d'Amers pour des Robots Mobile Autonomes en vue de la Localisation en environnements non-structurés*. Rapport de Stage de DEA, LAAS-CNRS, juin 1997.
- [Mallet 1998] A. Mallet & S. Lacroix. *Toward real-time 2D localization in outdoor environments*. Dans International Conference on Robotics and Automation, pages 2827–2832. Leuven (Belgium), mai 1998. IEEE.
- [Mallet 2000] A. Mallet, S. Lacroix & L. Gallo. *Position Estimation in Outdoor Environments using Pixel Tracking and Stereovision*. Dans International Conference on Robotics and Automation, pages 3519–3524. San Francisco, CA (USA), avril 2000. IEEE.
- [Marco 2000] M. Di Marco, A. Garulli, S. Lacroix & A. Vicino. *A Set Theoretic Approach to the Simultaneous Localization and Map Building Problem*. Dans 39th Conference on Decision and Control. Sydney (Australia), décembre 2000. IEEE.
- [Martin 1995] J. Martin & J. L. Crowley. *Comparison of Correlation Techniques*. Dans International Conference on Intelligent Autonomous Systems, pages 86–93. Karlsruhe (West Germany), mars 1995.
- [Matsui 2000] T. Matsui, H. Asoh & Thompson. *Mobile Robot Localization Using Circular Correlations of Panoramic Images*. Dans International

- Conference on Intelligent Robotics and Systems [IROS 2000], pages 269–174.
- [Matsumoto 1999] Y. Matsumoto, K. Ikeda, M. Inaba & H. Inoue. *Exploration and Map Acquisition for View-Based Navigation in Corridor Environment*. Dans 2nd International Conference on Field and Service Robotics, pages 341–346. Pittsburgh, PA (USA), août 1999. Nara Institute of Science and Technology.
- [Matthies 1992] L. Matthies. *Stereo Vision for Planetary Rovers : Stochastic Modeling to Near Real Time Implementation*. International Journal of Computer Vision, vol. 8, n° 1, pages 71–91, 1992.
- [Matthies 1995] L. Matthies, A. Kelly & T. Litwin. *Obstacle Detection for Unmanned Ground Vehicles : A Progress Report*. Dans International Symposium of Robotics Research. Munich (Germany), octobre 1995.
- [Matthies 1997] L. Matthies, C. Olson, G. Tharp & S. Laubagh. *Visual Localization Methods for Mars Rovers using Lander, Rover and Descent Imagery*. Dans International Symposium on Artificial Intelligence, Robotics and Automation in Space, pages 413–418. Tokyo (Japan), juillet 1997.
- [Maurette 1998] M. Maurette. *CNES Robotics Program*. Dans 5th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk (The Netherlands), décembre 1998. CNES.
- [McTamaney 1987] L. S. McTamaney. *Real-Time Intelligent Control*. IEEE Expert, vol. 2, n° 4, pages 55–68, 1987.
- [Miller 1989] D. P. Miller, F. J. Atkinson, B. Wilcox & A. H. Mishkin. *Autonomous Navigation and Control of a Mars Rover*. Dans 11th IFAC Symposium on Automatic Control in Aerospace, pages 127–130. Tsukuba (Japan), juillet 1989. JPL.
- [Miyake 1997] N. Miyake, T. Aono, K. Fujii, Y. Matsuda & S. Hatsumoto. *Position Estimation and Path Control of an Autonomous Land Vehicle*. Dans International Conference on Intelligent Robots and Systems, pages 690–696. Grenoble (France), septembre 1997. IEEE.
- [Moorehead 1999] S. Moorehead, R. Simmons, D. Apostolopoulos & W. L. Whittaker. *Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica*. Dans 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space. Noordwijk (The Netherlands), juin 1999.
- [Morrison 1996] J. Morrison & T. Nguyen. *On-Board Software for the Mars Pathfinder Microrover*. Dans 2nd International Conference on Low-Cost Planetary Missions. Laurel, MD (USA), avril 1996. IAA.
- [Moutarlier 1991] P. Moutarlier & R. Chatila. *Incremental free-space modelling from uncertain data by an autonomous mobile robot*. Dans International Workshop on Intelligent Robots and Systems, pages 1052–1058. Osaka (Japan), novembre 1991. IEEE.
- [Murphy 2000] R. R. Murphy. Introduction to AI Robotics, chapitre Localization and Map Making, pages 375–434. The MIT Press, décembre 2000.

- [Murrieta-Cid 1997] R. Murrieta-Cid, P. Lasserre & M. Briot. *Color segmentation in principal regions for natural outdoor scenes*. Dans Workshop on Electronic Control and Measuring Systems. Toulouse (France), juin 1997.
- [Murrieta-Cid 1998] R. Murrieta-Cid, M. Briot & N. Vandapel. *Landmark identification and tracking in natural environment*. Dans International Conference on Intelligent Robots and Systems, pages 179–184. Victoria (Canada), octobre 1998. IEEE.
- [Nagatani 1998] K. Nagatani, H. Choset & S. Thrun. *Towards Exact Localization without Explicit Localization with the General Voronoi Graph*. Dans International Conference on Robotics and Automation, pages 342–348. Leuven (Belgium), mai 1998. IEEE.
- [Nebot 1999] E. M. Nebot & H. F. Durrant-Whyte. *High Integrity Navigation Architecture for Outdoor Autonomous Vehicles*. Robotics and Autonomous Systems, vol. 26, n° 2-3, pages 81–97, 1999.
- [Novara 1998] M. Novara. *Scenarios for Automation and Robotics in Long-Term Space Exploration and Resources Exploitation*. Dans 5th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk (The Netherlands), décembre 1998. ESA - System studies division.
- [Olin 1991] K. E. Olin & D. T. Tseng. *Autonomous Cross-Country Navigation*. IEEE Expert, vol. 6, n° 4, pages 16–30, août 1991.
- [Olson 1997] C. Olson. *Mobile Robot Self-Localization by Iconic Matching of Range Maps*. Dans 8th International Conference on Advanced Robotics, pages 447–452. Moterey, CA (USA), juillet 1997.
- [Olson 1998] C. F. Olson & L. H. Matthies. *Maximum Likelihood Rover Localization by MATCHING Range Maps*. Dans International Conference on Robotics and Automation, pages 272–277. Leuven (Belgium), mai 1998. IEEE.
- [Olson 2000a] C. Olson, L. Matthies, M. Schoppers & M. Maimone. *Robust Stereo Ego-Motion for Long Distance Navigation*. Dans Conference on Computer Vision and Pattern Recognition. Hilton Head Island, SC (USA), juin 2000. IEEE.
- [Olson 2000b] C. F. Olson. *Landmark Selection for Terrain Matching*. Dans International Conference on Robotics and Automation, pages 1447–1452. San Francisco, CA (USA), avril 2000. IEEE.
- [Parra 1997] C. Parra & M. Devy. *Modelling and self-localization in natural environments*. Dans 13th International Conference on CAD/CAM Robotics and Factories of the Future. Pereira (Colombie), décembre 1997. IEEE.
- [Pleska 2000] E.M.A. Pleska, J.F. Kieffer & P. Bouniol. *Les senseurs inertiels du XXIème Siècle*. Revue Scientifique et Technique de la Défense, n° 49, pages 115–132, juillet 2000.
- [Putz 2000] P. Putz. *The ESA Research and Development Programme in Space Automation and Robotics*. Dans 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk (The Netherlands), décembre 2000.

- [Rastel 1998] L. Rastel & M. Delpech. *Rover Continuous Path Planning using Merged Perception*. Dans International Conference on Robotics and Automation. Leuven (Belgium), mai 1998. IEEE.
- [Roumeliotis 1999] S. I. Roumeliotis & G. A. Bekey. *3D Localization for a Mars Rover Prototype*. Dans 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space, pages 441–448. Noordwijk (The Netherlands), juin 1999.
- [Sara 1997] R. Sara & R. Bajcsy. *On occluding contour artifacts in stereo vision*. Dans Conference on Computer Vision and Pattern Recognition, pages 852–857. San Juan (Porto Rico), juin 1997. IEEE.
- [Seiz 2000] M. Seiz, P. Jensfelt & H. Christensen. *Active Exploration for Feature Based Global Localization*. Dans International Conference on Intelligent Robotics and Systems [IROS 2000], pages 281–287.
- [Simeon 1993] T. Simeon & B. Dacre-Wright. *A practical motion planner for all-terrain mobile robots*. Dans International Conference on Intelligent Robots and Systems, pages 26–30. Yokohama (Japan), juillet 1993. IEEE.
- [Simhon 1998] S. Simhon & G. Dudek. *A Global Topological Map formed by Local Metric Maps*. Dans International Conference on Intelligent Robots and Systems. Victoria (Canada), octobre 1998. IEEE.
- [Simmons 1991] R. Simmons & E. Krotkov. *An Integrated Walking System for the Ambler Planetary Rover*. Dans International Conference on Robotics and Automation, pages 2086–2091. Sacramento, CA (USA), avril 1991. IEEE.
- [Simmons 1994] R. Simmons. *Task-Level Control for Autonomous Robots*. Dans Intelligent Robotics in Field, Factory, Service, and Space, pages 275–281, mars 1994.
- [Simmons 1995] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krisnaswamy, Y. Shinoda, W. Whittaker & P. Klarer. *Experience with Rover Navigation for Lunar-Like Terrains*. Dans International Conference on Intelligent Robots and Systems. Pittsburgh, PA (USA), 1995. IEEE.
- [Stentz 1999] A. Stentz, M. Ollis, S. Sheding, H. Herman, C. Fromme, J. Pedersen, Y. Hegadorn, R. McCall, J. Bares & R. Moore. *Position Measurements for Automated Mining Machinery*. Dans 2nd International Conference on Field and Service Robotics, pages 299–304. Pittsburgh, PA (USA), août 1999.
- [Stone 1996] H. W. Stone. *Mars Pathfinder Microrover : A Small Low-Cost, Low-Power Spacecraft*. Dans AIAA Forum on Advanced Developments in Space Robotics, août 1996.
- [Suh 1993] S. Suh, J. Kang, W. Jee, M. Jung & K. Kim. *Estimating ALV Position in Mountainous Area*. Dans International Conference on Intelligent Robots and Systems, pages 2178–2185. Yokohama (Japan), juillet 1993. IEEE.
- [Sukkarieh 1997] S. Sukkarieh, E. Nebot & H. Durrant-Whyte. *The GPS Aiding of INS for Land Vehicle Navigation*. Dans 1st International Conference on Field and Service Robotics. Canberra (Australia), août 1997.

- [Sukkarieh 1999a] S. Sukkarieh, E. Nebot & H. Durrant-Whyte. *A high integrity IMU GPS navigation loop for autonomous land vehicle applications*. IEEE Transactions on Robotics and Automation, vol. 15, n° 3, pages 572–578, juin 1999.
- [Sukkarieh 1999b] S. Sukkarieh, E. Nebot & H. Durrant-Whyte. *A high integrity IMU GPS navigation loop for autonomous land vehicle applications*. IEEE Transactions on Robotics and Automation, vol. 15, n° 3, pages 572–578, juin 1999.
- [Thorpe 1988] C. Thorpe, M. Hebert, T. Kanade & S. Shafer. *Vision and Navigation for the Carnegie-Mellon Navlab*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, n° 3, pages 362–373, mai 1988.
- [Thorpe 1991a] C. Thorpe, M. Hebert, T. Kanade & S. Shafer. *Toward Autonomous Driving : the CMU Navlab. Part I : Perception*. IEEE Expert, vol. 6, n° 4, pages 31–42, août 1991.
- [Thorpe 1991b] C. Thorpe, M. Hebert, T. Kanade & S. Shafer. *Toward Autonomous Driving : the CMU Navlab. Part II : Architecture and Systems*. IEEE Expert, vol. 6, n° 4, pages 45–52, août 1991.
- [Thorpe 1997] C. Thorpe. *Mobile Robots and Smart Cars*. Dans 1st International Conference on Field and Service Robotics. Canberra (Australia), août 1997.
- [Thrun 1998] S. Thrun. *Learning metric-topological maps for indoor mobile robot navigation*. Artificial Intelligence, vol. 99, n° 1, pages 21–71, 1998.
- [Tirumalai 1995] A. P. Tirumalai, B. Schunck & R. C. Jain. *Evidential Reasoning for Building Environment Maps*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, n° 1, pages 10–20, janvier 1995.
- [Vandapel 2000] N. Vandapel, R. Chatila, S.J. Moorehead, M. Hebert, S. Lacroix, M. Devy, D. Apostolopoulos & W. Whittaker. *Evaluation of Computer Vision Algorithms for Autonomous Navigation in Polar Terrains*. Dans 6th International Conference on Intelligent Autonomous Systems. Venice (Italy), juillet 2000.
- [Volpe 1999] R. Volpe. *Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype*. International Journal of Robotics Research, vol. 18, n° 7, pages 669–683, juillet 1999.
- [Washington 1999] R. Washington, K. Golden, J. Bresina, D.E. Smith, C. Anderson & T. Smith. *Autonomous Rovers for Mars Exploration*. Dans Aerospace Conference. Aspen, CO (USA), 1999. IEEE.
- [Weisbin 1992] C.R. Weisbin, M. Montenerlo & W. Whittaker. *Evolving Directions in NASA's Planetary Rover Requirements and Technology*. Dans Missions, Technologies and Design of Planetary Mobile Vehicles. Toulouse (France), septembre 1992.
- [Whittaker 1997] W. R. Whittaker, D. Bapna, M. W. Maimone & E. Rollins. *Atacama Desert Trek : A Planetary Analog Field Experiment*. Dans International Symposium on Artificial Intelligence, Robotics and Automation in Space. Tokyo (Japan), juillet 1997.

- [Xiong 1997] Y. Xiong & L. Matthies. *Error analysis of a real time stereo system*. Dans Conference on Computer Vision and Pattern Recognition, pages 1087–1093. San Juan (Porto Rico), juin 1997. IEEE.
- [Zabih 1994] R. Zabih & J. Woodfill. *Non-parametric Local Transforms for Computing Visual Correspondence*. Dans 3rd European Conference on Computer Vision. Stockholm (Sweden), mai 1994.
- [Zhang 1991] Z. Zhang & O. D. Faugeras. *Estimation of Displacements from Two 3D Frames Obtained from Stereo*. Rapport de recherche n° 1440, INRIA, juin 1991.
- [Zwynsvoorde 2000] D. Van Zwynsvoorde. *Construction incrémentale de modèles topologiques pour la navigation d'un robot mobile*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, décembre 2000.

