



HAL
open science

Stratégies de perception et de déplacement pour la navigation d'un robot mobile autonome en environnement naturel

Simon Lacroix

► **To cite this version:**

Simon Lacroix. Stratégies de perception et de déplacement pour la navigation d'un robot mobile autonome en environnement naturel. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 1995. Français. NNT: . tel-00146607

HAL Id: tel-00146607

<https://theses.hal.science/tel-00146607>

Submitted on 15 May 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.

Année 1995

Thèse

*Présentée au Laboratoire d'Analyse
et d'Architecture des Systèmes du CNRS*

*en vue de l'obtention du Grade de Docteur
de l'Université Paul Sabatier de Toulouse*

*Spécialité : **Robotique***

par **Simon LACROIX**
Ingénieur de l'Ecole Centrale de Paris

Stratégies de perception et de déplacement pour la navigation d'un robot mobile autonome en environnement naturel

Soutenue le Jeudi 13 Avril 1995 devant le jury :

MM.	Georges	GIRALT	Président
	Raja	CHATILA	Directeur de Thèse
	Michael	BRADY	Rapporteur
	Martial	HEBERT	Rapporteur
	Alain	LIEGEOIS	Rapporteur
	Malik	GHALLAB	Examineur
	Maurice	BRIOT	Examineur
	Jean-Louis	LACOMBE	Examineur

Rapport LAAS N° 95246

Thèse préparée au Laboratoire d'Analyse
et d'Architecture des Systèmes du CNRS
7, Avenue du Colonel Roche
31077 Toulouse Cedex (France)

*“La vérité c’est que le machinisme pose à l’ordre social
actuel un problème qu’il ne pourra pas résoudre et sous
lequel il succombera. Car la machine tend partout à se
substituer à l’homme. Que ferez-vous des salariés ? C’est
dire que la Nation sera obligée de proclamer et d’organiser
le droit au travail.”*

Jean Jaurès

Avant Propos

Ce mémoire est l'aboutissement de plus de quatre années passées dans l'équipe "Robotique et Intelligence Artificielle" du LAAS. Je me souviendrai longtemps de la formidable ambiance qui règne dans cette équipe, ainsi que des diverses personnalités qui la composent. Je suis donc tout d'abord particulièrement reconnaissant à Georges Giralt de m'avoir si promptement téléphoné après avoir reçu une lettre envoyée en Avril 1990, et de m'avoir accueilli au sein de ce groupe dont il était à l'époque le responsable.



L'ensemble des travaux présentés dans ce mémoire ont été réalisés sous la direction de Raja Chatila, dont les orientations, encouragements, conseils (et doutes parfois !) m'ont été extrêmement utiles : tout cela est présent dans chacun des chapitres qui suivent. Merci mille fois Raja, pour toutes ces trois années passées en ta compagnie ! Merci beaucoup à Malik Ghallab aussi, pour m'avoir encadré durant mon DEA, et pour sa patience sans limites.



Je suis également très reconnaissant à Michael Brady, Martial Hebert et Alain Liégeois d'avoir accepté de porter un jugement sur mon manuscrit, ainsi qu'à Maurice Briot et Jean-Louis Lacombe, qui ont euz aussi accepté de me consacrer leur journée du 13 Avril 1995...

Ce travail a pu être réalisé grâce à l'ensemble des personnes de l'équipe RIA et du laboratoire qui font du LAAS un lieu agréable où domine la bonne humeur, et dans lequel on ne compte pas les heures passées. Salut amical aux gourous des machines et aux gourous des robots de RIA ! Salut aux gens du magasin, à celles et ceux de la gestion du personnel, de la cafétéria, de la documentation, de l'édition...

Et je ne peux oublier les copains (mais j'en oublie sûrement), pour les moments passés ensembles, au LAAS comme ailleurs. Salut aux anciens grâce auxquels j'ai beaucoup appris, Philippe, Paul, Pierrick, Vianney, Pichou... Salut à tous ceux avec lesquels j'ai partagé toutes ces années, particulièrement à Sara, Matthieu, Fred, Christophe... Salut à l'ensemble des thésards et stagiaires de DEA (pas loin d'une quarantaine !) que j'ai pu rencontrer. Salut aussi aux néo-Toulousains de Miramar et des Rosiers !

Salut spécial plein de reconnaissance à Jackie et Nic pour le coup de pouce final...

Et bien entendu, une pensée toute particulière et bien plus personnelle pour ma famille, et surtout pour Christine, pour sa patience, sa tolérance et bien plus encore...



Juin 1995

Table des matières

Table des matieres	i
Table des figures	iii
Présentation Générale	1
1 Contexte de travail : Navigation autonome en environnement naturel	3
1.1 Un cas d'école : l'exploration planétaire	4
1.2 Les approches et réalisations connues	7
1.3 L'architecture globale proposée au LAAS	9
1.4 Notre approche de la navigation	12
Première Partie : Modélisation de l'environnement	17
2 Introduction	19
2.1 Utilité des représentations	19
2.2 De la nécessité de représentations globales	20
2.3 Quel modèle pour la navigation ?	21
3 Classification de données 3D brutes	25
3.1 Objectifs	25

3.2	Méthodes existantes	26
3.3	Caractéristiques des capteurs 3D	27
3.4	Méthode proposée	35
3.5	Définition et construction des cellules	36
3.6	Classification bayésienne d'images stéréo	42
3.7	Classification d'images laser	51
3.8	Conclusions - Extensions possibles	54
4	Construction d'un modèle topologique global	57
4.1	Fusion de perceptions locales	57
4.2	Structuration : construction d'un graphe de régions	64
4.3	Gestion du Modèle	69
4.4	Vers une modélisation hétérogène	70
	Seconde Partie : Stratégies de Navigation	73
5	Analyse du problème	75
5.1	Présentation	75
5.2	Les actions à déterminer	77
5.3	Éléments de décision	79
6	Approche proposée	83
6.1	Planification d'une tâche de perception : généralités	83
6.2	Détermination d'une tâche de recalage	87
6.3	Évaluation d'une tâche de classification	90

6.4	Détermination d'un chemin	93
6.5	De la difficulté d'une approche exhaustive de la navigation	95
6.6	Une approche heuristique	96
7	Résultats expérimentaux	109
7.1	Le projet EDEN	109
7.2	Les expériences de navigation	113
7.3	Résultats en Simulation	120
8	Conclusion générale	123
A	Quelques projections de l'espace des prototypes	125
	Références bibliographiques	129

Table des figures

1.1	Le robot Adam dans l'environnement d'expérimentation du LAAS	7
1.2	Les principaux composants d'une architecture pour un système robotique d'intervention à distance	9
1.3	Illustration du principe de planification hiérarchique des déplacements	14
1.4	Le rôle du planificateur de chemin (navigation) dans la boucle "Perception-Décision-Action"	16
2.1	La carte de régions est un affinement de la description initiale du tronçon	23
3.1	Le télémètre 2-axes d'Adam	28
3.2	Système de déflexion du faisceau	28
3.3	Mode de balayage non corrélé	29
3.4	Vue perspective d'une image de points 3D acquise en environnement intérieur	30
3.5	Définition des résolutions radiale et angulaire, et résolution radiale sur un sol plan	30
3.6	Image stéréoscopique gauche, et pixels effectivement corrélés	32
3.7	Points 3D issu d'un algorithme de stéréovision	33
3.8	Mode de balayage corrélé	33
3.9	Modèle sténopé d'une caméra	33
3.10	Résolution angulaire d'une caméra, résolution radiale sur un sol plan	34
3.11	Projection orthogonale des points	36
3.12	Discrétisation associée	36
3.13	Définition de la discrétisation adoptée	37
3.14	Définition du repère R_p par rapport aux repères R_m , R_r et R_c	38
3.15	Un arc d'ellipse est approximé par une séquence d'arcs de cercle	38

3.16	Discrétisation de la zone perçue par une caméra avec un roulis de 12°	39
3.17	Comportement de l'attribut "densité relative" en fonction du terrain perçu	40
3.18	Calcul du vecteur normal en un point d'une image laser	41
3.19	Affichage des points d'une cellule et des pixels correspondants pour prototypage	44
3.20	Densité de probabilité de l'attribut "densité relative"	45
3.21	Densité de probabilité de l'attribut "norme des écarts types sur les coordonnées du vecteur normal"	45
3.22	Résultat de la première étape de classification	49
3.23	Amélioration après filtrage et scission	49
3.24	Résultat de la classification de deux autres images	49
3.25	Classification d'une image à trois résolutions différentes	51
3.26	L'algorithme de classification d'une cellule de points laser	52
3.27	Les tests sur les attributs pour classer les cellules	53
3.28	Classification d'une image laser	54
4.1	Deux acquisitions de paires d'images stéréoscopiques	61
4.2	Étiquettes après application de la loi de décision	61
4.3	Altitudes et variances sur l'altitude des images stéréoscopiques classifiées	63
4.4	Altitudes et variances sur l'altitude de deux images laser classifiées	63
4.5	Les pixels contours d'une région à grossir	65
4.6	Résultat du grossissement avec propagation par 4-voisins, 8-voisins, et alternée	65
4.7	Avec un regroupement de régions basé sur un critère d'homogénéité de tous les champs des pixels, certaines régions sont réduites à un pixel	66
4.8	Définition des nœuds et des arcs du graphe	66
4.9	fusion des étiquettes de deux images laser classifiées	67
4.10	Classes de terrain et altitudes d'un modèle issu de la fusion de 8 images laser	67
4.11	Un modèle construit sur la base de 18 perceptions : carte de régions globale et bitmap courant	67
4.12	3000 régions déterminées avec un critère d'homogénéité des pixels strict définissent 24000 nœuds	69

4.13	450 régions déterminées avec un critère d'homogénéité des pixels "tolérant" définissent 5500 nœuds	69
4.14	Les différentes représentations de l'environnement présentes dans le système	71
5.1	Quelle décision prendriez-vous ?	77
5.2	Les entrées et sorties de la fonction navigation	82
6.1	Un modèle de détectabilité idéal en 2D	88
6.2	MNT d'une perception et écarts types sur l'élévation	88
6.3	Comportement de la fonction de recalage pour un amer	89
6.4	Comportement de la fonction de recalage pour trois amers	89
6.5	Test de visibilité d'un amer	90
6.6	Un modèle de terrain avec trois amers : altitudes, étiquettes après grossissement et zones de visibilité des amers	90
6.7	Probabilité d'erreur sur l'étiquetage des images stéréo en fonction de la distance au capteur	91
6.8	Définition du modèle de confiance de l'étiquetage des images laser	92
6.9	Discretisation choisie pour évaluer une tâche de classification	92
6.10	Les trois "meilleures" tâches de classification déterminées en 3 endroits différents	92
6.11	Une classification qui maximise le gain d'information mais est de bien peu d'intérêt pour rallier le but fixé	93
6.12	Dans ces deux configurations particulières, la meilleure décision est de se rendre au point "A" pour percevoir la région "X"	96
6.13	Une approche exhaustive de la navigation	97
6.14	Pondération des coûts nominaux en fonction de la confiance sur l'étiquetage	99
6.15	Chemin trouvé sans considérer les incertitudes sur la modélisation	103
6.16	Chemin trouvé en considérant les incertitudes sur la modélisation	103
6.17	Traverser la zone étiquetée obstacle est moins coûteux que la contourner	103
6.18	Traverser la zone accidentée s'avère moins coûteux que la contourner	103
6.19	Introduction de la derive pour évaluer le coût	103
6.20	Le chemin résultat est dévié vers l'amer	103
6.21	Résultat de la navigation	105

6.22	Les régions correspondantes	105
6.23	Le modèle de terrain effectivement passé au planificateur de trajectoires 2D	106
6.24	Trajectoire planifiée (en gris : les distances aux obstacles)	106
7.1	La procédure PRS correspondant à la tâche ‘‘Aller à [But]’’	111
7.2	La couche fonctionnelle d’Adam : les flux de données	112
7.3	Aperçu de l’allure du site d’expérimentation Geroms	114
7.4	Le robot préfère regarder à droite plutôt que de s’engager entre les deux obstacles	116
7.5	... ce qui semble être un bon choix	116
7.6	... le but est atteint après la sixième perception	116
7.7	Le robot tente de contourner les obstacles par la gauche	117
7.8	Il passe finalement par la droite	117
7.9	Le but final est atteint à la 10ème perception	117
7.10	Caractéristiques du modèle de terrain construit	119
7.11	Performances de l’implémentation de l’algorithme A^*	119
7.12	La baie de San Francisco : Carte d’élévation originale	121
7.13	Modèle construit par le robot : le but est rallié après 29 déplacements	122
7.14	Caractéristiques du modèle de terrain construit	122

Présentation Générale

Une machine que l'on peut qualifier d'intelligente doit être capable d'exécuter de manière *autonome* des tâches (enchaînement d'actions) en *interaction* avec le monde réel. Pour cela, elle doit posséder la capacité de raisonner sur les tâches à accomplir en fonction de la connaissance qu'elle a de son environnement : pour la conception de telles machines, nous sommes convaincus de l'efficacité et de la robustesse d'une approche *délibérative*, dans laquelle les décisions sont prises après prédiction de leurs conséquences et évaluation de leur intérêt.

Ce type d'approche requiert de bonnes *représentations* des capacités d'action de la machine, mais aussi et surtout de l'environnement dans lequel elle évolue. Ces représentations de l'environnement doivent d'une part être adaptées aux tâches à exécuter, et d'autre part être construites de manière *autonome* et *active* : elles sont rarement connues a priori, et nécessitent donc l'acquisition *contrôlée* de données perceptuelles pour leur élaboration. Cette problématique générale de "perception active" [Bajcsy 88] est désormais abordée sous de nombreux aspects différents, et tout particulièrement dans le domaine de la robotique mobile : il s'agit en effet d'un domaine d'application exemplaire des concepts d'autonomie décisionnelle et opérationnelle propres à une machine intelligente.

La *navigation autonome d'un robot* illustre parfaitement ce besoin de chercher activement des informations perceptuelles pertinentes, particulièrement dans un environnement initialement très peu connu. Cette tâche consiste à enchaîner les actions de perception et de déplacement qui permettent d'atteindre un but fixé, et elle requiert de la part du robot :

- la capacité de construire des représentations de l'environnement dans lequel il évolue à partir des données fournies par ses capteurs ;
- la capacité d'analyser ces représentations afin de décider des actions appropriées à sa situation et au but fixé ;
- et la capacité de mettre en œuvre et de contrôler l'exécution des actions décidées.

Les travaux présentés dans ce mémoire traitent des deux premiers points dans le contexte de la navigation en *environnement naturel* : nous allons présenter des algorithmes de traitement des données perçues qui permettent la construction d'une représentation particulière de l'environnement, et une procédure qui permet au robot de planifier ses actions de perception et de déplacement sur la base de cette représentation.

Le premier chapitre précise le contexte de travail dans lequel notre contribution s'insère. Il introduit une stratégie particulière de la navigation en terrain naturel inconnu, qui met en

avant le besoin d'une fonction décisionnelle particulière (le "planificateur de chemin"), dont le rôle est de contrôler les actions de perception et de déplacement du robot.

Le mémoire s'articule ensuite suivant deux parties distinctes :

- La première partie présente la construction de la représentation de l'environnement dédiée au planificateur de chemin. Après une brève introduction concernant les représentations de l'environnement nécessaires à la navigation en général, un chapitre entier est consacré à une méthode originale de classification probabiliste des données tridimensionnelles. Le chapitre suivant présente la manière dont les données ainsi analysées sont fusionnées en un modèle global, et comment ce modèle est structuré en un graphe de connexité de régions afin de permettre la mise en œuvre du planificateur de chemin. Cette partie s'achève par la présentation d'un cadre structurel dans lequel s'insère l'ensemble des représentations nécessaires à la navigation.
- La seconde partie du mémoire est consacrée au planificateur de chemin lui-même. Un chapitre introductif précise les décisions qui sont à sa charge, les différents choix associés, et les informations dont il dispose. Les techniques nécessaires à la détermination des actions de perception et de déplacement sont alors présentées, et une approche exhaustive qui consiste à utiliser toutes les informations disponibles est ébauchée. La difficulté et la complexité algorithmique de cette approche exhaustive nous amènent alors à proposer une approche heuristique qui s'est avérée satisfaisante.

L'ensemble des travaux présentés dans ce manuscrit ont été développés et implémentés avec le souci constant d'être validés par des expérimentations en site réel, étape que nous jugeons indispensable à tout travail de recherche en robotique. Ainsi les travaux présentés sont illustrés par des résultats obtenus avec des données réelles tout au long du mémoire, et un dernier chapitre est consacré à la présentation d'expérimentations réalisées en environnement naturel avec le robot Adam, et de simulations qui permettent d'analyser le comportement de nos méthodes à long terme.

Chapitre 1

Contexte de travail : Navigation autonome en environnement naturel

Les premiers robots mobiles qui se sont aventurés hors des bâtiments ont tout d'abord commencé à évoluer sur des routes. D'un point de vue strictement robotique, une grande partie des problèmes soulevés par ce nouvel environnement se rapprochent de ceux rencontrés dans les bâtiments : les véhicules se déplaçant sur un sol plan, ils n'ont pas besoin d'avoir des capacités de locomotion complexes, et les fonctionnalités de perception consistent essentiellement à estimer les limites de la route (ce qui n'est pas toujours simple, mais est d'autant plus aisé si elle est balisée) et à détecter la présence d'éventuels obstacles. Le problème reste néanmoins très complexe, notamment à cause de contraintes draconiennes de sécurité, particulièrement lorsque plusieurs véhicules sont en circulation, et de la nécessaire rapidité d'exécution du système : pour réaliser des performances satisfaisantes, le véhicule doit être constamment en mouvement, et ne peut donc pas se permettre de se plonger dans une série de considérations diverses à chaque événement détecté. Il doit agir vite et de manière sûre, ce qui favorise pour ces applications des approches "réactives".

Mais il existe aussi de nombreux domaines d'application de la robotique mobile en environnement naturel totalement non-structuré, qui imposent des contraintes fortes sur les capacités de locomotion des véhicules, mais aussi sur leur capacité de percevoir l'environnement et d'y planifier des déplacements sûrs [Giralt 93]. A titre d'exemple, citons les quelques applications suivantes :

- Sécurité civile : il arrive des catastrophes, naturelles ou provoquées par l'homme, qui interdisent toute intervention humaine sur le site où elles ont lieu (incendies, pollutions extrêmement dangereuses. . .). Dans ce cas, un véhicule doté de capacités d'autonomie peut être d'une grande utilité, en luttant notamment contre les causes qui sont à l'origine de la catastrophe, ou en aidant à lutter contre ses conséquences [Lacombe 91, Giralt 91] ;
- Missions d'exploration : il existe encore de nombreuses régions à explorer, particulièrement sous la mer ou sur d'autres planètes [G.Giralt 92], dans lesquelles des con-

traintes diverses, technologiques ou dues à l'environnement, empêchent l'homme de s'aventurer ;

- Exploitation de ressources naturelles : sous la mer et sur d'autres planètes, mais aussi sur terre (forêts, mines, régions de l'antarctique...), certaines ressources pourraient être exploitées par des machines, évitant ainsi à l'homme de s'exposer à des risques, et réduisant par là les coûts d'exploitation ;
- Enfin, nous ne pouvons éviter de citer les applications militaires, qui pour l'instant se restreignent essentiellement à des missions de reconnaissance ;

Le point commun à la plupart de ces applications est qu'elles présentent des contraintes fortes sur l'environnement (l'homme ne peut se rendre sur les sites d'intervention, par impossibilité matérielle ou à cause de trop grands dangers), mais aussi sur les communications entre un opérateur et la machine. Ces contraintes de communication sont dues à une très grande distance entre le site et l'opérateur, à une faible bande passante ou à des coupures fréquentes. Ajoutées au fait que l'environnement d'intervention est souvent très mal connu *a priori*, ces contraintes impliquent que la machine soit dotée d'une *très grande capacité d'autonomie* : elle doit être capable de décider seule de l'ensemble des actions à exécuter (notamment de déplacement et de perception) pour mener à bien une mission particulière, et ne peut compter sur une aide rapide et permanente d'un opérateur. Les approches basées sur les concepts de téléguidage et de téléopération ne sont plus applicables, l'homme ne pouvant intervenir dans la boucle de contrôle du robot qu'à de rares instants pendant lesquels il ne dispose que de peu d'informations concernant le robot et son environnement.

Nous allons préciser dans ce chapitre les contraintes spécifiques à l'exploration planétaire, domaine d'application dans lequel notre contribution s'insère. Nous présenterons alors brièvement les principales contributions connues sur la navigation autonome dans des contextes similaires. Nous décrirons ensuite l'approche de la navigation en terrain naturel développée au sein de notre équipe, ainsi que les difficultés particulières qu'elle soulève, et les problèmes à la résolution desquels ce mémoire est consacré.

1.1 Un cas d'école : l'exploration planétaire

1.1.1 Des objectifs scientifiques

Notre propos n'est pas de développer un argumentaire en faveur de l'exploration des planètes et satellites qui composent le système solaire, mais nous rappelons tout de même quelques objectifs scientifiques majeurs qui justifient les gros investissements humains et financiers nécessaires. Nombreux sont les chercheurs qui se réjouiraient de disposer de données précises et complètes concernant la composition et la structure d'autres planètes, qu'ils soient astrophysiciens, géologues, climatologues, ou même biologistes. Au delà de leur curiosité scientifique, tous ces chercheurs sont convaincus de l'utilité directe de ces explorations et observations pour notre vie quotidienne. Elles permettraient notamment de mieux comprendre les évolutions géologiques et climatiques de notre planète (effet de serre, modifications du climat...), et donc de les prévoir, et par là de s'y préparer.

Mentionnons aussi les possibilités que donnerait une présence permanente de machines sur des planètes : outre le fait que les spécialistes considèrent cette présence comme une étape nécessaire préalable à toute exploration humaine, cela ouvrirait de nouvelles possibilités d'observation de l'espace, et permettrait aussi l'exploitation de leurs ressources (les agences spatiales considèrent désormais sérieusement l'exploitation de ressources lunaires).

1.1.2 Un problème de rêve pour les roboticiens

Pour mener à bien ces objectifs, les planétologues s'accordent à dire qu'une sonde fixe n'est pas suffisante : ainsi le succès des deux missions Viking sur Mars est loin d'avoir satisfait leur soif de données. Ils aimeraient disposer de moyens de cartographie précis, de mesures diverses et de collectes d'échantillons en divers endroits distants les uns des autres, et ce sur de longues durées de temps [Costard 92]. Ils sont aussi conscients que l'ensemble des missions ne peut être défini *a priori*, et que les résultats des premières missions définiront les suivantes. En bref, ils aimeraient disposer d'une base mobile avec laquelle des interactions sont possibles, et qui soit capable d'interpréter et d'exécuter les ordres qui lui sont transmis.

Ce contexte applicatif ambitieux pose un problème de choix qui agite particulièrement la communauté des roboticiens. Il nécessite une plate-forme mobile dotée d'une très grande autonomie décisionnelle, qui doit satisfaire à des contraintes et des spécifications exigeantes :

- Les besoins propres à une mission d'exploration rendent nécessaires de longs déplacements de plusieurs dizaines, voire centaines de kilomètres, et le robot doit pouvoir rallier des objectifs précis ;
- Une mission n'est pas définie une fois pour toutes : en fonction de son déroulement et des résultats intermédiaires, il doit être possible de la modifier. Il faut donc maintenir un lien de contrôle entre la terre et le robot, afin de pouvoir spécifier de nouvelles opérations à exécuter. Pour cela, il est notamment nécessaire de connaître l'état et la position du robot ;
- Les contraintes de communication entre le robot et la terre sont très fortes : dans le cas de Mars par exemple, les délais peuvent atteindre plus de 40 minutes aller et retour, les débits sont de l'ordre de 1 à 2 kilo-bits par seconde et les fenêtres de communication théoriques sont d'environ 8 heures par jour. Ces contraintes prohibent bien entendu toute téléopération directe, et les échanges d'informations entre le robot et la terre ne peuvent se faire qu'à un certain niveau d'abstraction ;
- L'environnement du robot est initialement totalement inconnu ou presque, et est extrêmement varié et complexe ; de plus, le robot n'en a connaissance qu'à travers ses propres capteurs, qui fournissent des données partielles et imprécises. Les seules représentations de l'environnement dont disposera le robot seront celles qu'il construit, et seront donc elles-mêmes partielles et imprécises ;
- L'environnement n'étant que très peu connu *a priori*, une mission ne peut être définie qu'à un haut niveau d'abstraction, elle ne peut être précisée en détail. Le robot doit donc être capable d'interpréter de façon autonome une mission en fonction de son propre état et des connaissances qu'il a acquises sur son environnement ;

- Enfin, le robot peut se trouver face à des situations dont il ne sait se sortir. Il est en effet illusoire de prétendre réaliser un robot qui mènera à bien *toutes* les tâches qui lui sont demandées, particulièrement dans des environnements très complexes ; afin de garantir le succès d'une mission, il faut préserver la possibilité d'une intervention humaine directe, qui peut s'effectuer à un très bas niveau de commande.

Mentionnons aussi les contraintes technologiques propres à toute application spatiale : la masse du système est nécessairement limitée, la puissance disponible est une ressource rare, les variations de température sont très grandes, les performances de l'informatique embarquée sont bien moindres que celles que l'on peut espérer pour une application terrestre... Ces contraintes compliquent sérieusement la tâche des roboticiens, qui doivent notamment considérer d'épineux problèmes de gestion de ressources, informatique, électronique et énergétique.

1.1.3 Les projets impliquant le LAAS

Le premier projet de robotique mobile en terrain naturel auquel a participé le LAAS était le projet Euréka AMR, débuté en 1987 [Lacombe 91]. Le groupe a contribué à la conception et l'intégration du système de perception (modélisation de terrain à partir de données tridimensionnelles et localisation extéroceptive), et le projet a notamment permis à Framatome et Matra Marconi Space d'acquérir et d'instrumenter le robot Adam.

En 1989, le Centre National d'Études Spatiales a engagé le projet "Véhicule Automatique Planétaire" (VAP [Giralt 92, Maurette 92]), dont les objectifs étaient de démontrer la faisabilité technique d'un véhicule autonome d'exploration planétaire. Pour ce faire, le CNES s'est assuré le partenariat des principaux laboratoires de recherche français travaillant dans le domaine de la robotique : le Commissariat à l'Énergie Atomique (CEA), le Centre National de la Recherche Scientifique (CNRS), l'Institut National de la Recherche en Informatique et Automatique (INRIA) et l'Office National d'Études et de Recherches Aérospatiales (ONERA). Ces quatre organismes ont fondé le groupement "Robots d'Intervention sur Site Planétaire" (RISP), au sein duquel le LAAS mène une participation très active, en collaborant notamment sur les aspects de génération des déplacements autonomes (modélisation de l'environnement, navigation et raisonnement géométrique pour la planification de mouvements [Chatila 93b]), de structure décisionnelle (gestion des tâches et contrôle de l'activité du véhicule [Alami 93a]), de planification de mission au sol et de téléprogrammation [Alami 93b].

Fin 1993, le CNES a engagé le projet Euréka IARES (Illustrateur Autonome de Robotique mobile pour l'Exploration Spatiale [?, Richard 94]), qui inclut cette fois des partenaires européens et industriels, et qui a pour objectif de valider les concepts fonctionnels utilisables par de futurs robots planétaires. Défini dans un contexte d'exploration de la planète Mars, le projet devait déboucher sur la réalisation d'un démonstrateur prototype en 1997, objectif aujourd'hui remis en cause par le contexte programmatique du CNES.

La phase de conception système et de conception préliminaire des sous-systèmes d'IARES s'est déroulée durant l'année 1994. Certains travaux de ce mémoire [Lacroix 95b] ont été

proposés pour la conception du sous-système dit de "Génération des Déplacements Autonomes", pour lequel nous avons travaillé avec les partenaires suivants : Alcatel Espace, Alcatel Alsthom Recherche, l'INRIA et bien entendu le CNES.

1.1.4 Le projet EDEN

Le robot mobile tout-terrain Adam (figure 1.1), propriété de Framatome et de Matra Marconi Space, est en prêt au LAAS depuis Octobre 1992. Grâce à ses capacités de locomotion et de calcul et à l'ensemble des capteurs dont il est doté¹, Adam est un outil qui permet l'expérimentation et la validation d'algorithmes, étapes que nous jugeons indispensables à la réalisation d'un travail de recherche appliquée. De nombreux doctorants et stagiaires ont notamment pu bénéficier de ce support expérimental, et ses possibilités sont encore loin d'avoir été toutes explorées.



Figure 1.1 : *Le robot Adam dans l'environnement d'expérimentation du LAAS*

Adam nous a notamment permis de mettre en place le projet interne EDEN (Expérimentation de Déplacement en Environnement Naturel), dont l'objectif est le développement et la mise en oeuvre des capacités nécessaires à la navigation autonome en environnement naturel [Chatila 93a, Nashashibi 94, Lacroix 94a]. Les thèmes abordés dans le cadre de ce projet couvrent la plupart des aspects liés à la robotique mobile autonome : perception et modélisation de l'environnement, génération de trajectoires, architecture matérielle et logicielle, structure décisionnelle et contrôles des activités du robot... Le dernier chapitre de ce mémoire présente plus en détail les différents travaux qui ont été réalisés dans le cadre de ce projet.

¹Ses caractéristiques seront détaillées en fin de mémoire, lorsque nous présenterons les expérimentations que nous avons effectuées

1.2 Les approches et réalisations connues

Ce n'est qu'à partir des années 80 que des projets ont consisté à réaliser des robots mobiles véritablement *autonomes* en environnement naturel. Sans chercher à être exhaustif, nous présentons ici brièvement les principaux projets qui ont débouché sur des réalisations expérimentales.

1.2.1 Autonomous Land Vehicle (ALV)

Initié au début des années 80 par l'agence de défense américaine (DARPA), le projet ALV [Olin 91] a permis de nombreux développements dans les domaines de l'architecture logicielle d'un robot [Linden 86], de la modélisation de l'environnement à partir de données télémétriques [Daily 88b], et d'évitement d'obstacles en temps réel. Les premières expériences consistèrent à suivre une route, et en 1987 des expériences réalisées à Denver montraient la capacité de ALV de se déplacer en terrain naturel, en suivant une route initialement planifiée sur une carte d'élévation et en évitant les obstacles imprévus [Daily 88a].

Le projet a aussi permis de développer "NavLab" à l'université Carnegie Mellon [Goto 87, Thorpe 91a, Thorpe 91b] : NavLab est un véhicule commercial équipé de capteurs (télémètre laser et caméra couleur) et de calculateurs, il a montré des capacités de suivi de route sans s'interrompre, en analysant les données fusionnées de ses capteurs [Shafer 86, Hebert 89a, Thorpe 88]. Il est aussi capable d'évoluer en terrain naturel.

1.2.2 Ambler

À l'institut de robotique de l'université Carnegie Mellon, le projet Ambler était au carrefour de recherches concernant les principaux domaines de la robotique mobile [Bares 89, Simmons 91, Krotkov 91]. Le projet visait à développer un système capable de se déplacer avec un minimum d'interaction avec l'homme, dans un contexte d'exploration planétaire. L'architecture de locomotion retenue était un hexapode de conception originale, dont les dimensions impressionnantes répondaient à des contraintes fortes sur ses capacités de traversabilité et sur l'énergie consommée par kilomètre parcouru [Krotkov 92]. De nombreuses avancées ont pu être réalisées dans le contexte de ce projet : elles concernent principalement la modélisation incrémentale de l'environnement à partir de données télémétriques [Hebert 89b, Kweon 88], la planification de déplacement d'un robot marcheur, et l'intégration matérielle et logicielle d'un système robotique. Le projet est désormais stoppé, mais ses résultats servent à la réalisation d'un nouveau projet (Apex [Simmons 93]).

1.2.3 Autres projets

Mentionnons pour mémoire d'autres projets passés ou en cours qui ont pour but la réalisation d'un véhicule tout-terrain autonome : Robby au Jet Propulsion Laboratory ; Unmanned Ground Vehicle (UGV), projet qui réunit de nombreux laboratoires américains sous la hou-

lette de la DARPA ; le projet DARDS en France, auquel participe notamment la société Dassault...

1.2.4 Les approches "comportementalistes"

Les contraintes technologiques fortes inhérentes à toute application spatiale ont conduit la NASA à délaisser les gros systèmes développés durant les années 80, jugés coûteux et complexes, pour s'intéresser tout particulièrement aux robots "réactifs", bien plus petits et plus simples [Weisbin 92]. L'architecture de contrôle de ces systèmes est basée sur une approche comportementaliste [Gat 92, Gat 94], initialement développée au MIT. Plusieurs petits véhicules ont ainsi été conçus au Jet Propulsion Laboratory [Wilcox 92] (la famille Rocky [Miller 92], Go-For [?], ...).

Sans entrer dans le débat qui oppose les approches comportementales aux approches délibératives, nous considérons que si de tels robots sont vraisemblablement robustes (parce que simples), ils sont loin de satisfaire l'ensemble des spécifications que les scientifiques attendent d'une mission d'exploration.

1.3 L'architecture globale proposée au LAAS

Partant du principe qu'un robot n'est pas une "créature" qui vagabonde librement suivant son inspiration, mais bien une "machine" qui effectue des tâches non répétitives, et qu'un opérateur doit pouvoir contrôler en fonction des besoins et de l'évolution de la mission, nous proposons une architecture du système composée de deux entités physiques distinctes qui communiquent entre elles : (i) la station de contrôle et (ii) le robot proprement dit (figure 1.2).

Cette architecture repose sur le concept de "télé-programmation au niveau tâche", qui permet à l'homme de faire partie de la boucle de décision au niveau de la programmation de mission, alors que le robot possède toutes les capacités d'exécution autonome. Cette approche est développée depuis de nombreuses années, et s'est affinée lors de la participation du groupe à différents projets appliqués de robotique d'intervention [Giralt 88, Degallaix 91, Laurette 91]. Elle nous apparaît particulièrement adaptée dans le cas d'un robot d'exploration planétaire, où l'environnement n'est que très peu connu *a priori* et où les contraintes de communication sont fortes. Elle a notamment été proposée dans le cadre des projets VAP et IARES.

1.3.1 La station de contrôle

Parce que l'opérateur ne dispose que d'informations incertaines et incomplètes concernant le robot et son environnement, il ne peut spécifier la mission qu'en termes de *tâches* d'un haut niveau d'abstraction, que le robot doit interpréter en fonction du contexte afin de générer les commandes appropriées. L'opérateur peut ainsi émettre des requêtes telles que "va

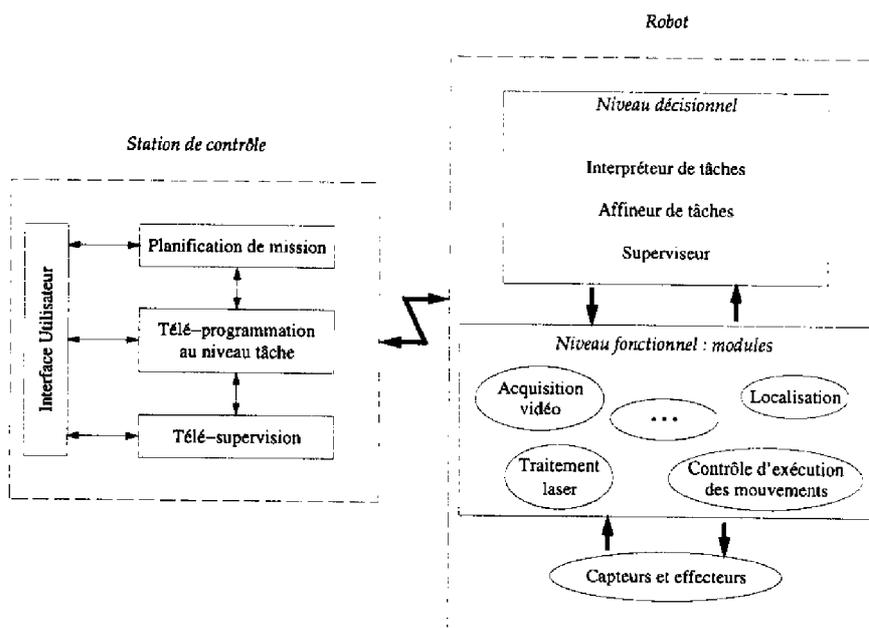


Figure 1.2 : Les principaux composants d'une architecture pour un système robotique d'intervention à distance

prendre cet objet", "explore cette région", "retourne à la position initiale"... La station de contrôle lui fournit des moyens logiciels d'aide à la planification de ces tâches et produit un *plan de mission interprétable*, c'est-à-dire une séquence de tâches exécutables par le robot. Elle lui permet aussi de superviser - ou de modifier la cas échéant - le déroulement d'une mission.

a) Planification de mission

Une mission décrite à un très haut niveau d'abstraction (telle qu'un scientifique l'exprimerait par exemple) est bien souvent loin d'être directement réalisable de manière autonome par le robot : elle peut s'étaler sur plusieurs jours ou consister en des déplacements de plusieurs dizaines de kilomètres. La phase de planification de mission permet de "dégrossir" une telle mission, en considérant les contraintes temporelles entre les différentes tâches de la mission (déplacement d'un site à l'autre, acquisition et envoi de panorama, déploiement de matériel...), et les contraintes d'utilisation des ressources du robot que l'on peut déterminer *a priori*. Ceci est réalisé grâce au système de planification temporelle *IxTeT*, qui raisonne sur des contraintes symboliques et des relations temporelles entre des instants [Ghallab 89]. Le plan résultant est le "plan nominal", succession de tâches partiellement ordonnées.

b) Télé-programmation

La phase de télé-programmation consiste à décomposer les différentes tâches exprimées dans le plan nominal en procédures ou actions interprétables par le robot (plan de mission interprétable ou *télé-programme*). Il s'agit d'un processus d'affinement, qui consiste à déterminer *a priori* (ie à partir des connaissances très incomplètes dont dispose l'opérateur) les variantes possibles du plan nominal, afin de spécifier au robot sa liberté d'action et les situations à

éviter [Perret 94].

Typiquement, un télé-programme contient une succession de tâches à accomplir, accompagnées de leurs modalités d'exécution : spécifications de situations particulières et des réactions appropriées, et indications concernant le contexte de la mission (contraintes à respecter sur le temps, la consommation d'énergie...).

Il contient aussi des informations inaccessibles au robot, et notamment une carte de la zone où s'effectue la mission. Dans un contexte d'exploration planétaire, cette carte est un réseau d'itinéraires possibles au sein desquels le robot peut se déplacer [Fagegaltier 94]. Elle est très grossière (à partir d'images satellites par exemple, on ne peut espérer disposer d'une carte de résolution plus fine qu'une dizaine de mètres en général), et est structurée en tronçons, pour lesquels des caractéristiques générales sont connues [?] : type et degré de difficulté supposés du terrain, pente moyenne, possibilité de recalage sur des amers lointains...

c) Télé-supervision

Le rôle de la télé-supervision est de surveiller le bon déroulement de la mission (suivi) et de permettre à l'opérateur d'intervenir en cas d'échec (dépannage). Le suivi est réalisé grâce à l'envoi par le robot de rapports d'exécution ou de données acquises à certaines étapes clés de la mission, actions qui sont spécifiées dans le télé-programme. Le dépannage consiste en l'envoi de procédures de test et d'éventuelles modifications du télé-programme en cas de situation anormale.

1.3.2 Le robot

Le robot reçoit de la station de contrôle le télé-programme. Il est doté de capacités décisionnelles qui lui permettent de décomposer chaque tâche à exécuter en actions élémentaires (principalement la commande de ses capteurs et actionneurs, et l'envoi de rapports d'exécution ou de requêtes à la station de contrôle), en fonction du contexte de la mission, de son état courant et des représentations de l'environnement dont il dispose.

a) Le niveau décisionnel

Le rôle de ce niveau est (i) d'interpréter les tâches envoyées par la station de contrôle en fonction de l'état d'exécution de la mission, du robot et de son environnement ; (ii) de transformer ces tâches en séquences d'actions exécutables (affinement) ; et (iii) de superviser leur exécution par le niveau fonctionnel, en analysant toute situation nouvelle et en décidant des réactions appropriées, conformément aux modalités d'exécution reçues.

Le paragraphe suivant (1.4) présente le processus d'affinement de la tâche de navigation que nous avons retenu.

b) Le niveau fonctionnel

Le niveau fonctionnel comprend les fonctions d'action (déplacement et acquisition de données) et diverses fonctions de traitement de données (traitement d'image, génération de trajectoire, boucles de contrôle...). Toutes ces fonctions, qui peuvent partager des don-

nées et des ressources, sont encapsulées au sein de “modules” définis de manière générique [Chatila 90, Fleury 94]. Chaque module contient un “gestionnaire de module”, qui reçoit les requêtes émises par le niveau décisionnel, vérifie leur validité et surveille leur exécution. Les modules interagissent entre eux grâce à des échanges de messages, ou en lisant des données produites par un autre module, et communiquent leur résultat à travers des structures de données exportées.

L'avantage d'une telle architecture modulaire est qu'elle est ouverte (aspect non négligeable pour un robot de laboratoire), et surtout qu'elle permet de définir aisément les séquences d'action et de traitement des données nécessaires à la réalisation d'une tâche. Nous verrons à la fin du mémoire les différents modules mis en jeu dans les expériences d'EDEN, et comment les connections entre eux sont déterminées par le niveau décisionnel lors de la tâche de navigation.

1.4 Notre approche de la navigation

1.4.1 La tâche canonique : ‘‘Aller à [But]’’

Afin de préciser le problème auquel nous nous intéressons, nous décrivons ici la tâche qu'un robot d'exploration sera très souvent amené à réaliser : il s'agit de la tâche ‘‘Aller à [But]’’, qui consiste à rallier de façon autonome un but arbitrairement distant. Bien d'autres tâches complexes doivent aussi être exécutées de manière autonome (collecter des échantillons, déployer du matériel...), mais quasiment toutes les missions qu'un opérateur peut spécifier au robot comporteront un certain nombre d'instances de la tâche ‘‘Aller à’’ ; sa réalisation est donc d'une importance fondamentale.

Il s'agit à proprement parler d'une tâche de *navigation*, qui implique que le robot soit capable de réaliser les quatres actions suivantes :

- Modéliser l'environnement² ;
- Se localiser dans l'environnement ;
- Planifier des déplacements ;
- Assurer la bonne exécution des déplacements planifiés.

Bien entendu, chacune de ces actions est plus ou moins aisée à réaliser suivant la nature de l'environnement dans lequel évolue le robot et la connaissance initiale que l'on en a, mais aussi suivant les capacités de locomotion du robot. Les environnements naturels sont certainement les plus difficiles : leur modélisation à partir des données sensorielles n'est pas triviale (il est notamment difficile de s'y localiser), et y planifier des déplacements dont la bonne exécution est garantie est très délicat.

À l'initialisation de cette tâche, les informations disponibles sont les suivantes :

²Cette étape est nécessaire lorsque l'environnement est *a priori* inconnu. Dans le cas contraire, il s'agit ‘‘d'interpréter’’ l'environnement afin de s'y localiser.

- La description du but à atteindre : il peut s'agir d'une position (X, Y) donnée, d'un endroit particulier que le robot est capable d'identifier, ou plus généralement d'une zone présentant certaines caractéristiques ;
- Un certain nombre de contraintes à respecter lors de l'exécution de la tâche, spécifiées lors de la phase de télé-programmation (sur le temps et l'énergie notamment) ;
- Et éventuellement une description de l'environnement de la mission (elle peut parfois être inexistante) : il s'agit dans notre cas de la carte d'itinéraires que nous avons mentionnée dans le paragraphe 1.3.1. Sa faible résolution ne permet en aucun cas de planifier des mouvements du robot *a priori*.

1.4.2 Une approche hiérarchique

Le principe de résolution hiérarchique d'un problème est une approche classique en intelligence artificielle, et de nombreux auteurs ont abordé ainsi la navigation d'un robot mobile [Linden 86, Metea 87, Garcia 91, Kuipers 91] et les problèmes de planification de trajectoire [Pignon 91]. En ce qui nous concerne, nous avons choisi de hiérarchiser l'ensemble des décisions qui constituent l'affinement de la tâche 'Aller à' en trois niveaux de planification distincts, qui raisonnent chacun sur une représentation de l'environnement qui lui est propre (figure 1.3) :

- **Planification d'itinéraires** : À partir de la définition de la mission et de la description initiale de l'environnement, la phase préalable de planification de mission a permis de définir un chemin nominal (succession de tronçons), ainsi que d'éventuels chemins alternatifs. Le planificateur d'itinéraires dispose des mêmes informations sur l'environnement et la définition de la mission (qui représente son but global³), et a la charge de suivre le déplacement du robot au sein des tronçons. Il sélectionne éventuellement un des chemins alternatifs, en fonction de l'état courant de l'exécution de la mission et de son contexte. Il détermine ainsi une succession ordonnée de sous-buts à atteindre ;
- **Planification de chemins** : Son but global est le premier des sous-buts déterminés par le planificateur d'itinéraires. Le planificateur de chemin dispose de la description initiale du tronçon courant (d'une taille de l'ordre de quelques centaines de mètres), ainsi que d'informations perçues par le robot. Son rôle est de déterminer au sein du tronçon un chemin qui mène à son but global, ainsi qu'un sous-but à rallier le long de ce chemin : il contrôle ainsi les planificateurs de trajectoire ;
- **Planification de trajectoires** : Afin que le robot puisse atteindre le sous-but trouvé par le planificateur de chemins, il est nécessaire de déterminer la séquence de déplacements élémentaires à effectuer. C'est le rôle du planificateur de trajectoire, qui dispose pour cela d'une description précise de l'environnement, dont la taille n'excède pas une ou deux dizaines de mètres.

³Le terme "but" est à comprendre au sens large pour chacun de ces planificateurs : il définit bien sûr une position à atteindre, mais aussi l'ensemble des contraintes à respecter.

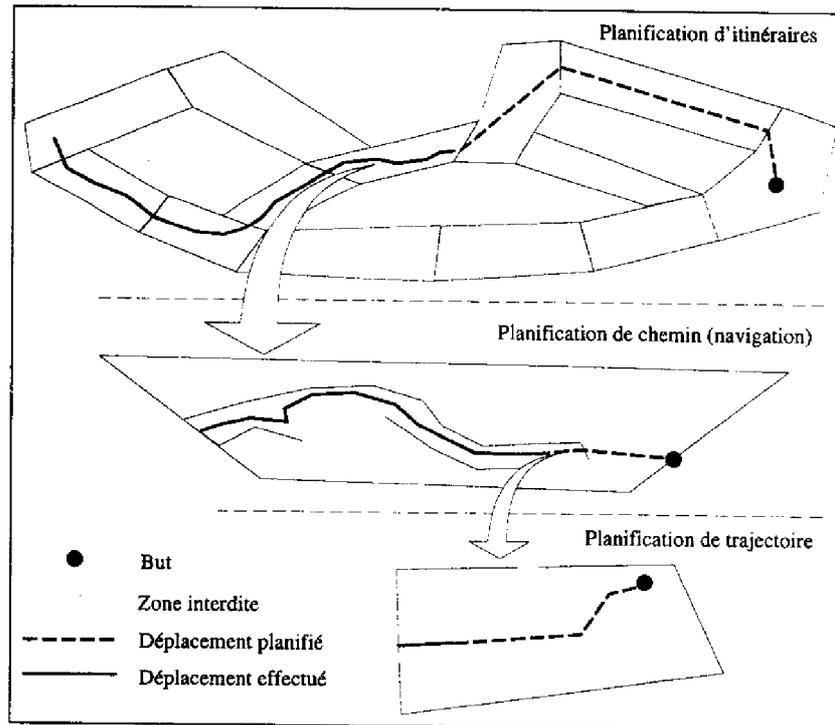


Figure 1.3 : Illustration du principe de planification hiérarchique des déplacements

Un des grands avantages d'une telle approche est qu'elle permet de structurer la résolution du problème : ainsi chaque niveau de planification contrôle le niveau qui lui est directement inférieur, en lui spécifiant un but et son domaine de travail (et notamment la représentation de l'environnement sur laquelle il va raisonner). Les cas d'échec sont aisément analysés : si un planificateur d'un certain niveau échoue (impossibilité de rallier le but qui lui est spécifié), c'est que la représentation de l'environnement correspondant au niveau supérieur est erronée. Il faut alors réviser cette représentation, et réactiver le niveau supérieur, afin de déterminer un nouveau but pour le planificateur qui a échoué.

Nous nous intéressons dans ce manuscrit au planificateur de chemins au sein d'un tronçon, dont l'introduction permet de réduire la complexité du problème : choisir un chemin pour rallier un but nécessite l'évaluation d'un grand nombre de chemins, ce qui serait extrêmement coûteux avec un planificateur de trajectoire.

1.4.3 Différents modes de déplacement

L'environnement dans lequel sera amené à évoluer le robot peut être très varié : il peut aussi bien s'agir de zones planes et dégagées d'obstacles que de zones extrêmement bouleversées, comprenant de nombreux obstacles, des crevasses, des pierriers... Il est clair que le degré de difficulté des divers problèmes liés à la navigation dépend fortement du type d'environnement dans lequel le robot se trouve, qu'il s'agisse de localisation, de modélisation, de planification de déplacement ou de contrôle d'exécution des trajectoires.

Par ailleurs, un élément clé qui permet de juger de l'intelligence d'un système, et donc de son efficacité, est sa faculté d'adaptation à son environnement. Nous avons donc défini trois modes d'exécution des déplacements différents, qui sont déterminés par le type de terrain à traverser. Ils correspondent chacun à l'application d'un planificateur de trajectoire particulier :

- Un mode "planifié 3D", appliqué lorsque le terrain est accidenté. Il repose sur la réalisation d'une trajectoire planifiée en considérant des contraintes de non-collision avec l'environnement et sur la stabilité du robot [Siméon 93, Dacre-Wright 93a] ;
- Un mode "planifié 2D", appliqué lorsque le terrain est essentiellement plan, mais présente néanmoins des obstacles. Il repose sur l'exécution d'une trajectoire qui a été planifiée en ne considérant que les contraintes de non-collision avec ces obstacles⁴ ;
- Et un mode dit "réflexe"⁵, appliqué lorsque le terrain est supposé quasiment libre de tout obstacle : le robot évolue alors en suivant une simple valeur de consigne (cap ou position), sans avoir préalablement planifié de trajectoire, mais en surveillant toutefois de façon continue la présence d'éventuels obstacles grâce à ses capteurs.

Le choix de doter le robot de différents modes de déplacement correspond en fait à l'application d'un principe d'économie des moyens, qui vise à garantir un maximum d'efficacité : ainsi il n'est pas nécessaire de planifier une trajectoire si le terrain ne présente aucune difficulté, et il est inutile de vérifier des contraintes sur la stabilité du robot (procédure lourde et coûteuse en temps) si le terrain est plan. De même, les procédures de modélisation fines, généralement coûteuses en temps, ne sont exécutées que lorsqu'elles s'avèrent nécessaires, et sur des portions de terrain qu'il faut déterminer. L'application de ce principe nécessite donc que les fonctions d'acquisition et de traitement des données doivent être activement contrôlées, en fonction de l'état de l'environnement et des besoins courants de la tâche : on vise ainsi une utilisation "intelligente" des capacités perceptuelles du robot.

1.4.4 Conséquences : de l'importance du planificateur de chemins

S'il est certain que l'application du mode de déplacement le mieux adapté à l'environnement immédiat du robot permet d'augmenter son efficacité, il est aussi clair que le fait de disposer de plusieurs modes de déplacement complique considérablement le système. La question qui se pose notamment avant chaque déplacement est la sélection du mode le plus adéquat et de la prochaine tâche de perception à exécuter.

Nous avons donc pour cela défini une stratégie particulière pour la navigation au sein d'un tronçon : elle consiste à mettre à jour *systématiquement* (*i.e.* chaque fois que des données sensorielles sont acquises) une représentation de l'environnement basée sur une rapide analyse des données. La première partie du manuscrit est consacrée à la construction de cette représentation, qui est mémorisée sur l'ensemble du tronçon. Elle est l'information de base qui va être utilisée par le planificateur de chemin à *chaque étape du déplacement* pour déterminer :

⁴Les termes 2D et 3D sont ici relatifs à la dimension des modèles sur lesquels travaillent les planificateurs, et non à la dimension de l'espace des configuration du robot : ainsi le planificateur 2D peut raisonner sur les trois paramètres de position (x, y, θ) .

⁵C'est par abus de langage que nous considérons qu'il s'agit d'un mode de planification

- Un sous-but à atteindre ;
- Le mode de déplacement à appliquer pour le rallier ;
- Et la prochaine tâche de perception à exécuter.

Le choix de cette stratégie définit une instance particulière de la boucle “Perception-Décision-Action” (figure 1.4) : le planificateur de chemin contrôle les planificateurs de trajectoire et les capacités perceptuelles du robot, en respectant les contraintes définies par le planificateur d’itinéraires, et définit ainsi les stratégies de navigation à suivre (par abus de langage, nous désignerons souvent par le mot “navigation” la fonctionnalité de planification de chemin).

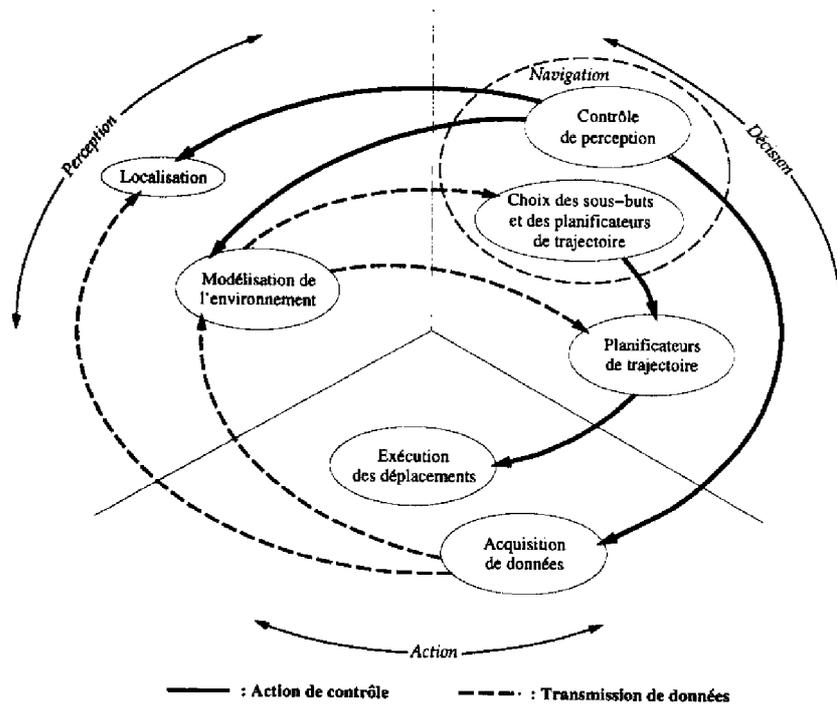


Figure 1.4 : Le rôle du planificateur de chemin (navigation) dans la boucle “Perception-Décision-Action”

Première Partie

Modélisation de l'environnement

Chapitre 2

Introduction

2.1 Utilité des représentations

Dans une approche délibérative, l'autonomie et l'efficacité du robot dépendent grandement de sa capacité à extraire des caractéristiques cohérentes avec l'état réel de l'environnement à partir des données perceptuelles. Pour la navigation telle que nous l'avons présentée, une représentation de l'environnement est nécessaire pour chacune des trois principales fonctionnalités suivantes : *(i)* planification de déplacements, *(ii)* planification de perception et *(iii)* localisation.

- **Planification de déplacements** : suivant notre principe de résolution hiérarchique du problème de la navigation, chaque action de déplacement est exécutée après avoir été planifiée en deux temps :
 - la procédure de *planification de chemin* (navigation) permet de déterminer la route à suivre pour rallier le but, et en déduit un but intermédiaire à rallier, ainsi que les planificateurs de trajectoire adéquats selon la nature du terrain à traverser ;
 - et une procédure de *planification de trajectoire* détermine alors la séquence de commandes à exécuter pour atteindre le but intermédiaire sélectionné.

Ces deux procédures nécessitent un modèle de l'environnement, qui sert à évaluer la faisabilité des actions correspondantes (suivre un chemin ou une trajectoire), ainsi que leur intérêt et le coût nécessaire à leur réalisation ;

- **Planification de perception** : la principale fonctionnalité des tâches de perception est d'acquérir plus d'informations pertinentes pour une meilleure exécution de la tâche de navigation. Elles sont exécutées après avoir été planifiées, et requièrent donc un modèle de l'environnement qui permet de tester la satisfaction de contraintes perceptuelles, de prédire le résultat de leur activation, et finalement d'en évaluer l'intérêt dans le contexte de la tâche à exécuter ;
- **Localisation** : afin de contrôler le déroulement de la mission, mais aussi de garantir que les représentations construites sont cohérentes avec la réalité de l'environnement, il

est nécessaire de connaître précisément la position du robot. Les capteurs proprioceptifs du véhicule (centrale inertielle, inclinomètres, odométrie...) ne sont pas toujours fiables, et leurs données doivent être fréquemment corrigées en localisant le robot *par rapport* à l'environnement grâce aux capteurs extéroceptifs. Ceci est réalisé par une procédure de mise en correspondance de caractéristiques de l'environnement nouvellement perçues avec un modèle préalablement construit et mémorisé ;

Le robot doit donc disposer d'une connaissance de l'environnement qui lui permette de réaliser des calculs numériques assez précis pour la tâche de localisation par exemple, mais aussi d'effectuer des raisonnements d'un assez haut niveau d'abstraction, qui nécessitent des capacités de prédiction et d'évaluation du résultat d'une action de déplacement ou de perception. Suivant la fonctionnalité pour laquelle elles sont requises, les représentations ont donc des caractéristiques très différentes.

C'est pourquoi nous utilisons le terme "représentations de l'environnement" au pluriel : il existe un modèle spécifique pour chaque fonctionnalité, qui ne contient que les caractéristiques pertinentes pour cette fonctionnalité (on parle de représentations *dédiées*). Nous verrons à la fin de cette partie du mémoire comment les diverses représentations peuvent s'insérer dans un schéma structurel simple, qui permet d'explicitier leurs éventuelles relations et d'organiser les différents modules de traitement des données perçues.

2.2 De la nécessité de représentations globales

Il y a un grand intérêt à construire et à maintenir des représentations *globales* de l'environnement, qui couvrent l'ensemble des zones perçues par le robot lors de l'exécution de la mission. Par rapport à une approche où le système ne disposerait que de représentations locales (*i.e.* qui ne couvrent que la zone dernièrement perçue), cela permet notamment d'aborder les problèmes suivants :

- **Traitement des impasses** : le robot peut se trouver au fond d'une impasse de laquelle il doit être capable de sortir de façon autonome. C'est clairement grâce à une représentation globale de l'environnement qu'il pourra résoudre efficacement ce problème : la notion même d'impasse ne peut exister dans une représentation locale, et le robot risque de s'obstiner à tenter de rallier le but en continuant à se "fourvoyer". Dans ce cas, des stratégies un peu plus évoluées pourraient permettre de détecter que le véhicule ne peut plus avancer, et donc de décider de passer par un autre chemin : mais sans carte globale, le chemin ne peut être choisi qu'à l'aide d'heuristiques plus ou moins aléatoires, et il est alors difficile de garantir que le robot remplira sa mission avec succès et efficacement. Cette capacité de remettre en cause un chemin que le robot a commencé à exécuter en décidant d'effectuer un demi-tour est à nos yeux une raison qui à elle seule justifie la nécessité de construire une représentation globale de l'environnement ;
- **Retour en arrière** : si la tâche canonique que nous voulons réaliser consiste essentiellement à rallier un but, l'opérateur peut néanmoins demander au robot de revenir à

sa position initiale, voire d'effectuer plusieurs allers et retours (cas d'une mission de transport par exemple). Il est évident dans ce cas que les performances du robot seront considérablement améliorées si l'environnement est modélisé et mémorisé :

- Localisation : par essence, une fonctionnalité de localisation sur la base de données extéroceptives nécessite de conserver en mémoire des zones particulières de l'environnement (amers) précédemment détectés ;
- Enfin, mentionnons que cartographier l'environnement peut être aussi en soi un des objectifs de la mission.

L'environnement est découvert de façon *incrémentale* par les capteurs extéroceptifs du robot, et on doit donc disposer d'algorithmes qui permettent d'agrèger les données acquises en des représentations globales. Ces données étant toujours *partielles* et *imprécises*, les représentations doivent *tolérer des variations* : même si le robot évolue dans un environnement supposé statique¹, la connaissance qu'il en a évolue.

Hormis les contraintes à respecter sur l'occupation mémoire, la difficulté de maintenir des représentations globales cohérentes avec l'état réel de l'environnement réside donc essentiellement dans la gestion de leur caractère incertain et évolutif.

2.3 Quel modèle pour la navigation ?

2.3.1 Contributions connues

La navigation raisonne en termes de chemins et non de trajectoires : elle ne nécessite pas une représentation qui permette de raisonner dans l'espace des configurations du véhicule, mais une représentation globale plus qualitative : on parle souvent de "carte cognitive" [Kuipers 83]. Ces représentations contiennent différents niveaux de description : on distingue généralement un niveau géométrique qui permet d'identifier des amers et des lieux, et un niveau topologique qui exprime des relations spatiales et d'accessibilité entre les différents lieux (routes) [Levitt 90, Kuipers 91].

Dans des environnements intérieurs, il existe de nombreuses contributions qui concernent la construction de telles représentations. Des représentations décrivant le monde par des régions polygonales [Chatila 85, Moutarlier 91] sont notamment bien adaptées. Des descriptions basées sur une grille régulière, éventuellement hiérarchisées (*quad-tree*), sont aussi très utilisées [Moravec 85, Elfes 86, Matthies 88, Zelinsky 92] : elles sont aisées à construire et à manipuler, et leur structure définit des relations de voisinage simples. D'autres auteurs [Kuipers 88, Basye 89] construisent des représentations basées sur des éléments caractéristiques de l'environnement (objets), dont les positions relatives définissent un graphe à partir duquel ils extraient des relations d'accessibilité.

La grande difficulté de construire de tels modèles dans des environnements naturels tient essentiellement au fait que ces environnements ne sont pas du tout structurés : d'une part

¹C'est la seule contrainte simplificatrice que nous imposons sur l'environnement

ils ne présentent pas de caractéristiques que l'on peut estimer à l'aide de primitives géométriques simples (plans et segments notamment), et il est donc difficile d'en extraire des objets aisément reconnaissables ; et d'autre part ils n'ont pas de structure topologique triviale, ainsi que l'on en rencontre dans des environnements intérieurs (pièces, portes, couloirs...).

La plupart des efforts concernant la modélisation d'environnements naturels ont généralement consisté à construire des modèles purement géométriques, mais il existe néanmoins quelques travaux qui en extraient des représentations topologiques :

- Des représentations basées sur des grilles (éventuellement hiérarchisées) sont souvent utilisées pour leur simplicité [Metea 87, Stentz 94b] ;
- L'extraction de caractéristiques géométriques de certains points du modèle géométrique permet de définir un "graphe d'objets" [Fagegaltier 94, Kweon 91], et peuvent servir à segmenter l'environnement en régions homogènes [Parodi 86, Gaw 86, Asada 88].

2.3.2 Notre choix

Dans notre approche, la navigation décide des actions de déplacement et de perception, conformément aux contraintes déterminées par le planificateur d'itinéraire. Elle a pour cela besoin d'une représentation de l'environnement qui réponde aux spécifications suivantes :

- Elle doit inclure les éventuelles informations connues *a priori* concernant le tronçon d'itinéraire courant : ces informations guident la navigation au sein du tronçon, en l'empêchant notamment d'en sortir, et en lui faisant préférer les zones supposées faciles à traverser ;
- Elle doit représenter l'environnement en termes de "classes de navigabilité", liées aux modes de déplacement possibles. La navigation pourra ainsi évaluer la faisabilité, le coût et l'intérêt de l'application des différents modes de déplacement ;
- Elle doit permettre d'évaluer la faisabilité et l'intérêt d'actions de perception. En particulier, elle doit comprendre la description des zones dans lesquelles une procédure de localisation extéroceptive est réalisable ;

Pour la détermination des chemins, une description précise ("métrologique") n'est pas nécessaire : vérifier les contraintes de stabilité et de non-collision avec l'environnement (tel que le ferait un planificateur de trajectoire) pour l'ensemble des chemins évalués serait bien trop coûteux, une représentation plus qualitative suffit. Nous avons pour cela choisi une représentation topologique, qui décrit l'environnement par une partition en régions. Chaque région est décrite par différents attributs liés au terrain (nature physique et géométrique), qui permettent de déterminer le mode de déplacement à y appliquer, mais aussi par d'autres attributs liés à la mission initialement compris dans la description du tronçon (tels la visibilité d'amers lointains par exemple).

Par contre, la détermination d'actions de perception requiert une description plus précise : il est en effet nécessaire de tester la vérification de contraintes perceptuelles (visibilité notamment) pour pouvoir prédire ce que peut percevoir le robot en une position donnée.

Mais la contrainte due à la limitation mémoire du système oblige à trouver un compromis entre une représentation précise qui occupe une grande place mémoire, et une représentation moins volumineuse, mais qui ne contient pas forcément toutes les informations que l'on peut extraire des données sensorielles (figure 2.1).

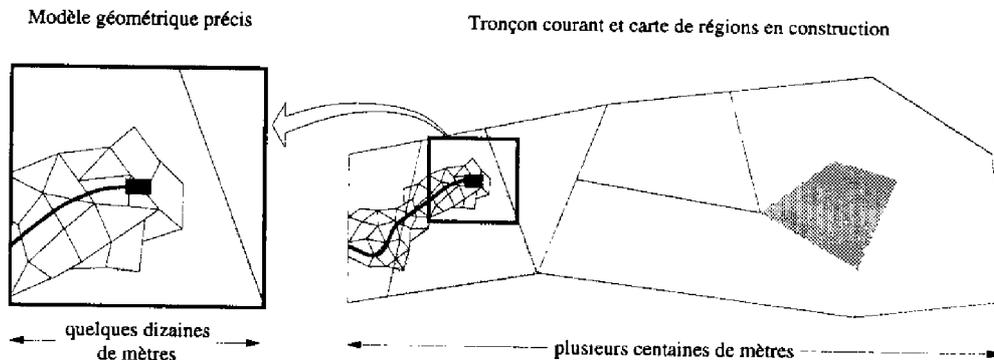


Figure 2.1 : La carte de régions est un affinement de la description initiale du tronçon. Elle contient un modèle géométrique précis autour du robot

Les deux chapitres suivants présentent les algorithmes que nous avons développés pour construire à partir des données acquises une représentation globale sur l'ensemble du tronçon (la "carte de régions", qui contient les informations sur l'environnement nécessaires à la navigation), qui est explicitée de manière plus précise aux abords immédiats du robot.

Chapitre 3

Classification de données 3D brutes

Le développement d'une procédure qui permet d'estimer rapidement la nature du terrain à partir des données perceptuelles est fondamental dans notre approche : cette procédure est systématiquement effectuée à chaque incrément de l'exécution de la tâche de navigation, et elle est à la base de la mise à jour de la représentation de l'environnement nécessaire à la navigation. Ce chapitre présente la technique que nous avons retenue pour analyser les données 3D fournies par les capteurs du robot : nous commençons par y préciser les caractéristiques qu'elle doit avoir, puis nous passons brièvement en revue les techniques existantes. Nous présentons ensuite les caractéristiques de nos capteurs, caractéristiques qui ont beaucoup influencé nos choix. La méthode est ensuite décrite, et des résultats expérimentaux sont analysés.

3.1 Objectifs

Pour construire la représentation de l'environnement nécessaire à la navigation telle que nous l'avons définie dans le chapitre précédent, la procédure d'analyse rapide des données perçues (que nous appelons "classification") doit avoir les caractéristiques suivantes :

- Elle doit tout d'abord classifier l'ensemble de la zone perçue en une partition de *régions*, dont les caractéristiques indiquent les modes de déplacement qui y sont applicables. Nous avons retenu les trois classes de terrain suivantes :
 1. La classe *plan*, sur laquelle le robot planifie des trajectoires en 2D : les régions de cette classe sont navigables, sans aucune contrainte sur l'attitude du robot ;
 2. La classe *accidenté* : le robot peut éventuellement s'y déplacer, mais en exécutant des trajectoires planifiées en mode 3D, tout au long desquelles des contraintes de stabilité et de non-collision avec l'environnement doivent être vérifiées ;
 3. Et enfin la classe *obstacle*, infranchissable pour le robot.

Notons que ces trois classes de terrain ne dépendent que des caractéristiques *géométriques* du terrain (pente moyenne, variations d'altitude, "rugosité"...), alors que la nature *physique* du terrain (sable, cailloux, herbe...), contraint également les déplacements du robot. Ces informations sur la nature du terrain doivent donc aussi caractériser chaque région.

- Les données classifiées étant à la base d'un modèle global de l'environnement, il sera nécessaire de les fusionner, et donc de résoudre les éventuels conflits qui peuvent survenir lors de la perception d'une même zone de terrain à partir de différents points de vue. Ces conflits apparaîtront inévitablement, les capteurs que nous utilisons produisant toujours des données entachées d'erreurs. Il s'agit d'un aspect non négligeable qui générera des erreurs dans la classification : il est nécessaire de connaître la fiabilité des résultats de la procédure de classification.
- Enfin, il est clair que la méthode de classification doit être la plus rapide possible : c'est la procédure de modélisation qui est exécutée le plus souvent, et les performances globales du robot seraient bien dégradées si elle était coûteuse en temps.

Ces caractéristiques, et notamment la contrainte de rapidité d'exécution, font que la procédure de classification que nous avons retenue est essentiellement *qualitative* : le modèle global qu'elle est destinée à construire n'est pas un modèle géométrique précis de l'environnement, mais plus une description "sémantique", sur laquelle seront planifiées des tâches, et non des actions élémentaires.

3.2 Méthodes existantes

Les données 3D que les capteurs permettent d'extraire d'un environnement naturel sont généralement des "images de points". Le caractère non-structuré de tels environnements fait qu'il est difficile, voire impossible, d'obtenir des primitives géométriques plus complexes, telles des segments ou des facettes, que des techniques de stéréovision permettent d'obtenir dans un environnement intérieur. Les capteurs le plus souvent utilisés sont le télémètre laser, la stéréovision par corrélation de pixels, et les capteurs basés sur l'analyse d'une image vidéo de la scène sur laquelle est projeté un motif de lumière structurée.

Il existe bien entendu de très nombreuses contributions dans la littérature qui concernent l'analyse d'images de points 3D acquises en environnement naturel. Nous en présentons ici quelques unes, dont les caractéristiques sont comparables à celles que nous venons de définir.

3.2.1 Méthodes basées sur une carte d'élévation

Les premiers efforts de modélisation d'environnements naturels à partir d'images de points 3D ont consisté à établir une carte d'élévation (ou Modèle Numérique du Terrain - MNT) [Asada 88, Hebert 89b, Kweon 92]. Il est en effet plus facile de travailler sur une structure de grille cartésienne régulière que sur une image de points 3D bruts répartis de façon très irrégulière dans l'espace.

Mais une carte d'élévation reste en quelque sorte une "image de points", et n'apporte pas beaucoup plus d'informations sur l'environnement qu'une image brute. Elle n'est une fin en soi que pour des applications cartographiques, ou pour des procédures qui travaillent directement sur une telle représentation (comme le recalage par corrélation de pixels de MNT). Il est donc souvent nécessaire d'appliquer une procédure de traitement supplémentaire sur ces cartes pour en extraire des informations pertinentes pour la navigation. Ainsi, planifier des trajectoires sur un MNT nécessite de construire un espace de configuration [Dacre Wright 93b], ou de rechercher les placements possible des "pieds" dans le cas d'un robot marcheur [Hebert 89b].

Certaines méthodes consistent à appliquer sur la carte d'élévation des techniques classiques de traitements d'image (masques de convolution), qui permettent de déterminer en chacun des points de la grille des attributs tels que la pente ou la courbure de la surface qu'ils approximent. Des procédures d'extraction de contours basées sur ces attributs permettent alors de détecter les obstacles [Daily 88b] ; ou des procédures de segmentation des points de la carte conduisent à la définition de régions qui respectent des critères d'homogénéité sur ces attributs [Asada 88, Nashashibi 92] : à la rapidité d'exécution près, ces techniques satisfont les spécifications que nous cherchons à réaliser.

3.2.2 Méthodes basées sur des images de points 3D bruts

L'inconvénient des méthodes précédentes est justement qu'elles requièrent la détermination d'une carte d'élévation, procédure coûteuse en temps, surtout lorsque les incertitudes sur les données acquises sont propagées tout au long des calculs.

De nombreux auteurs se sont donc attachés à analyser directement les images de points brutes. Certaines techniques consistent à détecter des obstacles par une simple analyse des profils correspondant à une colonne de points [Grandjean 93, Pagnot 95], ou grâce à une étude de l'élévation des points par rapport au sol [Krotkov 94] ; d'autres consistent à effectuer une segmentation des images sur des attributs de normale ou de courbure afin d'extraire des objets de la scène [Hebert 86, Betge-Brezetz 94], ou à analyser des cellules sur la base d'attributs définis par les points qui y appartiennent [Krotkov 94, Langer 94]. S'il existe des techniques qui consistent à segmenter des images acquises dans des environnements intérieurs en plusieurs classes de terrain liées à la navigabilité [Nashashibi 92, Ballard 94], on en trouve peu appliquées à des images d'environnements naturel.

3.3 Caractéristiques des capteurs 3D

Le robot Adam est équipé de deux systèmes de perception qui produisent des images de points 3D : un télémètre laser à 2 axes, et une paire de caméras utilisées en stéréovision par corrélation. Ces deux capteurs sont semblables, dans le sens où ils perçoivent un ensemble discret de points de l'environnement, et que leurs mesures sont toujours entachées d'incertitudes, voire erronées. Afin de pouvoir interpréter les données qu'ils fournissent, il est fondamental de bien connaître leur comportement : c'est ce à quoi nous nous attachons dans ce paragraphe.

3.3.1 Le télémètre 2-axes

a) Principe de fonctionnement

Avec notre télémètre laser, l'acquisition des données se fait à proprement parler par "lancer de rayon" : le faisceau du télémètre est orienté suivant deux axes de site (ϕ) et d'azimut (θ), et la mesure retournée est la distance ρ entre le télémètre et le point d'impact du rayon. Cette distance est déterminée par une mesure de déphasage ou de temps de vol de la lumière réfléchiée par la surface que vise le faisceau.

Adam (figure 3.1) et Hilare II sont équipés d'un télémètre à temps de vol¹ monté sur une platine orientable en azimut ; et un système mécanique de miroir tournant permet d'orienter le faisceau en site (figure 3.2).

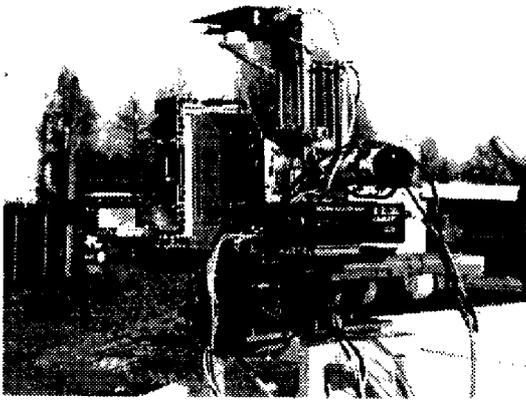


Figure 3.1 : Le télémètre 2-axes d'Adam

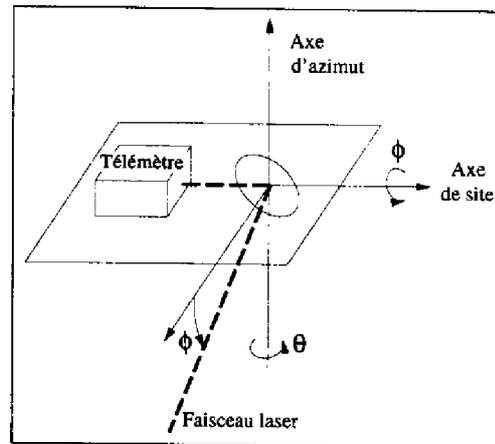


Figure 3.2 : Système de déflexion du faisceau

Le grand avantage d'un capteur basé sur des données télémétriques est qu'il est très peu dépendant des conditions d'illumination de la scène, et qu'il retourne des coordonnées de points sans qu'aucun calcul préalable ne soit nécessaire. Mais il est plus lourd et consomme bien plus d'énergie qu'un capteur basé sur l'analyse d'images stéréoscopiques. De plus, le temps d'acquisition de notre capteur est long : le télémètre requiert 20 milli-secondes pour acquérir un point, et il faut donc plus de trois minutes pour acquérir une image de 100 lignes et 100 colonnes (on parle "d'image laser"). Un télémètre dont le temps d'acquisition est plus court, mais qui sera vraisemblablement moins précis, est actuellement en cours d'installation sur Adam.

b) Géométrie

Le système mécanique de déflexion du faisceau sur Adam fonctionne en *mode non corrélé* : la rotation en azimut se fait autour d'un axe constant, qui ne dépend pas de l'angle de site, et qui définit la verticale propre au capteur. La figure 3.3 illustre ce mode de balayage : pour un angle de site donné, le faisceau décrit un cône d'axe ($O_c Z_c$). Les points correspondant à un sol plan acquis à un angle de site constant sont donc répartis suivant un arc de cercle si

¹Il s'agit d'un télémètre DM 90 de la marque RiegI, disponible dans le commerce, qui est principalement utilisé par des topographes

l'assiette du capteur est horizontale, et suivant un arc d'ellipse sinon.

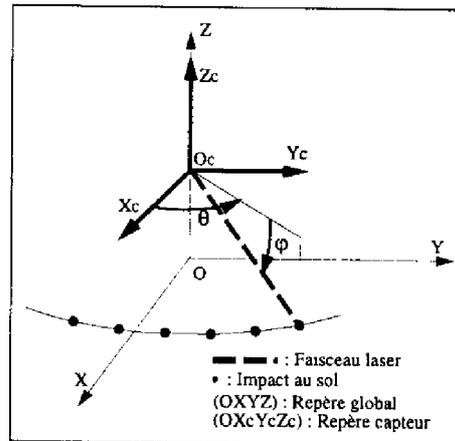


Figure 3.3 : Mode de balayage non corrélé

La réalisation mécanique du système de déflexion fait qu'en réalité les axes de rotation ne sont pas exactement orthogonaux, ni même concourants ; de plus, la mesure retournée par le télémètre est donnée à un décalage et un facteur d'échelle près. L'estimation de tous ces paramètres (par une procédure dite de *calibration intrinsèque* du capteur) n'est pas un problème simple : F. Nashashibi l'a abordé dans ses travaux de thèse [Nashashibi 93], et en a déduit un modèle "exact" de correction des mesures (θ, ϕ, ρ) , assez complexe, ainsi qu'un modèle simplifié qui s'est avéré suffisant. Ce modèle simplifié consiste en l'introduction de décalages sur les 3 mesures, et permet de considérer que les images sont exprimées dans un repère sphérique.

c) Résolution

Le système de contrôle de l'orientation du faisceau effectue des incréments angulaires *fixes* suivant les deux axes. La résolution angulaire du capteur est donc constante, ce qui produit des images de points 3D dont la densité est extrêmement variable : elle décroît notamment fortement avec la distance des points aux capteurs, et dépend de l'orientation des surfaces perçues. Nous attachons une grande importance à cette caractéristique : d'une manière générale, les données sont denses près du capteur, où leur interprétation est facile. Par contre, leur densité se dégrade très rapidement en s'éloignant du capteur, et l'interprétation que l'on peut en faire est d'autant plus délicate (figure 3.4).

La figure 3.5 définit ce que nous entendons par résolution angulaire et radiale des données : la résolution angulaire décroît linéairement avec la distance au capteur, tandis que la résolution radiale obéit à une loi plus complexe. Avec les notations précisées sur la figure, on a :

$$R_a = 2r \tan \frac{\Delta\theta}{2}$$

$$\text{et } R_r = \tan \Delta\phi \left(\frac{r^2 + H^2}{H + r \tan \Delta\phi} \right) \quad (3.1)$$

Figure 3.4 : Vue perspective d'une image de points 3D acquise en environnement intérieur, avec une résolution d'un demi-degré suivant les deux axes. Le trièdre représente le repère lié au capteur, la longueur de chacun de ses axes est de 1 mètre. On distingue aisément les points correspondant au sol : leur résolution est très grande au pied du capteur, et décroît en s'éloignant. Une planche verticale, orthogonale à l'axe de visée, ainsi qu'un carton partiellement perçu sur la droite du champ de vue se caractérisent par une forte densité de données. Il est plus difficile de deviner de quoi est composé le fond de la scène.

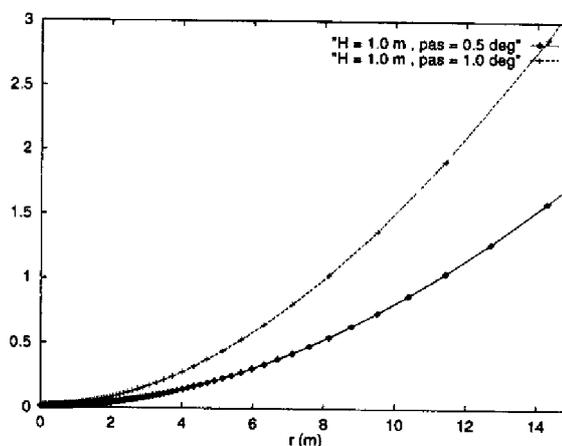
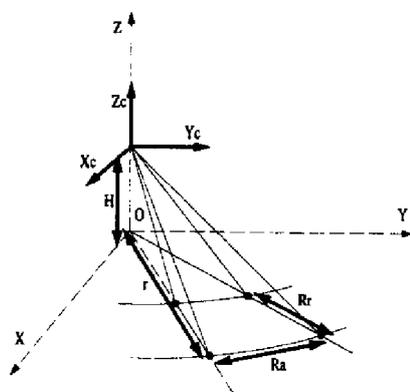
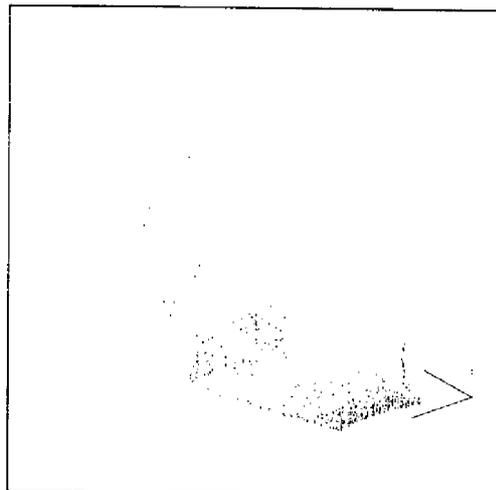


Figure 3.5 : Définition des résolutions radiale (R_r) et angulaire (R_a), et résolution radiale sur un sol plan

L'équation 3.1 qui lie la résolution radiale R_r à la hauteur H du capteur au sol et à la distance r des points mesurés est intéressante : elle est représentée sur la figure 3.5, avec une valeur de H égale à 1.0 mètre, et pour des pas angulaires de 1 et 0.5 degrés (les points situés sur les courbes correspondent aux points qu'un capteur retournerait avec de tels paramètres). Il apparaît immédiatement que la densité des données décroît très vite avec la distance au capteur : ainsi, à 8 mètres du capteur, l'écart de distance entre deux points correspondant à deux valeurs successives de l'angle de site ϕ est de 50 cm pour une résolution de 0.5° , et de près de 1 mètre pour une résolution de 1.0° . On voit bien la difficulté d'interpoler de telles données sur une grille cartésienne de faible résolution.

On pourrait bien entendu commander le balayage en site du capteur avec une loi qui donnerait une résolution radiale constante. Mais une telle loi est complexe², sans compter qu'elle ne résout pas le problème de la résolution angulaire en azimut. De plus, il ne faut pas oublier que l'environnement n'est que rarement plat : les images auraient toujours une résolution non-uniforme.

²Pour avoir une résolution radiale R_r constante sur un sol horizontal, les pas angulaires en site sont donnés par la formule : $\Delta\phi = R_r \tan^2 \phi \frac{(1 + \tan^2 \phi)}{1 - R_r \tan \phi + \tan^2 \phi}$

d) Incertitudes et erreurs

Il existe toujours une incertitude sur chacune des composantes du vecteur de mesure (θ, ϕ, ρ) associé à un point perçu par le télémètre. Si les incertitudes sur les valeurs de θ et ϕ ne dépendent que du capteur et peuvent être déterminées par une procédure de calibration statistique, les incertitudes sur ρ sont bien plus délicates à estimer. En effet, un nombre considérable de paramètres influent sur cette mesure : elle dépend de l'angle d'incidence du faisceau par rapport à la surface perçue, de la couleur et de la matière (réflectance) de cette surface, de la température du capteur... De nombreux auteurs ont analysé ces incertitudes [Hebert 91, Hoffman 91, Nashashibi 93] : il s'agit d'un problème complexe, d'autant plus que ces incertitudes dépendent fortement de l'environnement, et non seulement du capteur lui-même. En pratique, un modèle simple des incertitudes sur les données permet de s'affranchir d'une telle étude, mais sa propagation à travers les diverses transformations et traitements reste tout de même coûteuse en temps de calcul [Nashashibi 93].

Le télémètre retourne aussi des données erronées : des non-détections (aucune mesure retournée) se produisent par exemple lorsque l'angle d'incidence entre le rayon et la surface perçue est très faible, ou lorsque cette surface est humide. Par ailleurs, le diamètre du faisceau n'est pas négligeable et augmente avec la distance au capteur (par diffraction dans l'air), et il est fréquent qu'il intersecte deux entités de l'environnement différentes et distantes ; la valeur retournée est alors une moyenne, qui ne correspond à aucune primitive réelle. Enfin, nous avons remarqué dans les images acquises avec notre capteur des artefacts (mesures totalement incohérentes) dont nous n'expliquons pas l'origine. La plupart des ces artefacts se produisent heureusement sur des points isolés, et un simple filtrage médian de l'image acquise permet souvent de s'en débarrasser.

D'une manière générale, retenons que les erreurs sont plus nombreuses et les données plus imprécises à mesure que l'on s'éloigne du capteur.

3.3.2 Stéréovision par corrélation

a) Principe de fonctionnement

Le principe d'un capteur de points 3D basé sur un algorithme de stéréovision par corrélation est analogue à tout algorithme de stéréovision : mais ce sont directement les pixels des images stéréoscopiques qui sont appariés, et non des primitives (segments ou régions) extraites de ces images. Pour faciliter la mise en correspondance, une procédure de rectification des images est appliquée [Ayache 88] : cela permet de limiter la recherche d'appariement d'un pixel d'une image à une seule ligne de l'autre image. Le pixel apparié est celui qui maximise un score de corrélation calculé pour une fenêtre qui l'encadre, et à chaque paire de pixels appariés est associée une valeur de disparité d , qui représente la distance entre les deux pixels. Une procédure de triangulation permet de convertir les caractéristiques (i, j, d) d'un pixel en coordonnées (x, y, z) du point de l'environnement correspondant. Les améliorations de cette méthode et la puissance des processeurs de calcul rendent désormais cette procédure de corrélation relativement rapide : les "images stéréo" calculées au CNES [Hotz 91] sur lesquelles nous avons travaillé sont produites en 25 secondes avec une résolution de 512 pixels, et en 4 secondes avec une résolution de 128 pixels sur une Sparc Station 10.

La figure 3.6 présente une image de la caméra gauche du ban stéréoscopique acquise sur le site Geroms³, et les pixels qui ont effectivement été corrélés : le secteur vide sur la gauche de l'image correspond à la zone qui n'est pas perçue par la caméra de droite, et pour les mêmes raisons il existe une zone "d'ombre" derrière chaque obstacle. Remarquons que de nombreux pixels en bas à droite de l'image n'ont pas été corrélés (pixels blancs), alors qu'ils sont perçus par les deux caméras : il s'agit de non-détections dues au fait que l'image est sombre et peu contrastée sur cette zone (les figures ont été éclaircies pour des raisons de visibilité). L'image de points 3D résultante est néanmoins de très bonne qualité, et près de 86 % pixels ont été corrélés.

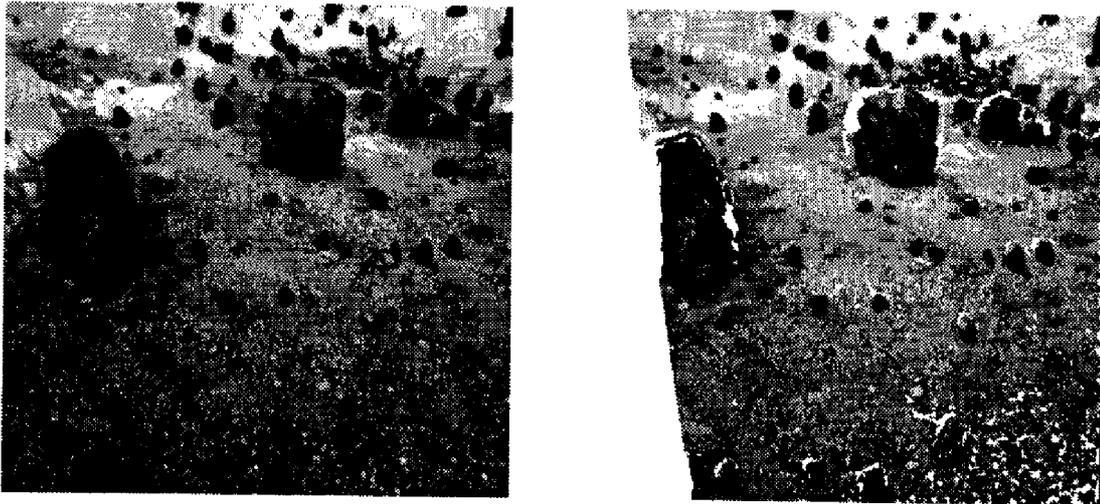


Figure 3.6 : Image stéréoscopique gauche, et pixels effectivement corrélés

La figure 3.7 montre l'image de points 3D correspondante, sous-échantillonnée d'un facteur 4 suivant les lignes et les colonnes pour des raisons de visibilité.

b) Géométrie

Les caractéristiques géométriques des caméras définissent un mode de balayage *corrélé* : la rotation en azimut se fait autour d'un axe qui dépend de l'angle de site (figure 3.8)⁴. Cette fois, les points d'un sol plan acquis pour un angle de site constant sont répartis suivant une droite, parallèle ou non au plan image, suivant que l'assiette de la caméra est horizontale ou non.

C'est la matrice de *calibration intrinsèque* de la caméra considérée qui permet de définir les angles de site et d'azimut associés à un pixel (i, j) de l'image. Si l'effet optique de distorsion est négligé, cette matrice est celle d'une projection perspective, qui associe à tout point perçu $P = (x, y, z)$ les indices (i, j) correspondant dans la matrice de pixels (ou plan image - figure 3.9). Il s'agit d'un modèle "sténopé" de la caméra (*pin-hole* en anglais).

³Geroms (Groupement pour les Essais en RObotique Mobile Spatiale) est un groupement qui comprend le CNES, le CERT et le LAAS, dont l'objectif est de proposer une base logistique d'expérimentation pour la robotique mobile en environnement naturel. Dans ce cadre, un site d'expérimentation de plus de 6000 m² a été construit au CNES

⁴Notons qu'il existe des systèmes mécaniques de déflexion d'un rayon laser qui effectuent un balayage en mode corrélé (ainsi le télémètre 2-axes qui équipait Hilare I)

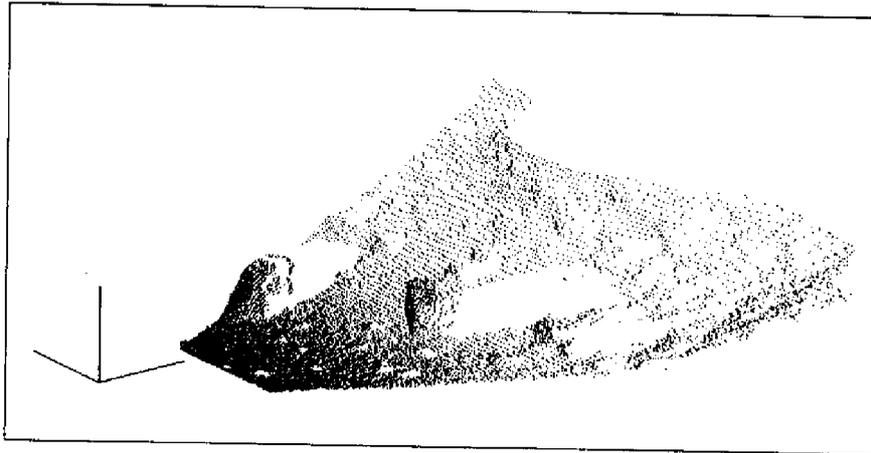


Figure 3.7 : Points 3D correspondant à l'image de la figure 3.6

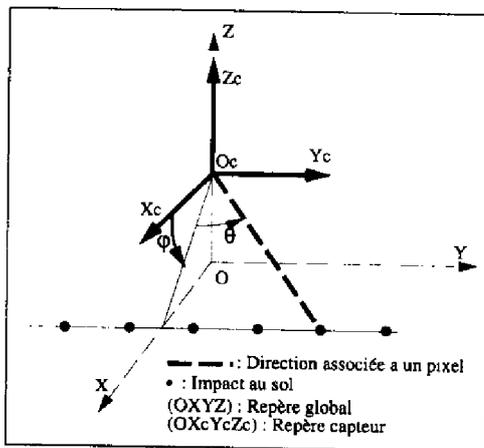


Figure 3.8 : Mode de balayage corrélé

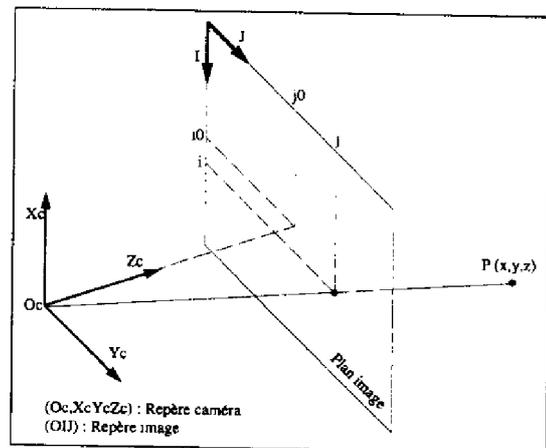


Figure 3.9 : Modèle sténopé d'une caméra

Une telle matrice s'écrit $\Pi = \begin{pmatrix} \alpha_i & 0 & i_0 & 0 \\ 0 & \alpha_j & j_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$; et on a donc : $\begin{cases} i = \alpha_i \frac{z}{x} + i_0 \\ j = \alpha_j \frac{z}{x} + j_0 \end{cases}$.

Les angles de site et d'azimut correspondants à un pixel (i, j) sont alors déterminés par :

$$\begin{cases} \theta = \arctan\left(\frac{i-j_0}{\alpha_j}\right) \\ \phi = \arctan\left(\cos \theta \cdot \frac{i-i_0}{\alpha_i}\right) \end{cases} \quad (3.2)$$

Dans le cas où la distorsion de l'objectif est trop importante pour être négligée, les coordonnées des pixels sont corrigées par un facteur proportionnel au carré de la distance du pixel au centre (i_0, j_0) [Orteu 91].

c) Résolution

Les propriétés de résolution angulaire d'un système stéréoscopique sont analogues à celles d'un télémètre laser 2-axes : la figure 3.10 représente la résolution radiale pour la colonne

médiane de cette même caméra, située à 1.0 mètre du sol et orientée de 15 degrés vers le bas, pour différentes résolutions image. La forme des courbes est la même que pour le télémètre laser, mais la résolution est bien meilleure. Le champ de vue vertical de la caméra considérée est de $28,5^\circ$, ce qui explique la présence d'une "zone aveugle" au pied du capteur (jusqu'à deux mètres) pour laquelle aucune donnée n'est mesurée.

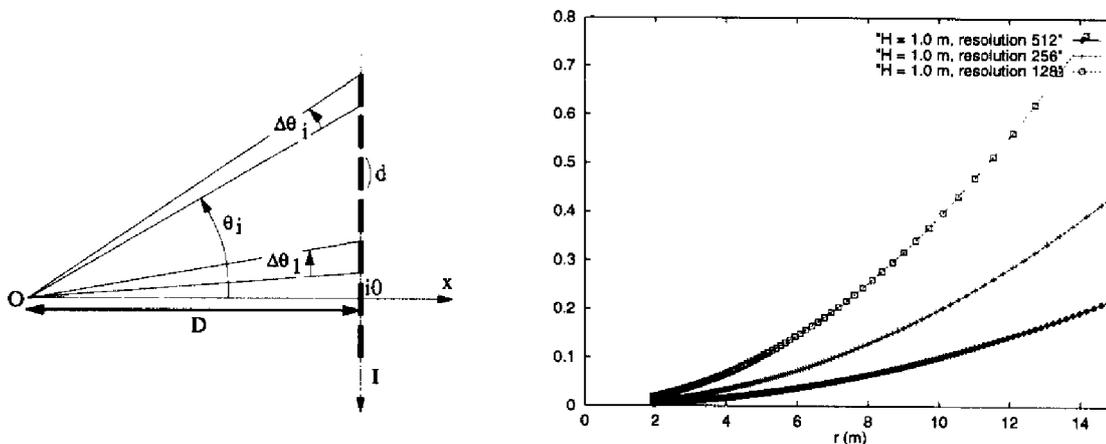


Figure 3.10 : Résolution angulaire d'une caméra, résolution radiale sur un sol plan

Remarque : La matrice de pixels d'une caméra étant située sur un plan, tous les pixels ne sont pas vus depuis le centre optique sous le même angle, et la résolution angulaire n'est donc pas constante. Ainsi, avec le banc stéréoscopique d'Adam qui est composé de deux caméras dont la focale est de 8mm , le plus grand angle sous lequel est vu un pixel de la colonne du milieu vaut 0.0588 degrés (pour l'indice i égal à i_0), et le plus petit vaut 0.0539 degrés. Cela représente une variation de l'ordre de 8% , que nous ne pouvons négliger. Notons aussi que la résolution radiale dépend de la colonne de pixels considérée (présence du facteur $\cos\theta$ dans l'équation 3.2), et qu'elle se dégrade sur les bords de l'image.

d) Incertitudes et erreurs

Les images de points 3D résultant de l'appariement des pixels ne sont bien entendu pas parfaites : il existe de nombreuses sources d'erreurs et d'incertitudes tout au long de la procédure, dont l'influence est délicate à déterminer [Matthies 87, Matthies 92]. Citons les incertitudes sur les matrices de calibration des caméras et du banc stéréoscopique, celles qui sont liées à la taille des pixels et à la correction de la distorsion des objectifs, l'incertitude sur l'estimation de la valeur de la disparité, et enfin les erreurs d'appariements. Notons aussi que le nombre de non-détections (absence d'appariement, donc de points 3D) peut être considérable si la scène est faiblement texturée : c'est notamment ce qui limite l'application de telles procédures dans des environnements intérieurs.

Chacun de ces facteurs a une influence sur la précision des coordonnées des points 3D, qui décroît bien sûr avec leur distance au capteur.

3.3.3 Conclusions

Cette brève présentation des caractéristiques de nos capteurs montre qu'ils sont des appareils de mesure riches, mais complexes et loin d'être parfaits. Ainsi, lorsque l'on désire construire un modèle géométrique précis de l'environnement, il est indispensable de considérer et de propager les erreurs sur les mesures dans toutes les opérations effectuées sur les données. Mais le modèle de ces erreurs est complexe à déterminer précisément, et leur prise en compte est délicate et alourdit considérablement les calculs. Par ailleurs, nous considérons comme essentielles les propriétés de résolution des capteurs : la qualité des mesures obtenues n'est jamais uniforme sur la zone perçue. Nous considérons donc que nos capteurs sont "myopes", et cet aspect doit bien sûr apparaître dans toutes les représentations que l'on construit à partir de leurs données ; cela a beaucoup influencé le développement de notre procédure de classification.

3.4 Méthode proposée

La plupart des techniques de classification présentées dans le paragraphe 3.2 sont basées sur des attributs calculés localement pour chaque point, qu'il provienne d'une image brute ou d'un MNT. Elles sont donc très sensibles aux incertitudes et erreurs du capteur considéré. De plus, la propriété de résolution des capteurs est rarement considérée de manière explicite (mais plutôt par le biais des incertitudes), en particulier pour les méthodes basées sur un MNT.

Pour le développement de notre méthode, nous ne pouvons pas ignorer ces deux aspects essentiels de nos capteurs ; mais la complexité des modèles d'erreurs et la contrainte de rapidité d'exécution font qu'il nous est difficile de propager *explicitement* les incertitudes. Comme nous désirons construire une description qualitative de la zone perçue, nous avons favorisé une approche *globale* : des attributs sont déterminés pour un ensemble de points, et leur analyse va permettre de déterminer à quelle classe de terrain cet ensemble correspond. Une approche similaire d'analyse d'images de points 3D peut être trouvée dans [Langer 94], et son extension à des lignes de points acquises en mouvement dans [Hebert 94]. Le fait que les attributs soient déterminés globalement les rend très peu sensibles au bruit sur les mesures mais n'empêche pas de le considérer, et nous allons voir que la discrétisation retenue pour déterminer les ensembles de points prend en compte la propriété fondamentale de résolution des capteurs.

Observons la figure 3.11 : il s'agit de la projection orthogonale suivant la verticale de l'image laser de la figure 3.4. Nous y distinguons encore aisément les zones qui correspondent au sol, ainsi que quelques obstacles : la planche verticale au centre et le carton à droite du champ de vue se caractérisent notamment par une forte densité de points. Quant aux zones occultées par ces obstacles, elles se caractérisent par une très faible densité de points.

La densité des points sur cette image apparaît clairement comme un facteur essentiel dans notre processus d'interprétation, et c'est ce qui nous a guidé pour classifier les images : le principe de notre méthode repose sur une discrétisation de la zone perçue en cellules (ou *pavés*), qui met en évidence cet attribut de "densité". Cette discrétisation est présentée

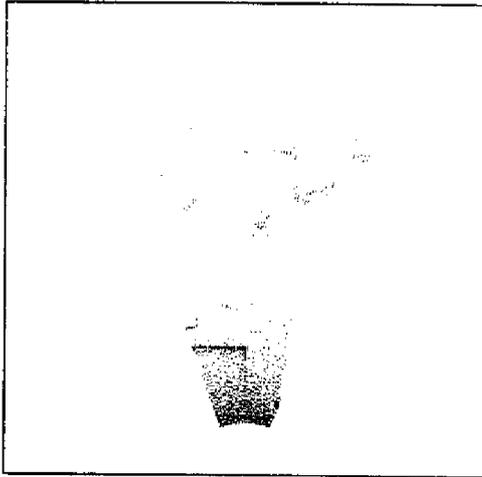
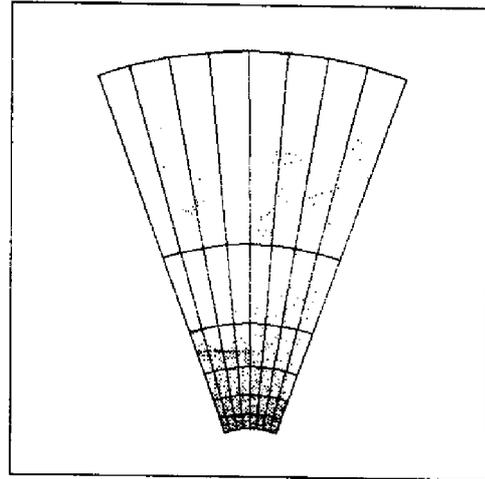
Figure 3.11 : *Projection orthogonale des points*Figure 3.12 : *Discrétisation associée*

figure 3.12 : les cellules ont une taille croissante avec leur distance au capteur, mais elles contiennent le même nombre de points lorsqu'elles couvrent une partie plane du sol⁵. Cet attribut de densité, ainsi que d'autres attributs *moyennés* sur l'ensemble des points d'une cellule vont être analysés de façon à pouvoir déterminer la classe à associer à chaque cellule.

En ce sens, nous abordons le problème comme un problème de classification "canonique" : l'étude des informations volumineuses contenues dans une image de points 3D se ramène à une étude dans un espace d'attributs réduit, dans lequel une partition correspondant aux classes désirées est établie. Nous présentons l'application d'une méthode de classification bayésienne classique pour les images stéréo (paragraphe 3.6), et une méthode simple de classification basée sur des seuils pour les images laser (paragraphe 3.7).

3.5 Définition et construction des cellules

3.5.1 Discrétisation adoptée

La discrétisation que nous utilisons correspond en fait à la projection sur le sol d'une discrétisation régulière (grille cartésienne) dans le repère lié au capteur (figure 3.13). Elle a le grand avantage de n'être définie que par la géométrie du capteur, et est donc "naturelle" pour le capteur : elle respecte sa résolution et n'introduit aucune régularité artificielle, par opposition à une discrétisation cartésienne de la zone perçue par exemple.

a) Cas du télémètre 2-axes

Chaque cellule est définie dans le repère capteur par un couple d'angles minimaux et maximaux en azimut ($\theta_{min}, \theta_{max}$), et par un couple d'angles minimaux et maximaux en site (ϕ_{min}, ϕ_{max}), avec les relations :

⁵Un point "appartient" à une cellule si et seulement si sa projection verticale est comprise dans les limites de cette cellule

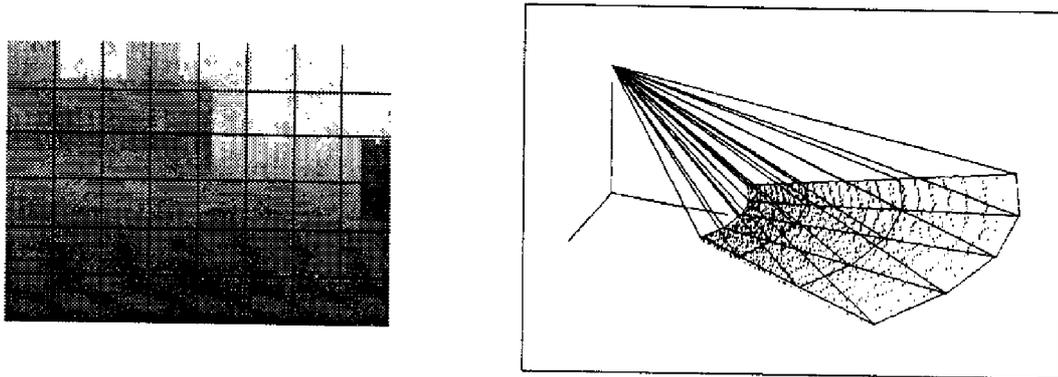


Figure 3.13 : Définition de la discrétisation adoptée. La figure de gauche représente sous la forme d'une image de luminance dans le repère capteur l'image laser des figures 3.4 et 3.11 : les coordonnées (i, j) d'un pixel de l'image correspondent aux angles en site et azimut (ϕ, θ) d'un point 3D, et la distance ρ de ce point au capteur est convertie en niveaux de gris. La discrétisation dans ce repère est une grille régulière, et la figure de droite montre sa projection dans le repère lié au sol

$$\begin{cases} \theta_{max} = \theta_{min} + \Delta_{\theta} \cdot \tau_{\theta} \\ \phi_{max} = \phi_{min} + \Delta_{\phi} \cdot \tau_{\phi} \end{cases}$$

où Δ_{θ} et Δ_{ϕ} sont les pas d'incrément en azimut et en site, et τ_{θ} et τ_{ϕ} sont les taux de discrétisation en azimut et en site. Ces derniers paramètres définissent la résolution de la cellule.

Le repère R_p dans lequel est exprimée la projection des cellules est défini comme suit (figure 3.14) : son assiette est parallèle à celle du sol, et son origine est située sur la verticale liée au capteur, à l'élévation supposée du sol. Le choix de ce repère permet d'exprimer l'intersection d'un plan décrit par des rayons émis à un angle d'azimut constant avec le plan (O_p, X_p, Y_p) par une droite qui passe par O_p . En effet, l'ensemble des plans définis par les angles d'azimut forme un faisceau de plans qui s'intersectent suivant la verticale liée au capteur : l'intersection de ce faisceau de plans avec (O_p, X_p, Y_p) définit donc un faisceau de droites, dont le point d'intersection est $(O_p, X_p, Y_p) \cap (O_c, Z_c)$, soit l'origine O_p .

Les rayons émis à un angle de site donné décrivent un cône, et ceux émis à un angle d'azimut donné décrivent un plan : lorsque la verticale liée au capteur correspond à celle du monde, la projection au sol des cellules est donc décrite par deux arcs de cercle et deux segments de droite. Dans le repère R_p , une cellule est alors représentée en coordonnées cylindriques par un couple d'angles minimaux et maximaux en azimut $(\theta_{cyl,min}, \theta_{cyl,max})$, et par un couple de rayons minimaux et maximaux (r_{min}, r_{max}) . Le test d'appartenance d'un point perçu à une cellule se réduit alors à de simples comparaisons sur ses coordonnées cylindriques exprimées dans le repère R_p .

Par contre, lorsque l'assiette du capteur est quelconque, la projection au sol des cellules est définie par deux arcs d'ellipse et deux segments de droite. Afin de conserver un test d'appartenance d'un point à une cellule simple, les ellipses définies par un angle de site constant sont approximées par une suite d'arcs de cercle centrés en O_p (figure 3.15).

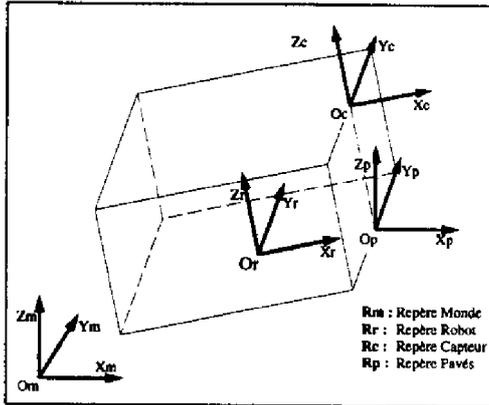


Figure 3.14 : Définition du repère R_p par rapport aux repères R_m , R_r et R_c

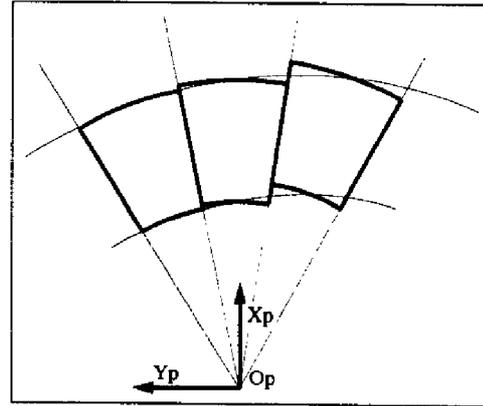


Figure 3.15 : Un arc d'ellipse est approximé par une séquence d'arcs de cercle

Une cellule est donc définie dans le repère R_p par deux couples de coordonnées. Ils sont déterminés en calculant l'intersection avec le plan $z_p = 0$ des quatre droites déterminées par la projection des quatre points qui la définissent dans le repère capteur R_c . Dans ce repère, l'équation paramétrique d'une droite correspondant au rayon d'angle d'azimut θ et d'angle de site ϕ s'écrit (elle passe par l'origine de R_c) :

$$\begin{cases} x_c = k \cos \phi \cos \theta \\ y_c = k \cos \phi \sin \theta \\ z_c = k \sin \phi \end{cases}$$

Grâce à la matrice de passage⁶ $T_{R_c \rightarrow R_p} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$, cette équation s'écrit

dans le repère R_p :

$$\begin{cases} x_p = k(R_{11} \cos \phi \cos \theta + R_{12} \cos \phi \sin \theta + R_{13} \sin \phi) + T_x \\ y_p = k(R_{21} \cos \phi \cos \theta + R_{22} \cos \phi \sin \theta + R_{23} \sin \phi) + T_y \\ z_p = k(R_{31} \cos \phi \cos \theta + R_{32} \cos \phi \sin \theta + R_{33} \sin \phi) + T_z \end{cases} \quad (3.3)$$

L'équation $z_p = 0$ permet de déterminer le paramètre k , et donc de calculer les coordonnées x_p et y_p , d'où l'on déduit les coordonnées cylindriques (r, θ_{cyl}) . L'approximation d'un arc d'ellipse par un arc de cercle est faite en prenant comme valeur du rayon la moyenne des rayons déterminés par les droites (θ_{min}, ϕ) et (θ_{max}, ϕ) .

b) Cas de la stéréovision

Dans ce cas, le principe de détermination des coordonnées des cellules est tout à fait similaire : une cellule est définie dans le repère capteur par deux couples (i_{min}, i_{max}) et

⁶Dans tout le manuscrit, les matrices homogènes de transformation d'un repère à l'autre sont notées avec la convention suivante : $T_{A \rightarrow B}$ est la matrice qui permet de calculer les coordonnées d'un objet exprimées dans le repère B à partir de ses coordonnées dans le repère A

(j_{min}, j_{max}) , et l'équation 3.2 permet de leur associer deux couples d'angles. L'équation 3.3 donne alors les coordonnées de la projection d'une cellule dans le repère R_p . Mais le mode de balayage associé à une caméra impose cette fois de représenter les coordonnées des cellules dans R_p par deux couples $(\theta_{cyl_{min}}, \theta_{cyl_{max}})$ et (x_{min}, x_{max}) . Lorsque l'assiette de la caméra dans le repère de laquelle sont exprimés les points 3D n'est pas horizontale, la droite correspondant à l'intersection d'un plan défini par une valeur de i avec (O_p, X_p, Y_p) n'est plus parallèle à l'axe (O, Y_p) ⁷. Elle est alors approximée par une séquence de segments parallèles à cet axe, de façon à ce que le test d'appartenance d'un point à une cellule puisse toujours être réalisé par une simple comparaison de ses valeurs "cylindro-cartésiennes" et des bornes qui définissent la cellule. La figure 3.16 présente l'allure de la discrétisation de la zone perçue lorsque les caméras ont un angle de roulis non nul.

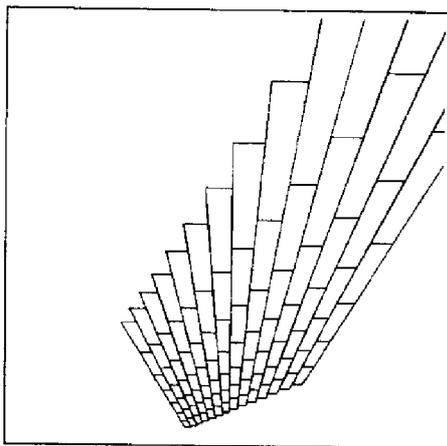


Figure 3.16 : Discrétisation de la zone perçue par une caméra avec un roulis de 12°

3.5.2 Estimation des attributs d'une cellule

Une fois que la discrétisation de la zone perçue est établie, une boucle sur l'ensemble des points 3D permet d'affecter à chaque cellule la liste des points qu'elle contient. La définition des cellules et la simplicité des tests d'appartenance rendent cette procédure de remplissage extrêmement rapide. Afin d'associer à chaque cellule une classe de terrain, des attributs moyens sont calculés sur l'ensemble de ses points. Ces attributs seront d'autant moins sensibles aux incertitudes et erreurs sur les mesures que le nombre de points d'une cellule est grand.

Nous avons retenu les attributs suivants :

Densité relative

Considérons que le capteur perçoit un sol parfaitement horizontal. Alors la projection sur le plan ($z_p = 0$) d'un point 3D dont les coordonnées (θ, ϕ) appartiennent à un certain pavé dans le repère capteur est située à l'intérieur de la projection au sol de ce même pavé, et ce quelle que soit l'attitude du capteur par rapport au sol. Cela implique que les cellules couvrant le sol ont un nombre de points constant, qui ne dépend que des taux de discrétisation τ_{az}

⁷En toute rigueur, seul un angle de tangage non nul produit ce phénomène.

et τ_{site} . Ce nombre est le *nombre nominal* N_{nom} des pavés, et l'on a de façon évidente : $N_{nom} = \tau_\theta \cdot \tau_\phi$. La densité relative d_r d'une cellule est définie comme étant le rapport du nombre de points qu'elle contient par son nombre nominal : $d_r = \frac{N}{N_{nom}}$.

Comme nous l'avons vu dans le paragraphe 3.4, cet attribut de densité est particulièrement intéressant : lorsqu'il est nul ou très faible, il signifie que la cellule considérée correspond à une zone occultée ou pour laquelle le capteur n'a pas retourné d'informations (cas d'une surface spéculaire pour le télémètre ou d'une surface non texturée pour la stéréovision par exemple) ; inversement, lorsqu'il est très élevé, il indique la présence d'un obstacle. Cette propriété est illustrée en 2 dimensions par la figure 3.17.

Cependant, une densité relative égale à l'unité n'implique pas nécessairement que la cellule correspond à un sol plan : ainsi la perception de surfaces pentues situées sous le plan ($z_p = 0$) peut donner une densité égale à la densité nominale ; de même, la perception d'un sol plan situé plus haut que le plan $z_p = 0$ donne une densité inférieure à 1. Cet attribut est donc ambigu, mais il reste néanmoins très discriminant, et est d'autant moins sensible au bruit que le nombre nominal de la cellule est grand.

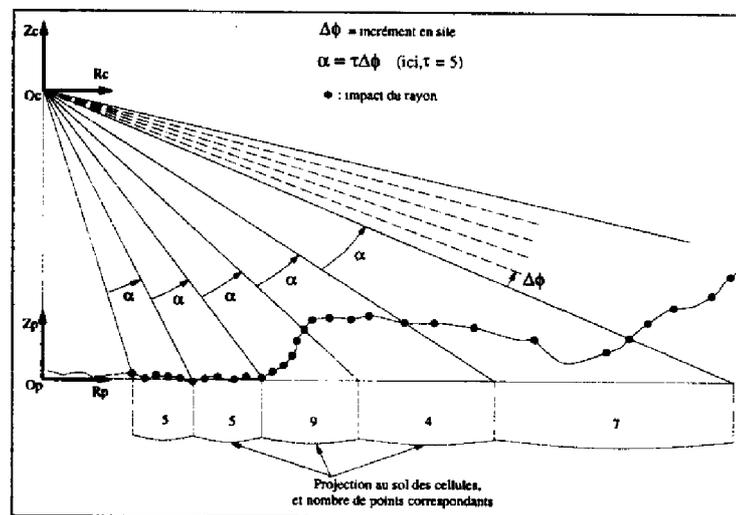


Figure 3.17 : Comportement de l'attribut "densité relative" en fonction du terrain perçu (cette figure est une coupe du terrain selon un plan vertical défini par un angle d'azimut constant)

Variance sur l'altitude des points

La variance σ_z^2 sur l'altitude des points d'une cellule est rapidement calculée, et est une très bonne indication sur le caractère plan du terrain correspondant à la cellule : ainsi une variance quasi-nulle est caractéristique d'une zone plane. Par contre, une grande valeur de cette variance ne permet pas de décider si le terrain est simplement pentu ou accidenté.

Variation maximale de l'altitude des points

L'écart d'altitude maximum sur tous les points d'une cellule peut être un attribut intéressant. Il permet notamment de déduire rapidement si une cellule correspond à une zone obstacle ou plane. Mais il est extrêmement sensible aux erreurs du capteur : un seul point totalement erroné peut suffire à le rendre très grand, et donc conduire à un étiquetage incohérent. Nous

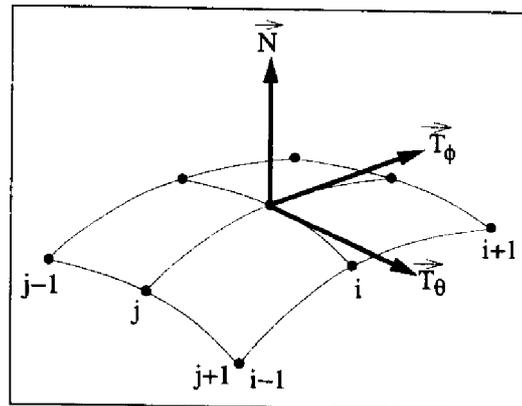
l'avons cependant retenu, en considérant que ces erreurs sont faciles à filtrer.

Vecteur normal moyen

Les coordonnées du vecteur normal de la surface correspondant à un point 3D sont aisément déterminables (figure 3.18). Elles fournissent une bonne information sur la pente moyenne de la cellule considérée : en particulier, la coordonnée suivant l'axe (O_p, Z_p) du repère R_p est égale au cosinus de l'angle du vecteur avec la verticale.

Figure 3.18 : Calcul du vecteur normal en un point d'une image laser : le vecteur tangent à la courbe décrite par les points à θ constant est déterminé par un masque de dérivation suivant les indices i sur les trois coordonnées cartésiennes des points : $\vec{T}_\theta = (\frac{\partial x}{\partial \theta} \ \frac{\partial y}{\partial \theta} \ \frac{\partial z}{\partial \theta})^T$. On procède de même pour déterminer \vec{T}_ϕ , et le vecteur normal au point considéré est donné par :

$$\vec{N} = \frac{\vec{T}_\theta \wedge \vec{T}_\phi}{\|\vec{T}_\theta \wedge \vec{T}_\phi\|}$$



Variances sur les coordonnées du vecteur normal moyen

Les seules coordonnées du vecteur normal moyen ne suffisent pas à déterminer si la cellule correspond à une partie de terrain plane, pentue ou accidentée, mais leur variance permet de lever ce doute. Ainsi, un vecteur normal moyen dont l'angle avec la verticale n'est pas négligeable et dont la variance sur les trois coordonnées est très faible est caractéristique d'une pente régulière, et un vecteur normal moyen dont les variances sont grandes est représentatif d'une zone de terrain très irrégulière. Afin de limiter le nombre d'attributs, nous ne considérons pas les 3 variances sur les coordonnées du vecteur normal, mais la "norme" de l'écart type sur les trois coordonnées.

Distance de la cellule au capteur

Cet attribut est un peu particulier, dans le sens où il n'indique aucune caractéristique du terrain correspondant à une cellule : il n'y a absolument aucune corrélation entre la distance à laquelle est perçu le terrain et sa nature. Par contre, elle est un excellent indicateur de la fiabilité des mesures en raison des caractéristiques du capteur concernant la résolution et les incertitudes, et la valeur des divers attributs précédents est fortement corrélée à cette distance (voir annexe A).

En conclusion, nous retenons 6 attributs scalaires différents : la densité relative d'une cellule, la variance sur l'altitude des points qu'elle contient, la variation maximale d'altitude, le cosinus de l'angle de son vecteur normal moyen avec la verticale, la norme des écarts types des coordonnées de ce vecteur normal, et la distance de la cellule au capteur. Le vecteur d'attributs est noté : $A = (d_r, \sigma_z, \Delta Z, N_z, \sigma_N, D_c)^T$.

Ces différents attributs ne sont pas totalement indépendants, mais les corrélations qui les lient ne sont pas toujours faciles à établir. On a donc intérêt à les considérer tous, afin

d'améliorer la qualité de la procédure de classification. Enfin, remarquons qu'aucun d'entre eux n'est caractéristique de la nature physique du terrain : seule la variance sur les coordonnées du vecteur normal, qui donne une indication sur la rugosité du terrain, pourrait peut-être aider à distinguer différentes classes de nature physique (à la condition que les données soient denses et peu bruitées). Mais la rugosité ne suffit pas à caractériser la nature du terrain : ainsi une partie de sable lisse et faiblement cohérent (donc dangereux) ne se distingue pas par sa rugosité d'une zone de terre battue praticable. Nous sommes convaincus que des attributs non géométriques, tels que la luminance (à partir de laquelle peuvent être déterminés des attributs de teinte, de texture...), la réflectance ou des caractéristiques liées à la réflexion d'ondes électromagnétiques (radars) sont nécessaires pour pouvoir déterminer sa nature physique.

3.5.3 Affinement de la discrétisation

Le seul inconvénient de la discrétisation proposée est sa résolution : ainsi, une cellule plane partiellement occupée par un obstacle sera étiquetée "obstacle" ou "accidentée" car ses attributs ne seront pas caractéristiques d'une surface plane. Cela aura pour effet de sur-contraindre le problème, en éliminant des passages possibles pour le robot. Nous avons donc défini un niveau inférieur de discrétisation, qui est obtenu en scindant les cellules en quatre sous-cellules, à la manière d'un quad-tree. Cet affinement n'est pas systématiquement nécessaire, la décision de son activation sera présentée dans les paragraphes 3.6.3 et 3.7. Les coordonnées et attributs d'une sous-cellule sont déterminés exactement de la même manière que ceux d'une cellule.

Cependant, pour que les attributs d'une cellule soient pertinents, il doivent être définis sur un grand nombre de points, sans quoi ils n'auront plus aucune valeur statistique. Nous ne pouvons donc pas répéter le processus de scission sur une sous-cellule, car s'il conduirait à une description très fine de la zone perçue, les résultats de la classification n'auraient plus beaucoup de sens. Il y a donc un juste milieu à trouver pour déterminer les taux de discrétisation τ_θ et τ_{phi} : ceux-ci doivent être suffisamment petits pour produire une description fine, mais suffisamment grands pour que les attributs des cellules qu'ils définissent soient statistiquement valables. Ce juste milieu dépend bien entendu de la résolution du capteur, et il apparaît immédiatement que les résultats de la classification seront d'autant précis que les données ont été acquises avec une bonne résolution.

3.6 Classification bayésienne d'images stéréo

3.6.1 Principe de la classification bayésienne

La classification bayésienne est une approche statistique puissante, qui permet d'affecter à chaque échantillon observé une probabilité d'appartenance à chacune des classes considérées. Cette classification est basée sur (i) les attributs mesurés, (ii) les probabilités *a priori* sur la répartition des classes, et (iii) sur les fonctions de densité de probabilité conditionnelle des attributs [Andrews 72, Postaire 87]. Toute procédure de classification bayésienne est réalisée

en trois étapes :

1. Phase d'apprentissage supervisé : elle permet de déterminer à partir d'un ensemble d'échantillons prototypes (auxquels un expert a associé une classe) les lois de probabilités qui donnent la valeur des attributs pour chaque classe considérée. En notant $A = (a_1, \dots, a_m)^T$ le vecteur d'attributs et $\{C_1, \dots, C_n\}$ les classes possibles, ces probabilités sont notées $P(A | C_i)$. Nous verrons dans le paragraphe suivant qu'elles sont dans notre cas difficiles à déterminer hors ligne, et qu'il est alors nécessaire de les *estimer* pour chaque mesure d'attribut ;
2. Estimation des probabilités conditionnelles *a posteriori* : la mesure du vecteur d'attribut permet d'estimer la probabilité d'être en présence de chacune des classes grâce à la formule de Bayes :

$$P(C_i | A) = \frac{P(A | C_i)P(C_i)}{P(A)} = \frac{P(A | C_i)P(C_i)}{\sum_{i=1}^n P(A | C_i)P(C_i)} \quad (3.4)$$

3. Prise de décision : une fois connues les probabilités conditionnelles de chacune des classes en fonction des attributs observés, la décision optimale est celle qui minimise le *risque* bayésien. En notant $r(C_s | C_i)$ la pénalisation subie lorsqu'un échantillon de classe C_i est classé C_s , le risque associé à une décision C_s est égal à :

$$R(C_s | A) = \sum_{i=1}^n r(C_s | C_i)P(C_i | A)$$

et donc :

$$C_{opt} = \text{Argmin}_{C_i}(R(C_i | A))$$

Les seules difficultés de cette procédure concernent l'estimation de bonnes fonctions de pénalisation pour la loi de décision (paragraphe 3.6.3), et surtout l'estimation des fonctions de densité de probabilité conditionnelle $P(A | C_i)$ (paragraphe 3.6.2).

3.6.2 Phase d'apprentissage supervisé

a) Prototypage

Nous disposons de 70 paires d'images stéréoscopiques qui ont été acquises sur le site d'expérimentation Geroms, et corrélées à la résolution image (512 pixels par ligne et colonne) grâce aux algorithmes développés au CNES. Ces images sont d'excellentes qualité, et très représentatives des divers types de terrains naturels auxquels un robot peut être confronté (à la nature physique près, le site Geroms n'étant constitué que de rochers et cailloux posés sur un gravier de pouzzolane, et de pentes couvertes de béton peint en gris). Nous avons retenu une dizaine d'images prototypes, qui couvrent les diverses configurations de terrain du site Geroms, et que nous avons sous-échantillonnées d'un facteur 2 suivant les lignes et les colonnes. Nous nous sommes ainsi placés dans un contexte réaliste, la procédure de

corrélation étant très coûteuse en temps lorsqu'elle s'effectue sur des images de 512 par 512 pixels⁸.

La discrétisation des cellules est effectuée pour chacune de ces images : nous avons choisi les valeurs suivantes des taux de discrétisation : $\tau_{theta} = 18$ et $\tau_{phi} = 14$, ce qui donne un nombre nominal N_{nom} égal à 252 pour les cellules. Le taux de discrétisation est plus faible suivant l'axe de site que suivant l'axe d'azimut à cause des propriétés de résolution des caméras : ainsi les cellules ont une forme moins "effilée", et la zone perçue est assez finement discrétisée suivant la direction de visée du capteur.

Une fois la discrétisation établie, *tous* les attributs sont estimés pour chaque cellule, et les points correspondants à chaque cellule, ainsi que les pixels de l'image vidéo correspondante sont affichés à l'écran afin que l'opérateur leur affecte une classe. La figure 3.19 montre un exemple de cellule que l'opérateur doit prototyper.

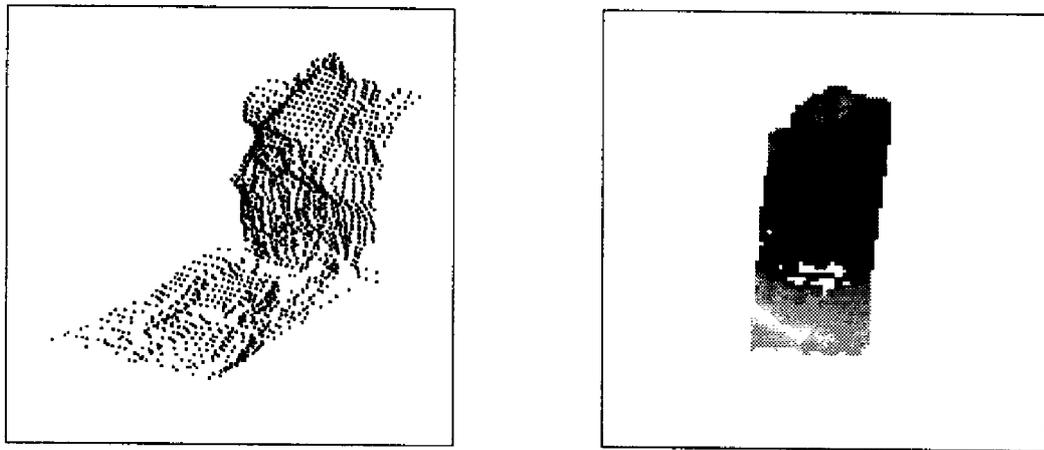


Figure 3.19 : Affichage des points d'une cellule et des pixels correspondants pour prototypage

Cette cellule contient clairement un obstacle (mais aussi une zone plane : il s'agit typiquement d'une cellule qu'on a intérêt à scinder lors de la phase de la classification), et ses attributs, cachés à l'opérateur afin de ne pas l'influencer dans ses choix, ont les valeurs suivantes (d_r , N_z et σ_N n'ont pas de dimension) :

$$A = \begin{pmatrix} d_r \\ \sigma_z \\ \Delta Z \\ N_z \\ \sigma_N \\ D_c \end{pmatrix} = \begin{pmatrix} 3.1383 & - \\ 0.098655 & m \\ 0.34 & m \\ 0.6505 & - \\ 0.3573 & - \\ 4.22 & m \end{pmatrix}$$

Les cinq premières coordonnées du vecteur d'attributs sont élevées : elles sont caractéristiques d'une cellule qui contient un obstacle.

La dizaine d'images que nous avons retenues définit 1250 échantillons de cellules : 880 ont

⁸Nous verrons dans le paragraphe 3.6.4 l'impact de la résolution des images sur les résultats de la classification

été étiquetées planes, 200 accidentées et 150 obstacles. On trouvera en annexe A quelques projections suivant deux attributs de l'espace des prototypes ainsi défini.

Cette phase d'apprentissage est bien entendu primordiale pour les résultats de la classification, elle doit donc être réalisée avec soin. En particulier, il est indispensable que l'opérateur connaisse bien le terrain perçu et le modèle du véhicule qui est destiné à y évoluer afin d'associer les bonnes classes aux cellules.

b) *Estimation des fonctions de densité de probabilité*

Le problème de l'estimation des fonctions de densité de probabilité est simple lorsqu'elles sont modélisables par fonctions analytiques (gaussiennes le plus souvent) : on a affaire à un problème de classification *paramétrique*, mono- ou multi-dimensionnel suivant que l'on dispose d'un ou plusieurs attributs. Les paramètres des lois sont alors déterminés hors ligne, à l'issue de la procédure d'apprentissage.

Mais dans notre cas, seul l'attribut de densité relative suit une loi normale, les lois des autres attributs sont difficiles à paramétrer (figures 3.20 et 3.21) : nous devons alors employer une méthode de classification *non-paramétrique*, qui consiste à estimer la valeur des probabilités conditionnelles *a priori* en ligne, par comparaison de la valeur du vecteur d'attribut A avec ses valeurs pour les prototypes.

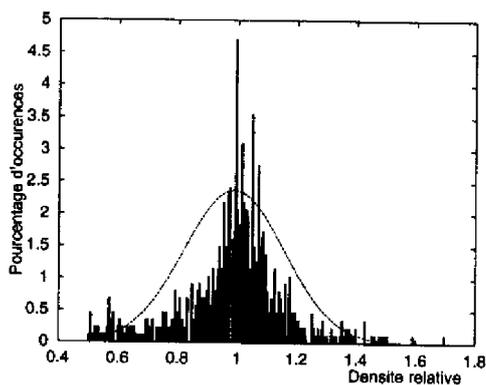


Figure 3.20 : La densité de probabilité de l'attribut "densité relative" suit bien une loi normale pour des cellules de la classe "plan"...

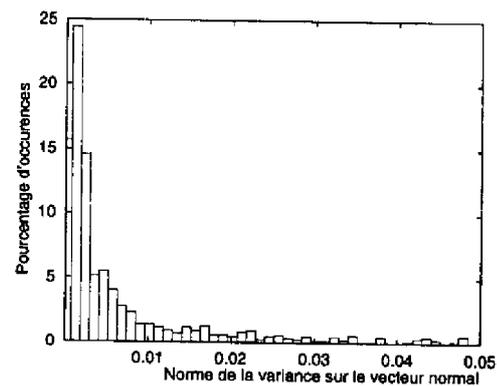


Figure 3.21 : ...mais ce n'est pas le cas pour l'attribut "norme des écarts types sur les coordonnées du vecteur normal"

Si la fonction de densité $P(A)$ ne présente pas de variations importantes à l'intérieur d'un domaine D de l'espace des attributs (hypothèse justifiable dans la plupart des cas et en particulier dans le nôtre), on peut approcher la valeur de $P(A)$ en tout point de D avec la valeur moyenne de $P(A)$ sur D . On a donc :

$$P(A | A \in D) = \frac{\int_D P(A) dA}{\int_D dA} = \frac{P(A \in D)}{V(D)} \quad (3.5)$$

où $V(D)$ est le volume du domaine D , et $P(A \in D)$ est la probabilité qu'une observation A soit située dans le domaine D . Si parmi les M prototypes disponibles, t d'entre eux sont situés dans D , alors le rapport $\frac{t}{M}$ est une bonne approximation de $P(A \in D)$.

L'estimateur 3.5 est utilisable s'il vérifie des propriétés de convergence [Postaire 87], ce qui est le cas pour deux méthodes d'estimation classiques : la méthode du noyau et la méthode des N plus proches voisins.

La méthode du noyau consiste à déterminer la taille du domaine D en fonction du nombre M de prototypes disponibles, en pondérant la contribution de chaque prototype en fonction de sa distance au point mesuré A_0 . Ceci a pour avantage de limiter l'influence des prototypes situés à la frontière du domaine (généralement un hypercube de côté h) : une faible variation autour de A_0 peut en effet faire varier considérablement l'estimation de $P(A_0)$ si les prototypes ne sont pas répartis régulièrement dans l'espace des attributs. L'estimateur est alors de la forme :

$$P(A_0) = \frac{1}{M} \sum_{m=1}^M \frac{1}{V(D_M(A_0))} \cdot \Phi\left(\frac{A_0 - A_m}{h}\right)$$

où Φ est le *noyau de l'estimateur*, et peut avoir une forme triangulaire, gaussienne... La valeur h doit être ajustée de manière à ce que l'estimateur vérifie les conditions de convergence. Elle est délicate à déterminer : une valeur trop grande a pour effet de "lisser" la fonction $P(A)$, et une valeur trop petite donne des variations de $P(A)$ non significatives. De plus, une valeur de h qui convient bien dans les régions de l'espace des prototypes de forte densité sera trop faible pour des régions de faible densité, et *vice versa*.

Cette méthode ne nous a pas paru applicable car elle requiert que l'espace des attributs soit très bien couvert par l'ensemble des prototypes : comme nous raisonnons dans un espace à 6 dimensions, il nous faudrait prototyper plusieurs milliers de cellules de chacune des classes considérées. Nous avons donc préféré appliquer la méthode des plus proches voisins, qui évite de fixer arbitrairement la taille du domaine D , mais l'ajuste en fonction de la densité de l'espace des prototypes au point A_0 considéré. Cette méthode consiste à déterminer le volume occupé par les t_M plus proches voisins situés autour de A_0 . Une valeur de t_M égale à $t_0\sqrt{M}$ garantit la convergence de l'estimateur 3.5.

Cette méthode est mieux adaptée à notre cas, mais peut néanmoins être améliorée : ainsi nous pouvons utiliser une méthode hybride, qui consiste à pondérer l'influence des t_M voisins en fonction de leur distance au vecteur d'attribut A_0 [Forhan 94]. Ceci a l'avantage de favoriser les prototypes proches de l'échantillon mesuré.

3.6.3 Réalisation pratique

a) Estimation des fonctions de densité de probabilité

Chaque cellule prototypée définit un point dans l'espace des prototypes de dimension 6. Comme la méthode d'estimation des fonctions de densité de probabilité par les plus proches voisins requiert une estimation de la distance dans cet espace, l'espace des prototypes est normalisé, et la définition de la distance s'écrit :

$$D(A_1, A_2) = \sum_{i=1}^6 \alpha_i \frac{|A_1^i - A_2^i|}{\beta_i},$$

où β_i est le coefficient de normalisation suivant la direction i , et les α_i sont les coefficients de pondération liés aux attributs A^i . Nous avons pris ces coefficients de pondération égaux à 1, mais il pourrait être intéressant d'étudier le caractère discriminant de chacun des attributs et d'en déduire d'autres valeurs.

Lors de la classification d'une image, il faut déterminer la valeur de la fonction de densité de probabilité de chaque cellule définie par la discrétisation, et donc le volume occupé par les t_M plus proches voisins. Afin de réduire la complexité de cette procédure (qui nécessite la recherche des t_M plus proches voisins parmi M prototypes et le calcul de leur volume), nous avons choisi d'affecter à chaque cellule la valeur de la fonction de densité de probabilité de son plus proche prototype, valeur qui a été préalablement déterminée pendant la phase d'apprentissage. Cette approximation est bien sûr d'autant meilleure que l'espace des prototypes est dense.

Notons que ne sont classifiées que les cellules dont le nombre de points (donc l'attribut d_r) est suffisamment grand : les autres sont étiquetées "non-perçu", et nous verrons qu'elles seront scindées.

b) Décision

Une fois les probabilités d'une cellule $P(A | C_i)$ estimées pour chaque classe considérée, la probabilité d'avoir chacune des classes $P(C_i | A)$ est calculée grâce à la formule de Bayes (la répartition *a priori* est équiprobable : la classification est ici locale, et les informations seront prises en compte lors du processus de fusion présenté dans le chapitre suivant). Il faut alors décider de la classe que l'on doit affecter à la cellule considérée. Une loi de pénalisation symétrique ($r(C_i | C_j)$ vaut -1 si $i = j$, 0 sinon) est peu appropriée dans notre cas, où il est bien plus risqué d'affecter la classe "plan" à une cellule réellement obstacle (non détection) que le contraire (fausse alarme). Nous avons donc empiriquement implémenté une loi de décision non symétrique, afin de limiter les non détections, mais en prenant toutefois soin de ne pas sur-contraindre le problème en générant trop de fausses alarmes. La matrice représentant la loi de pénalisation retenue s'écrit :

$$r(C_i | C_j) = \begin{pmatrix} -0.5 & 0.15 & 0.30 \\ -0.15 & -0.5 & 0.15 \\ -0.30 & -0.15 & -0.5 \end{pmatrix}$$

Notons qu'il est intéressant de multiplier les termes non diagonaux de cette matrice par un terme correctif fonction de D_c [Forhan 94] : les mesures sont plus fiables et les cellules sont plus petites près du capteur (elles ont donc moins de chance de couvrir une zone contenant plusieurs classes) que loin.

c) Contrôle de la scission

Une mesure intéressante de la qualité du résultat de la classification d'une cellule est donnée par l'entropie :

$$E(\text{Cell}_A) = - \sum_{i=1}^n P(C_i | A) \log P(C_i | A)$$

Cette fonction est maximale lorsque toutes les valeurs $P(C_i | A)$ sont égales, et est minimale lorsqu'elles sont toutes nulles, sauf une (qui vaut alors 1). Une grande valeur de l'entropie signifie que la cellule présente des attributs ambigus, ce qui est notamment le cas lorsqu'elle couvre une zone de terrain correspondant à différentes classes.

C'est donc cette mesure du caractère discriminant de l'étiquetage qui est utilisée pour décider si une cellule doit être scindée en 4 sous-cellules : chaque cellule est scindée lorsque son entropie est supérieure à une valeur seuil, déterminée empiriquement (les cellules étiquetées "non-perçu" sont donc systématiquement scindées : leur entropie vaut 1). Les étiquettes des sous-cellules sont alors déterminées en appliquant la formule de Bayes, en considérant que les probabilités *a priori* d'être en présence de chacune des classes sont celles de la cellule initiale.

d) Filtrage des cellules isolées

Il arrive des cas où les cellules sont "isolées", dans le sens où la classe qui leur a été affectée est différente de la classe de toutes leurs voisines. Cela conduit à un morcellement de la zone perçue qui est du à la discrétisation adoptée. Pour pallier cet inconvénient et obtenir une description du terrain plus homogène, les étiquettes de ces cellules sont révisées en considérant l'étiquette de la "sur-cellule" qui les inclut, formée de sa réunion avec ses voisins. L'étiquette de la cellule est déterminée par la formule de Bayes, en considérant que les probabilités *a priori* sont celles de la sur-cellule. Cette procédure de filtrage est en quelque sorte duale de la procédure de scission.

3.6.4 Résultats

a) Importance du filtrage et de la scission

La figure 3.22 (page 49) montre les résultats de la procédure de classification d'une image de résolution 256^2 avant l'application des procédures de filtrage et de scission. La classe "plan" est représentée en vert, la classe "accidenté" en orange, et la classe "obstacle" en rouge⁹. La zone perçue est essentiellement plane, mais on y distingue clairement la présence de deux rochers obstacles (au centre et à gauche de l'image), et le fond de l'image correspond à une zone plutôt accidentée. Ces caractéristiques sont assez bien exhibées par la classification, mais la discrétisation rend la description assez grossière : la figure 3.23 montre l'amélioration de la description de la scène après l'application des procédures de filtrage et de scission, exécutée sur les cellules dont l'entropie est grande.

Deux autres résultats de la procédure de classification sur d'autres scènes (perception d'une faille et image acquise avec un fort roulis) sont présentés figure 3.24.

⁹Ces conventions de couleur seront respectées tout au long du mémoire



Figure 3.22 : Résultat de la première étape de classification : pixels corrélés, classification de la zone perçue et reprojection dans le repère capteur

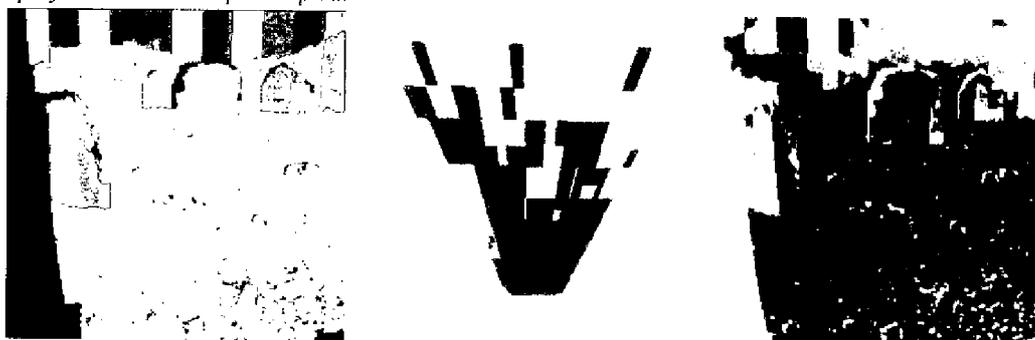


Figure 3.23 : Amélioration après filtrage et scission des cellules ayant une grande entropie (en gris foncé sur la figure de gauche)

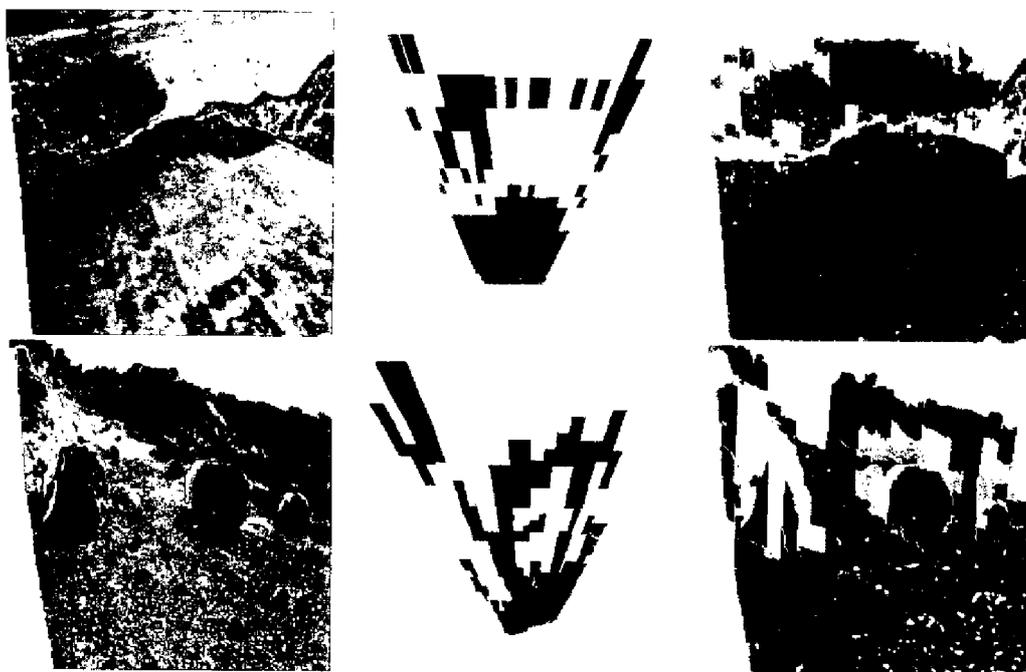


Figure 3.24 : Résultat de la classification de deux autres images

b) Influence de la résolution image

La figure 3.25 montre les résultats de la classification de l'image de la figure 3.22 à trois résolutions différentes : les résultats sont assez semblables, à la finesse de la description près bien entendu. C'est un aspect fondamental de notre méthode, qui est très robuste même pour des images de faible résolution : elle permet de se contenter d'images corrélées à la résolution 128^2 , qui sont calculées 16 fois plus rapidement que des images de résolution 512^2 .



Figure 3.25 : Classification d'une image à trois résolutions différentes (de gauche à droite, résolution 512^2 , 256^2 et 128^2 - les classes sont représentées en niveaux de gris : noir = obstacle, gris foncé = accidenté et gris clair = plan)

c) Performances

La performance en temps de la procédure de classification est bonne, si l'on tient compte du fait que la méthode est peu sensible à la résolution image. En effet, la complexité des calculs est une fonction linéaire du nombre de points traités. Sur une station Sparc 10, avec un code qui a été développé sans optimisation particulière, la procédure prend généralement un temps inférieur 0.5 milli-seconde par point : ainsi les images précédentes de résolution 256^2 sont classifiées en moins de 2 secondes, et les images à la résolution 128^2 en moins de 0.5 secondes. Il existe bien entendu de nombreux moyens de réduire ces temps de calculs (en stockant l'espace de prototypes dans une table de hachage par exemple), et la procédure est parallélisable.

3.7 Classification d'images laser

La grande différence entre les images issues du télémètre 2-axes et le capteur stéréo est la résolution : la meilleure résolution du télémètre (0.5°) reste 5 fois inférieure à la résolution obtenue en corrélant des images sous-échantillonnées d'un facteur 2 (résolution 256^2), et pour des raisons de temps d'acquisition, nous avons réalisé nos expérimentations de navigation en utilisant le télémètre avec une résolution d'un degré. Par conséquent, des cellules qui contiendraient un grand nombre de points pour une image laser auraient une résolution très grossière et rendraient la description de la zone perçue très imprécise. Pour avoir des cellules de taille raisonnable, nous avons choisi des taux τ_θ et τ_ϕ égaux à 6. Malheureusement, les prototypes classés pour ces taux se distinguent très mal les uns des autres : le faible nombre

de points des cellules affecte considérablement les valeurs statistiques des attributs¹⁰.

Nous avons donc implémenté une procédure de classification bien plus simple, basée sur des seuils sur chacun des attributs, mais qui s'est aussi avérée très robuste.

3.7.1 Algorithme de classification

```

ClasseCellule( Cell,  $d_r$ ,  $\sigma_z$ ,  $\Delta_z$  )
  Classe  $\leftarrow$  PremierTest(  $d_r$ ,  $\sigma_z$ ,  $\Delta_z$  )
  Si ( Classe  $\neq$  Plat )
    CalculeAttributsNormale( Cell )
    Classe  $\leftarrow$  SecondTest(  $d_r$ ,  $\sigma_z$ ,  $\Delta_z$ ,  $N_z$ ,  $\sigma_N$  )
    Si ( ( Classe  $\neq$  Plat ) Et ( Cell  $\neq$  Scindée ) )
      ScindeCellule( Cell )
      Pour (chaque sous-cellule) Faire :
        ClasseCellule( SousCell )
      Fin Pour
    FinSi
  FinSi
Fin

```

Figure 3.26 : L'algorithme de classification d'une cellule de points laser

Le principe de classification est simple (figure 3.26) : en cours de remplissage de chaque cellule, les attributs d_r , σ_z et Δ_z sont déterminés. Une fois toutes les cellules remplies, un premier test sur les valeurs de ces attributs permet de déterminer celles qui correspondent à une partie plane du sol (après avoir exclu et scindé les cellules qui contiennent trop peu de points). Les attributs concernant le vecteur normal moyen (N_z et σ_N) sont déterminés pour les cellules restantes, et un second test permet de leur affecter une des trois classes possibles. À l'issue de cette procédure, les cellules non-planes sont scindées, et la classification est réitérée pour chacune des quatre sous-cellules.

La procédure et les seuils nécessaires aux deux tests de classification sur les attributs d'une cellule ont été déterminés empiriquement, en observant des images prototypes, et en considérant les contraintes de locomotion du robot (hauteur maximale des marches franchissables, pentes maximales...). Le second test isole les cellules planes des cellules obstacles, et affecte la classe "accidentée" aux cellules restantes (figure 3.27).

3.7.2 Modèle d'erreur

Contrairement à la technique de classification statistique utilisée pour les images stéréoscopiques, notre classification heuristique des images laser ne donne pas de probabilité d'appartenance des cellules à chacune des classes, ni de probabilité d'erreur sur l'étiquetage. Mais il apparaît clairement que si des cellules étiquetées "accidenté" peuvent correspondre à des zones réellement planes, et que certaines cellules "obstacle" peuvent correspondre à des zones réellement accidentées ou même planes, l'inverse n'est que rarement vrai : il faudrait

¹⁰D'autant plus qu'il s'est avéré délicat pour un opérateur de classer des cellules peu denses, et dont on ne peut visualiser que les points 3D

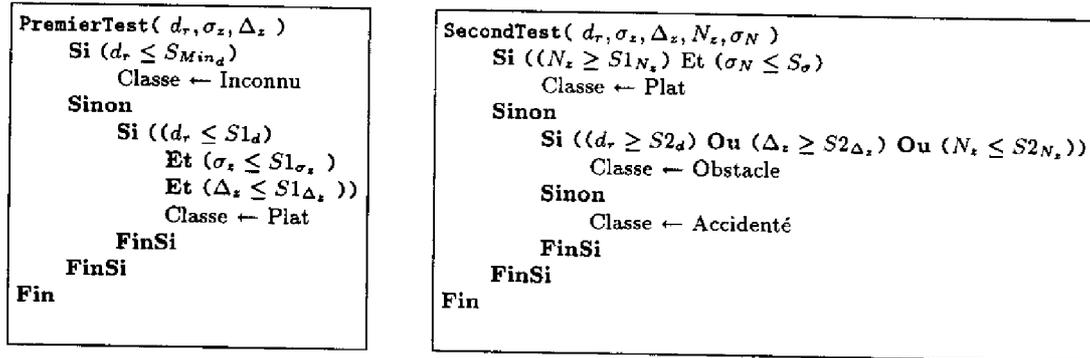


Figure 3.27 : Les tests sur les attributs pour classer les cellules

en effet que les points acquis sur une zone plane soient totalement erronés pour qu'ils génèrent une cellule dont les attributs correspondent à ceux d'une cellule obstacle par exemple. La détermination des seuils de classification a été réalisée de façon à ce que la méthode ne produise pas de non-détections, mais plutôt des fausses alarmes.

Par ailleurs, la probabilité que l'étiquetage soit faux grandit avec la distance des cellules au capteur : les mesures sont plus bruitées (et aussi plus souvent absentes, à cause du faible angle d'incidence), et surtout la surface des cellules augmente avec cette distance, et le terrain qu'elle couvre risque donc d'être peu homogène.

Nous avons donc défini empiriquement une *confiance* c sur l'étiquetage qui est une fonction linéaire de D_c , dont la pente dépend de l'étiquette :

$$c(C_i) = 1 - \alpha_i \frac{D_c}{D_{max}}, \text{ avec } \alpha_i = \begin{cases} 1.00 \text{ si Obstacle} \\ 0.75 \text{ si Accidenté} \\ 0.50 \text{ si Plat} \end{cases} \text{ et } D_{max} = 15.0 \text{ m}$$

3.7.3 Résultats

La figure 3.28 montre une image d'une scène perçue par le télémètre et le résultat de la classification des données acquises : On distingue bien les rochers présents, qui sont tous étiquetés "obstacle".

Le fait que la procédure donne d'excellents résultats sur des images acquises avec une faible résolution nous permet d'utiliser le télémètre avec un taux de balayage de 1° suivant les deux axes : cela permet de réduire par 4 le temps d'acquisition d'une image, aspect non négligeable lorsque l'on réalise des expérimentations. La performance des algorithmes de classification est alors excellente : ainsi l'image de la figure 3.28 a été classifiée en moins de 0.15 seconde.

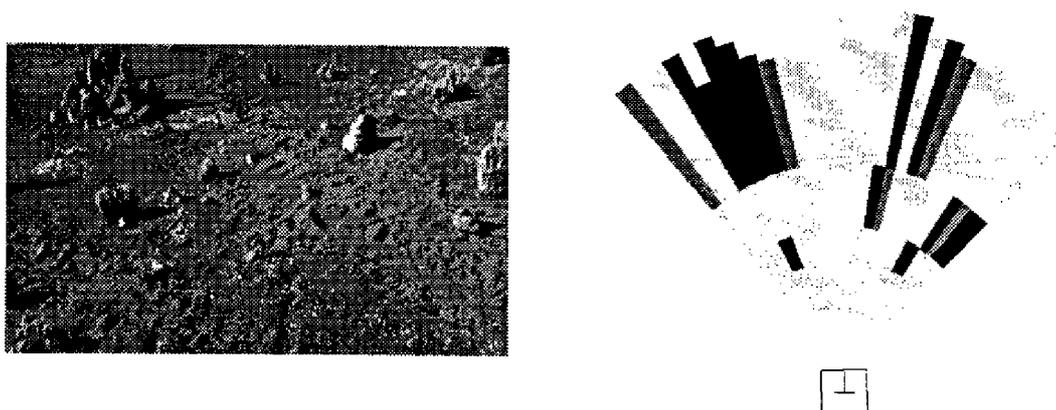


Figure 3.28 : Classification d'une image laser : image vidéo de la scène perçue et résultat

3.8 Conclusions - Extensions possibles

La méthode a été utilisée sur l'ensemble des couples d'images stéréoscopiques acquis sur le terrain Geroms dont nous disposons, et sur plusieurs centaines d'images laser acquises avec le télémètre d'Adam lors de démonstrations de navigation. Elle donne de bons résultats, satisfait l'ensemble des spécifications que nous en attendions (mis à part la détection de la nature du sol) et présente les avantages suivants :

- Comme elle raisonne sur des attributs calculés pour un grand nombre de points, elle est très peu sensible aux erreurs et incertitudes sur les mesures : elle permet notamment de s'affranchir d'une procédure de calibration géométrique et statistique précise du capteur utilisé. Les seuls paramètres qui influent considérablement sur son résultat sont les angles qui donnent l'attitude du robot (tangage et roulis) : ces paramètres sont généralement connus avec précision grâce aux capteurs proprioceptifs du véhicule ;
- Elle classe la zone perçue en trois classes de terrain (plus la classe "inconnu"), et en ce sens fait plus que beaucoup de méthodes de détection d'obstacles ;
- Enfin, elle associe à chaque cellule une incertitude sur son étiquetage : cet aspect est fondamental, et nous verrons qu'il s'agit d'un point clé nécessaire à la construction d'un modèle global, et à la planification de chemins et de tâches de perception.

Elle pourrait néanmoins être améliorée et étendue :

- *Amélioration de la finesse de la discrétisation* : la taille des cellules, et donc la finesse de la description de la zone perçue, dépend de la résolution des images. On aurait tout intérêt à utiliser la possibilité de corrélérer des images stéréoscopiques de différentes résolutions : la zone perçue serait d'abord classifiée sur la base d'images corrélérées à la résolution la plus grossière (64^2 par exemple), et les zones des images correspondant

aux cellules à scinder pourraient alors être corrélées à une résolution plus fine, le processus se répétant jusqu'à la résolution maximale (512^2). Cela permettrait d'obtenir une description très fine de la zone perçue, tout en évitant de corréler l'ensemble des pixels à la résolution la plus fine :

- *Perception en mouvement* : une technique similaire pourrait être implémentée pour un capteur qui perçoit des lignes de points lorsque le robot se déplace (avec une nappe laser par exemple), afin de détecter d'éventuels obstacles (cas du mode réflexe [Dunlay 88, Singh 91, Hebert 94]). Seule une bande de cellules qui se déplace avec le robot serait alors définie, et leur remplissage, l'estimation de leurs attributs et leur classification pourraient être déterminés en ligne. Par rapport à une méthode qui consisterait à analyser les lignes une à une, cela permettrait de conserver un historique des points acquis précédemment, et donc de limiter les fausses alarmes. L'aspect qualitatif de la méthode permettrait aussi de s'abstraire des problèmes soulevés par les vibrations du capteur ;
- *Définition de nouveaux attributs* : L'ensemble de attributs que nous avons définis n'est pas exhaustif, et l'introduction d'un attribut de courbure (qui est délicat à estimer, car très dépendant du bruit) ou d'altitude moyenne des cellules par exemple permettrait d'améliorer les résultats de la méthode ;
- *Coopération avec des images de luminance* : Dans le cas de la stéréovision, on sait associer à une cellule les pixels correspondant aux points 3D qui y appartiennent (c'est ce qui nous a permis de représenter les résultats de la classification dans le repère capteur). Une technique de classification tout à fait similaire pourrait être développée pour ces cellules de pixels, en se basant sur des attributs *moyens*, tels que la couleur, les variations de couleur, la texture... Cela permettrait notamment d'affecter une estimation de la nature du sol aux cellules, mais aussi d'affiner l'estimation de leur classe géométrique, en utilisant des lois de corrélation (qui existent réellement) qui lient la nature du sol et sa classe géométrique. Une telle coopération pourrait aussi être envisagée avec les valeurs de réflectance retournées par certains télémètres laser ;
- Enfin, une analyse des corrélations entre les attributs d'une cellule et l'intérêt de la zone de terrain qu'elle couvre pour la fonction de localisation extéroceptive permettrait de faire de la détection d'amers.

Chapitre 4

Construction d'un modèle topologique global

La procédure de classification des images de points 3D fournit une description de la zone perçue localement en termes de régions (les cellules définies par la discrétisation), pour lesquelles sont connues diverses caractéristiques. Afin de disposer d'un modèle de navigation global, il est nécessaire d'intégrer l'ensemble des images qui sont classifiées au fur et à mesure de l'évolution du robot : c'est ce à quoi nous nous intéressons dans ce chapitre. Nous y présentons la structure de données que nous avons choisie pour représenter ce modèle global, puis les techniques de fusion qui permettent de le mettre à jour. Nous décrivons ensuite la procédure de structuration qui permet de construire un graphe de régions, ainsi que la façon dont est géré le modèle de manière à limiter la mémoire qui lui est nécessaire. Le chapitre s'achève par la présentation d'un cadre structurel dans lequel figure l'ensemble des représentations de l'environnement nécessaires au robot pour mener à bien une tâche de navigation.

4.1 Fusion de perceptions locales

4.1.1 Quelle structure de données ?

Il est important de bien choisir la structure de données pour représenter les régions : elle va définir les informations dont on disposera sur la forme géométrique des régions, ainsi que les algorithmes (fusion, regroupements, grossissements...) que l'on peut réaliser sur ces régions.

Parce que la procédure de classification fournit une description polygonale de la zone perçue¹, nous avons tout d'abord envisagé de conserver une description polygonale des régions : une telle description a le grand avantage d'être compacte, et de décrire précisément la forme des régions. On peut notamment y effectuer des calculs de visibilité précis, y appliquer des procé-

¹les arcs de cercles qui définissent les cellules pour les images laser sont approximés par des segments de droite

dures de reconnaissance de lieux, ou en déduire un espace de configuration pour planifier les déplacements d'un véhicule polygonal [Pignon 91]. Il existe de très nombreuses contributions dans lesquelles un environnement intérieur est ainsi représenté [Moutarlier 91, Leonard 90] : il est vrai que ces environnements présentent de nombreuses arêtes facilement modélisables par des segments de droite, et sont donc naturellement adaptés à une représentation polygonale, ce qui n'est pas le cas des environnements naturels.

Mais le principal défaut des représentations polygonales de l'environnement est leur difficulté de manipulation : on se heurte très vite à des problèmes insolubles et générateurs de grosses erreurs si les divers algorithmes d'intersection ou de regroupement de polygones ne sont pas réalisés en précision infinie (calcul rationnel), ce qui alourdit considérablement les temps de calcul nécessaires [Moutarlier 91].

Nous avons préféré opter pour des régions construites sur la base d'une représentation discrète de l'environnement (grille ou *bitmap*). Ce type de représentation très courant en robotique a été utilisé aussi bien pour des environnements intérieurs [Moravec 85, Elfes 89] qu'extérieurs [Kweon 92], pour lesquels le manque de structure rend délicat la construction de toute représentation algébrique. Les bitmaps ont le grand avantage d'être aisés à gérer, et il existe de nombreux algorithmes simples, rapides et robustes qui permettent d'en extraire des régions, des frontières entre régions, ou d'effectuer des calculs de visibilité par lancer de rayon. De plus, le MNT produit par le processus d'interpolation des données [Nashashibi 93] est lui-même définie sur un bitmap : si nécessaire, il sera donc aisé d'établir une correspondance entre les deux modèles.

L'inconvénient des bitmaps par rapport à une description polygonale est qu'ils occupent un grand volume mémoire si leur résolution est fine, et ce d'autant plus que le nombre d'informations codées dans un pixel est grand. Une résolution grossière permet de pallier cet inconvénient, mais produit une description très approximative : nous avons préféré utiliser une résolution égale à celle du MNT (de l'ordre de quelques centimètres, une dizaine au maximum), et nous verrons dans le paragraphe 4.3 comment nous optimisons la taille mémoire nécessaire, en gérant un bitmap construit localement autour du robot, et une carte de régions qui est mémorisée sur l'ensemble des zones perçues.

4.1.2 D'une image classifiée à un bitmap

Une image classifiée est une liste de polygones : sa représentation sous la forme d'un bitmap est donc simplement déterminée grâce à un algorithme dit de "coloriage". Le bitmap est défini dans le repère global R_m lié au monde, et chaque quadrilatère qui définit une cellule est exprimé dans le repère R_p , propre à l'image de pavés. Avant la procédure de coloriage, les coordonnées des cellules doivent donc être exprimées dans le repère R_m , grâce à la matrice de transformation $T_{R_p \rightarrow R_m}$, qui est définie ainsi :

$$T_{R_p \rightarrow R_m} = T_{R_r \rightarrow R_m} \circ T_{R_c \rightarrow R_r} \circ T_{R_p \rightarrow R_c},$$

où $T_{R_r \rightarrow R_m}$ est la position du robot dans le repère du monde (donnée par le module de localisation) et $T_{R_c \rightarrow R_r}$ est la matrice de calibration extrinsèque du capteur. L'axe vertical

du repère R_p correspondant à l'axe vertical du repère R_m lié au monde, cette transformation se réduit à une rotation dans le plan horizontal ($O_m X_m Y_m$), suivie d'une translation suivant les trois dimensions.

4.1.3 Quelles informations garder ?

Dans chaque pixel du bitmap vont être codées les diverses informations utiles à la navigation données par la procédure de classification. Afin que le volume mémoire nécessaire à un pixel ne soit pas trop grand, chacune de ces informations est codée sur un certain nombre de bits, qui dépend de son domaine de variation et de la précision avec laquelle on désire la mémoriser. Nous avons retenu les informations suivantes :

- **Classe de terrain (ou étiquette)** : donnée par la loi de décision dans le cas des images stéréo et grâce à une classification heuristique dans le cas des images laser, l'étiquette d'une cellule est bien entendu une information fondamentale qu'il faut garder. N'ayant considéré que 4 classes de terrain (en comptant la classe "non-perçue"), 2 bits suffisent à représenter cette information ;
- **Probabilité d'erreur sur l'étiquetage** : il est important de connaître la confiance avec laquelle l'étiquetage a été réalisé : cette confiance va notamment être nécessaire pour mettre à jour l'étiquetage d'une zone perçue plusieurs fois (paragraphe 4.1.4), mais aussi pour structurer le bitmap (paragraphe 4.2), évaluer la faisabilité de certains déplacements et planifier des tâches de perception. Dans le cas des images stéréo classifiées, nous conservons aussi pour chaque pixel les probabilités qu'il appartienne à chacune des classes possibles, ce qui permettra de faire de la fusion bayésienne d'étiquettes. Chacune de ces probabilités est codée sur 7 bits, ce qui permet de les mémoriser avec une précision de l'ordre de 1% ;
- **Altitude moyenne** : bien qu'indicative, l'altitude moyenne des cellules est une information intéressante : elle permettra d'effectuer des calculs de visibilité, et aussi d'évaluer les variations d'altitude le long d'un chemin. L'altitude est codée sur 12 bits, ce qui permet d'avoir une précision de 1 centimètre et de pouvoir représenter des variations de l'ordre d'une quarantaine de mètres (une valeur de décalage définie sur l'ensemble du bitmap permet de coder des altitudes négatives) ;
- **Variance sur l'altitude** : elle exprime la confiance que l'on a sur l'altitude, et sera notamment nécessaire à la mise à jour de l'altitude d'une zone perçue plusieurs fois. Elle a un domaine de variation compris entre 0 et 0.2 cm^2 , prend souvent des valeurs inférieures à 10^{-4} cm^2 (voir annexe A), et est donc codée sur 12 bits ;

Bien que le vecteur normal moyen soit une bonne indication de la pente du terrain, nous avons choisi de ne pas le mémoriser dans les pixels : cela nécessiterait de nombreux bits supplémentaires (d'autant plus qu'il faudrait aussi coder les variances sur ses coordonnées pour effectuer sa mise à jour), et l'information d'altitude moyenne nous a semblé largement suffisante pour une estimation qualitative des variations d'altitude le long d'un chemin.

Signalons que quelques bits sont réservés afin de coder la date à laquelle chaque pixel a été dernièrement perçu : il s'agit d'une information qui peut être intéressante, elle est notamment

nécessaire à la gestion du bitmap. Enfin, le processus de structuration du bitmap nécessite que d'autres informations soient codées dans chaque pixel (numéros de régions, étiquettes après grossissements...); elles seront définies dans le paragraphe correspondant (4.2).

4.1.4 Mise à jour du bitmap : fusion d'images classifiées

A l'initialisation de la procédure de modélisation, la mise à jour du bitmap après la première perception ne pose aucun problème : il suffit de colorier les pixels du bitmap correspondant à chacune des cellules de l'image classifiée, en y codant l'ensemble des informations retenues. Mais après chaque nouvelle perception, il est nécessaire de combiner les informations extraites par la procédure de classification avec celles qui sont déjà présentes dans le modèle global. Les propriétés des capteurs et de la procédure de classification font qu'il apparaît de nombreux conflits entre ces deux sources d'information, et il faut donc être capable de les résoudre.

a) *Mise à jour de l'étiquette : cas des images stéréo*

Avant la nouvelle perception k , les probabilités d'étiquetage $\{P_{k-1}(q)\}$ affectées à un pixel sont les probabilités d'appartenance à une classe q . La $k^{\text{ème}}$ perception fournit le vecteur d'attributs A_k pour chaque cellule, et la formule de Bayes nous donne alors pour toutes les classes q :

$$P_k(q) = \frac{P(A_k | q)P(q | A_{k-1})}{P(A_k)} = \frac{P(A_k | q)P_{k-1}(q)}{\sum_q P(A_k | q)P_{k-1}(q)}, \quad (4.1)$$

où les $P_{k-1}(q)$ sont les probabilités *a priori* d'être en présence de la classe q , c'est-à-dire les probabilités figurant dans le modèle global avant la fusion, et les $P_k(q)$ les probabilités après la fusion. L'application de cette formule à chaque perception revient en fait à écrire que $P_k(q) = P(q | A_k, A_{k-1}, \dots, A_0)$.

Dans notre cas, les probabilités d'être en présence de la classe q sont calculées lors de la classification en considérant une répartition *équiprobable* des probabilités *a priori*, ce qui nous permet d'écrire la formule 4.1 de cette façon :

$$P_k(q) = \frac{P_{k_{local}}(q)P_{k-1}(q)}{\sum_q P_{k_{local}}(q)P_{k-1}(q)},$$

où les $P_{k_{local}}(q)$ sont les probabilités partielles données par la classification.

Une fois que les probabilités partielles d'un pixel sont modifiées, l'application de la loi de décision présentée dans le chapitre 3 permet de leur affecter une classe, ainsi qu'une probabilité d'erreur sur cette classe.

Cette séquence d'images présente le résultat de la fusion de deux images stéréo classifiées, acquises avant et après un déplacement d'environ deux mètres dans l'axe de la première



Figure 4.1 : Deux acquisitions de paires d'images stéréoscopiques : pixels pour lesquels une valeur de disparité a effectivement été trouvée par l'algorithme de stéréovision.

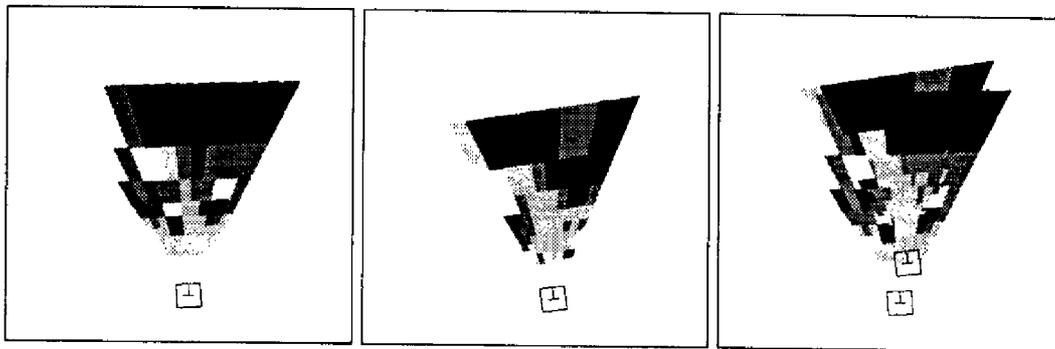


Figure 4.2 : Étiquettes après application de la loi de décision : 1ère et 2ème vues, et fusion des deux (le carré représente la position du ban stéréoscopique, il a un mètre de côté)

image, suivi d'une légère rotation sur la gauche : la figure 4.1 montre les pixels pour lesquels une valeur de disparité a effectivement été trouvée par l'algorithme de stéréovision.

b) Mise à jour de l'étiquette : cas des images laser

La procédure de classification que nous avons développée pour les images laser donne des informations sur l'étiquetage des cellules bien moins riches que pour les images stéréo : rappelons qu'elle ne fournit notamment pas de probabilité qu'une cellule appartienne à chacune de classes possibles, mais *une* classe, à laquelle est associée une confiance directement liée à la distance de la cellule au capteur.

Nous ne pouvons pas dans ce cas faire de fusion bayésienne, mais la confiance permet néanmoins de traiter les cas de conflit qui peuvent survenir : chaque pixel du bitmap perçu une nouvelle fois se voit attribuer l'étiquette de la cellule qui a la confiance la plus grande. Cette procédure simple s'est avérée robuste sur l'ensemble des expérimentations que nous avons menées (figure 4.9 page 67) (notons qu'elle est aussi applicable pour les images stéréoscopiques classifiées).

c) *Mise à jour de l'altitude*

Si l'on suppose que la répartition de l'altitude des points d'une cellule suit une loi gaussienne, alors le principe de maximum de vraisemblance permet de déterminer une nouvelle estimée de l'altitude et de la variance d'un pixel grâce aux deux formules suivantes :

$$Z_k = \frac{\sigma_{k_{local}}^2 Z_{k-1} + \sigma_{k-1}^2 Z_{k_{local}}}{\sigma_{k_{local}}^2 + \sigma_{k-1}^2}; \quad \sigma_k = \frac{\sigma_{k_{local}} \sigma_{k-1}}{\sqrt{\sigma_{k-1}^2 + \sigma_{k_{local}}^2}}$$

où (Z_{k-1}, σ_{k-1}) sont les informations initialement présentes dans le bitmap, $(Z_{k_{local}}, \sigma_{k_{local}})$ sont les informations données par la classification de la perception k , et (Z_k, σ_k) sont les informations contenues dans le modèle global après fusion de la perception k . Ces formules sont couramment utilisées [Nashashibi 93, Hebert 89b, Kweon 92], et présentent le grand avantage d'intégrer les informations de toutes les perceptions successives d'un même pixel.

La figure 4.3 présente la mise à jour de l'altitude et de la variance sur l'altitude des images stéréoscopiques classifiées de la figure 4.1, et la figure 4.4 présente cette mise à jour pour les images laser de la figure 4.9.

Remarquons qu'il n'est dans notre cas pas possible d'exécuter une procédure de corrélation entre les pixels correspondant à la nouvelle perception et ceux du bitmap, qui permettrait de recalculer éventuellement la position du robot [Moravec 85] : l'altitude des pixels étant uniforme pour tous les pixels d'une cellule, la correction de la position donnée par un score de corrélation maximum n'aurait pas beaucoup de sens. Notre procédure est une procédure "d'aggrégation", et non une procédure de "recalage-fusion". Elle requiert donc que la localisation (cap et position) du robot soit connue *précisément*, ce qui est le rôle d'un autre module (voir paragraphe 4.4).

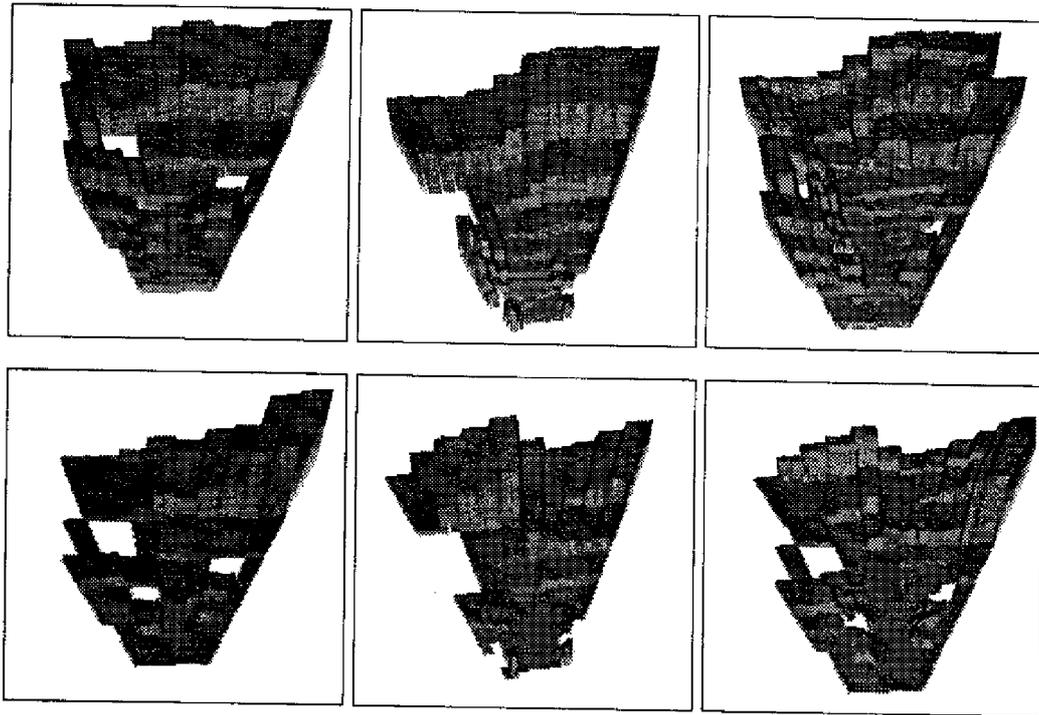


Figure 4.3 : Altitudes (en haut) et variances sur l'altitude (en bas) des images stéréoscopiques classifiées de la figure 4.1 : 1ère et 2ème vues, et fusion des deux

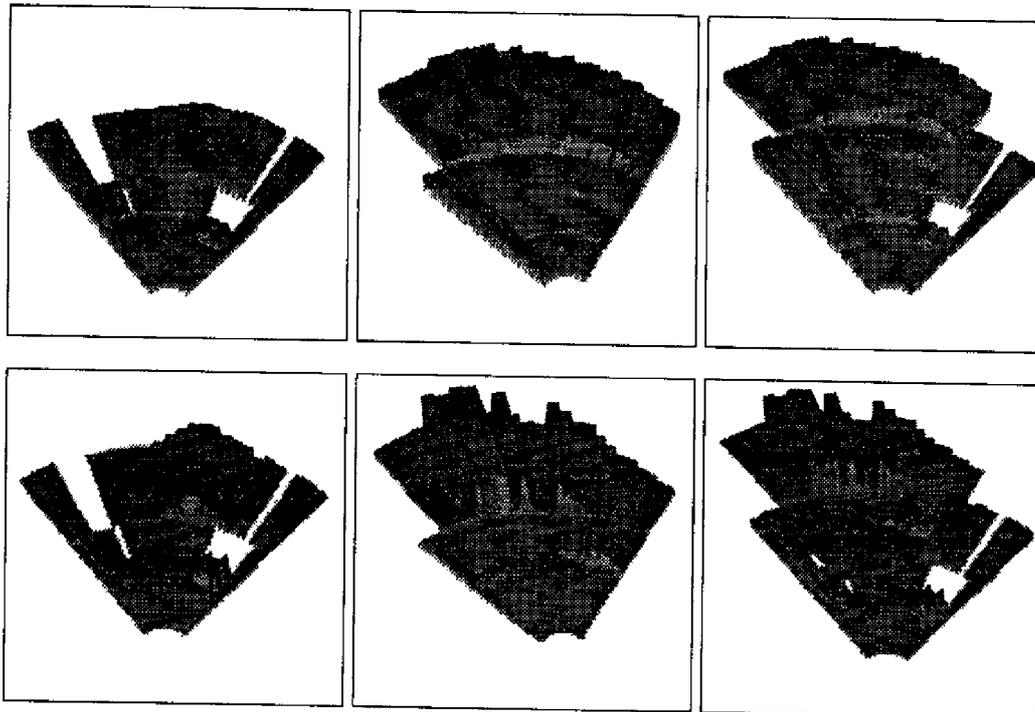


Figure 4.4 : Altitudes (en haut) et variances sur l'altitude (en bas) de deux images laser classifiées : 1ère et 2ème vues, et fusion des deux

4.2 Structuration : construction d'un graphe de régions

Pour pouvoir raisonner rapidement sur le bitmap, il est nécessaire de le "structurer" en une représentation qui permette d'évaluer rapidement des chemins : nous présentons ici le processus d'abstraction qui conduit à la définition d'un graphe de régions à partir du bitmap.

4.2.1 Un modèle de robot ponctuel

Pour évaluer des chemins et les modes de déplacement applicables, il est nécessaire de connaître la classe du terrain sur laquelle est situé le robot lorsqu'il est en une position donnée : ainsi, nous devons considérer que le robot intersecte un obstacle dès qu'une de ses parties est située sur une zone obstacle, et de même il est dans une zone accidentée ou inconnue dès qu'une de ces parties l'est. En considérant un modèle de robot circulaire, on détermine ces cas d'intersection très rapidement grâce à une procédure de grossissement des zones dites "contraintes" (obstacles, accidentées et inconnues). Ces grossissements définissent dans le bitmap des "étiquettes grossies", et nous permet alors de raisonner sur un robot ponctuel.

Les grossissements sont effectués selon les règles de priorité suivantes, définies en considérant que le robot est situé dans la région qui l'intersecte la plus contrainte :

- Le grossissement des régions obstacles est prioritaire sur celui des zones inconnues et accidentées ;
- Le grossissement des régions accidentées est prioritaire sur celui des zones inconnues ;
- Et enfin les régions contraintes étiquetées avec une certaine confiance sont prioritaires sur les régions de même classe, mais de confiance inférieure.

Nous avons implémenté un algorithme de grossissement de régions classique, qui consiste à propager les étiquettes grossies dans le bitmap à partir des pixels frontières d'une région (figure 4.5). Notons que la définition du voisinage d'un pixel (4-connexité ou 8-connexité) influe sur le résultat du grossissement d'une région : les deux types de voisinages définissent un grossissement qui n'est pas isotropique. L'utilisation alternée de ces deux types de voisinage permet de se rapprocher d'un grossissement isotropique (figure 4.6).

La figure 4.10 (page 67) présente le résultat de l'étape de grossissement d'un modèle de terrain. Notons que les étiquettes et confiances après grossissement doivent être mémorisées de façon distincte des étiquettes et confiances perçues : il est en effet nécessaire de conserver ces dernières pour effectuer la fusion des nouvelles perceptions. Deux champs supplémentaires sont donc nécessaires dans le codage des pixels.

4.2.2 Regroupement en régions

Une fois que la procédure de grossissement est effectuée, l'étiquette grossie d'un pixel indique la classe de terrain sur laquelle se trouverait le robot à la position correspondant à ce pixel.

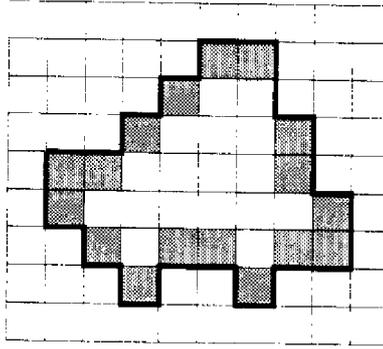


Figure 4.5 : Les pixels contours d'une région à grossir

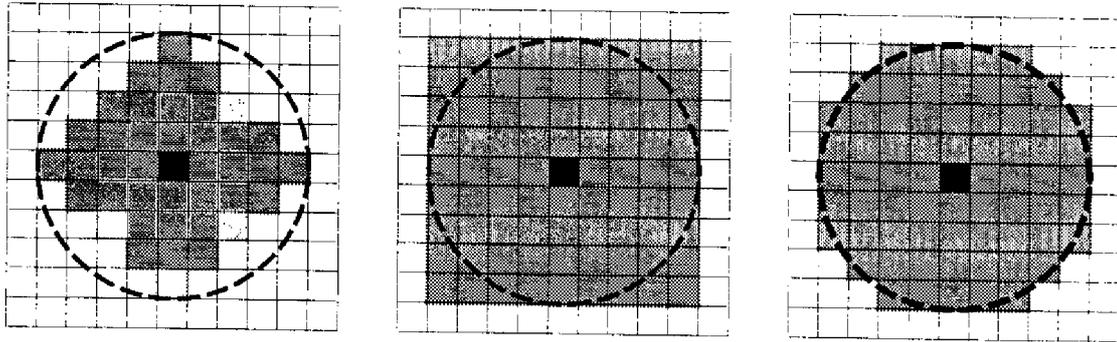


Figure 4.6 : Résultat du grossissement avec propagation par 4-voisins, 8-voisins, et alternée

Il n'est pas question de considérer que ces pixels définissent un graphe, le grand nombre de nœuds ainsi définis rendant extrêmement coûteuse toute recherche de chemin : une procédure de regroupement en régions permet de représenter le bitmap sous une forme à la fois plus compacte et plus abstraite.

Cette phase ne présente aucune difficulté (nous avons choisi l'algorithme récursif de regroupement), mais il est nécessaire de définir le critère d'homogénéité des pixels qui permet de les regrouper. Un critère basé sur l'égalité de *tous* les champs des pixels présente le grand avantage de produire une carte de régions qui contient *toutes* les informations présentes dans le bitmap (le processus de structuration est alors inversible). Mais les régions qu'il détermine sont souvent petites, notamment à cause des procédures de fusion et de grossissement qui génèrent du "bruit" sur le bitmap (figure 4.7). Avec un tel critère, le bitmap de la figure 4.10, qui couvre une surface totale de 1300 m², est segmenté en près de 3000 régions, dont la surface moyenne est de 0.45 m². Nous reviendrons sur cet aspect dans le paragraphe suivant.

4.2.3 Détermination du graphe des régions

Après que les régions aient été déterminées, un algorithme de suivi de contours inter-pixels permet d'extraire les relations de connexité qui existent entre elles. Ces relations déterminent un graphe, dont les nœuds et les arcs sont définis comme indiqué figure 4.8. Seuls les nœuds

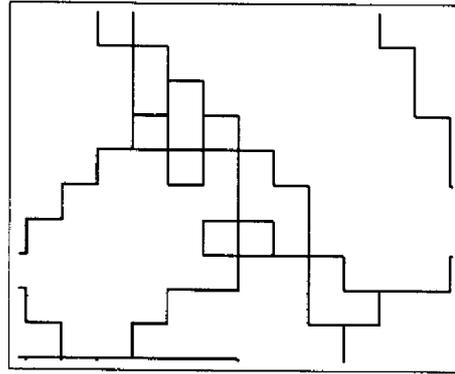


Figure 4.7 : Avec un regroupement de régions basé sur un critère d'homogénéité de tous les champs des pixels, certaines régions sont réduites à un pixel

du graphe sont déterminés par cet algorithme, les arcs ne seront développés qu'en cours de recherche. Remarquons qu'avec cette définition des nœuds et des arcs du graphe, les variations d'altitude d'un chemin dans le graphe sont déterminées par le franchissement d'un nœud, et non d'un arc : les régions ont en effet une altitude uniforme.

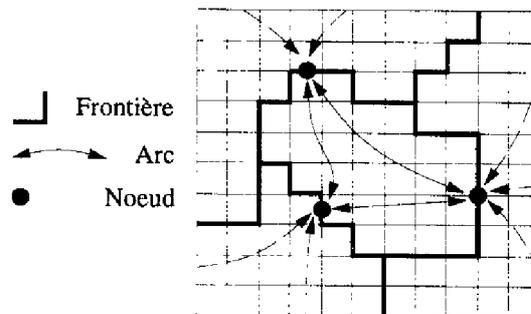


Figure 4.8 : Définition des nœuds et des arcs du graphe

Cette figure illustre l'inconvénient de la définition des contours des régions que nous avons adoptée, par rapport à une définition polygonale, qui permet la détermination d'un graphe de visibilité : la distance cartésienne entre deux nœuds peut être souvent inférieure à la distance du chemin qu'il faut effectivement suivre pour rallier les deux nœuds sans changer de région. Cela est d'autant plus gênant que les régions sont grandes. Par ailleurs, les nœuds définissent en quelque sorte une discrétisation (ou maillage) de l'environnement, et l'on a intérêt à ce qu'elle soit la plus fine possible si l'on désire avoir une bonne évaluation des distances : cela milite pour une définition des régions basée sur un critère d'uniformité de toutes les valeurs définies pour chaque pixel.

Mais nous avons vu qu'une telle définition produit de nombreuses régions de la taille d'un pixel, qui n'ont pas beaucoup de signification pour faire une recherche de chemin qualitative, et leur grand nombre ralentira la recherche dans le graphe. En pratique, nous déterminons les régions d'un critère d'homogénéité qui tolère des variations de quelques % sur l'altitude et la confiance des régions. Avec une tolérance de 5 cm sur l'altitude et de 5 % sur la confiance, le modèle précédent produit 450 régions, de surface moyenne égale à 2 m², qui définissent 5500 nœuds (figures 4.12 et 4.13) : cela est très largement suffisant pour la navigation.

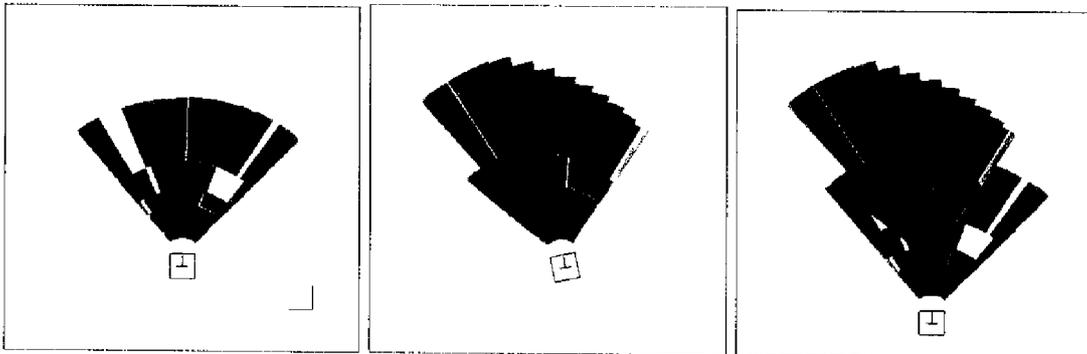


Figure 1.9 : Fusion des étiquettes de deux images laser classifiées : 1ère et 2èmes vues, et fusion des deux (le carré représente les positions d'Adam lors des acquisitions : il a 3 mètres de côté)

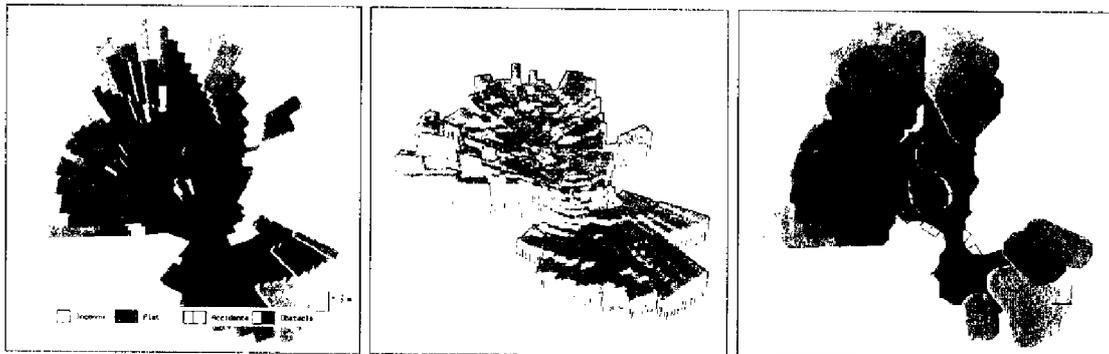


Figure 4.10 : Classes de terrain, altitudes, et étiquettes après grossissement d'un modèle issu de la fusion de 8 images laser (les classes de terrain sont représentées avec des couleurs qui indiquent leur confiance)

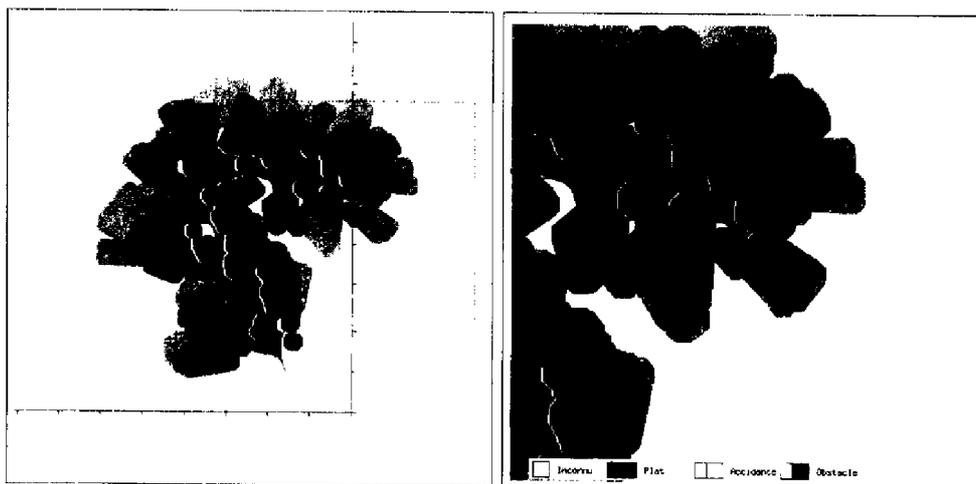


Figure 4.11 : Un modèle construit sur la base de 18 perceptions : carte de régions globale et bitmap courant

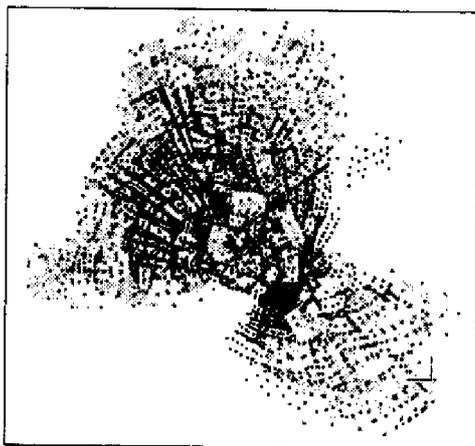


Figure 4.12 : 3000 régions déterminées avec un critère d'homogénéité des pixels strict définissent 24000 nœuds

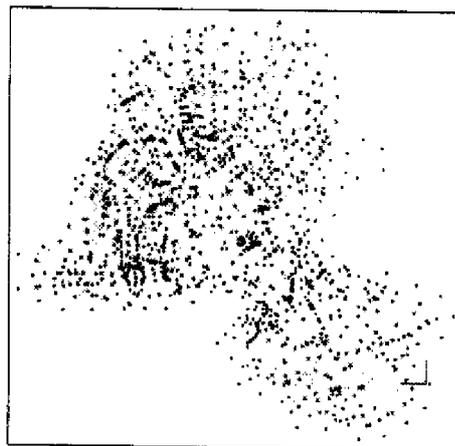


Figure 4.13 : 450 régions déterminées avec un critère d'homogénéité des pixels "tolérant" définissent 5500 nœuds

4.3 Gestion du Modèle

4.3.1 Un bitmap "glissant"

Le principal inconvénient du bitmap est sa taille mémoire, qui devient vite très grande dans notre cas où l'ensemble des informations codées dans chaque pixel nécessite en tout 8 octets : une zone de 50x50 m décrite avec une échelle de 10 cm occupe 2 Mo. La carte des régions construite sur le bitmap est elle bien plus compacte : ainsi les 450 régions du modèle précédent occupent en tout 65 Ko.

Nous avons donc choisi de conserver la carte de régions sur l'ensemble de la mission, et le bitmap se déplace avec le robot : cela nécessite notamment d'être capable d'effectuer la procédure de structuration localement (sur les zones qui sont modifiées par la dernière perception), mais aussi de pouvoir "replonger" les régions dans le bitmap lorsque l'on repasse sur une zone déjà perçue. La carte de régions de la figure 4.11 occupe en tout 600 Ko, elle couvre près de 4000 m².

4.3.2 Introduction des données initiales

Les données initiales concernant le tronçon courant doivent être considérées par la fonction navigation, et sont donc utilisées pour initialiser la carte de régions et le bitmap. Outre des informations générales concernant la mission (telles les zones de visibilité des amers lointains par exemple), ces données décrivent le type de terrain supposé : cette description peut se faire *via* le biais de probabilités partielles, ou en indiquant directement les modes de locomotion applicables.

Nous considérons dans notre cas qu'elles sont elles aussi décrites par un bitmap, de résolution

grossière (de l'ordre de quelques mètres), dont les pixels appartiennent à l'une des quatre classes suivantes :

- La classe "réflexe", où l'environnement est supposé suffisamment plan pour que le robot puisse évoluer en mode réflexe ;
- La classe "planifié plan", pour laquelle le robot est contraint de planifier ses déplacements ;
- La classe "difficile", pour laquelle il est supposé que l'environnement est très contraint ;
- Et enfin la classe "interdit", qui sert notamment à contraindre l'évolution du robot dans le tronçon d'itinéraire courant.

4.3.3 Performances

L'ensemble de la procédure de mise à jour prend quelques secondes à chaque nouvelle perception, réparties comme suit :

<i>Fonction</i>	Décalage du bitmap	Fusion	Grossissements	Mise à jour de la carte de régions	Total
<i>Temps (s)</i>	0.67	0.12	1.50	1.88	4.17

4.4 Vers une modélisation hétérogène

Nous avons vu que de nombreuses représentations différentes du terrain sont nécessaires pour mener à bien une tâche de navigation autonome. Il n'existe pas à notre connaissance de modèle mathématique "universel" qui unifie l'ensemble des informations contenues dans ces différentes représentations : elles doivent donc coexister ensemble.

4.4.1 Un cadre structurel

Il y a un grand intérêt à organiser ces différentes représentations qui coexistent au sein du système selon une hiérarchie fonctionnelle : cela permet de gérer leur mise à jour dans un cadre structurel simple. La figure 4.14 montre les diverses représentations d'un environnement extérieur construites par des algorithmes développés au LAAS : les flèches indiquent les données nécessaires à la construction d'une représentation.

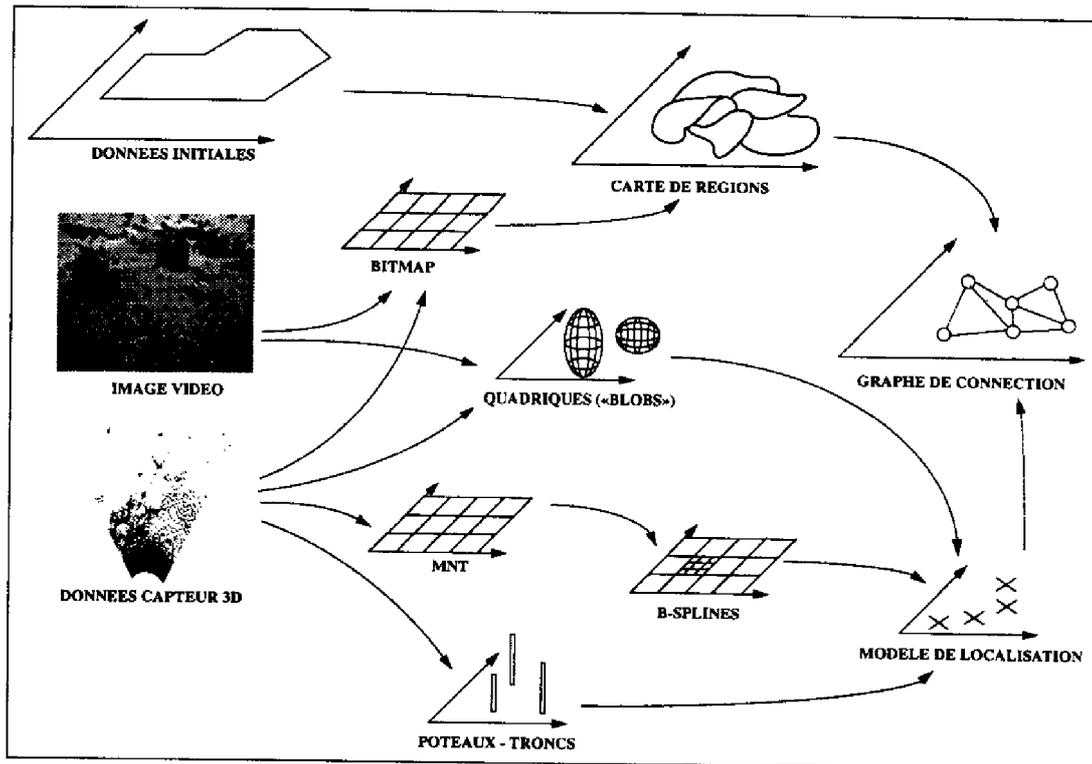


Figure 4.14 : Les différentes représentations de l'environnement présentes dans le système

Dans cette figure, on distingue :

- Les informations fournies par les capteurs (images vidéo et images de points 3D) et les données initiales concernant le terrain ;
- Les représentations dont nous venons de présenter la construction (bitmap, carte de régions et graphe de connexion) ;
- Le modèle numérique du terrain (MNT [Nashashibi 93]) ;
- Une représentation dédiée au recalage : le modèle B-splines, qui permet d'extraire du MNT des zones d'intérêt pour la procédure de recalage par corrélation [Fillatreau 93b] ;
- Deux représentations dédiées au recalage et à la *localisation*, basées sur une description de la scène en termes d'objets (modélisés par des ellipsoïdes [Betge-Brezetz 94] ou des poteaux [Fillatreau 93a]) ;

Remarque : Nous distinguons le terme "recalage" du terme "localisation" comme suit : une procédure de recalage est appliquée pour corriger la position du robot entre deux positions successives (l'incertitude sur la position est suffisamment faible), alors qu'une procédure de localisation est appliquée lorsqu'une zone précédemment modélisée est perçue à nouveau après un *grand nombre* d'étapes de déplacement (cas d'une boucle par exemple) : dans ce cas, l'incertitude sur la position du robot est telle qu'une procédure de recalage par corrélation n'est pas réalisable en un temps raisonnable. Il est alors nécessaire

de disposer d'une procédure de "reconnaissance de lieux", ce que peut permettre une représentation composée d'objets distincts modélisés.

Bien entendu, ce schéma n'est pas exhaustif : rien n'empêche d'y ajouter d'autres représentations. Son principal intérêt est d'explicitier les dépendances entre les différentes représentations, il permet ainsi d'organiser aisément les différents traitements concernant la modélisation de l'environnement.

Par ailleurs, toutes les représentations ne sont pas construites systématiquement : seule la carte de régions est mise à jour à chaque acquisition de données, et c'est la fonction navigation qui détermine quelles autres représentations doivent être éventuellement mises à jour.

4.4.2 Maintien de la cohérence

Chacune des différentes représentations présentes dans ce schéma décrit le *même environnement* sous des formes diverses, et elles doivent donc être cohérentes entre elles. Il y a deux sources possibles d'incohérences dans les représentations : les erreurs de modélisation et les erreurs de localisation.

- Les algorithmes de traitement et la structure des modèles construits tolèrent les incohérences de modélisation au sein d'une représentation (grâce aux incertitudes), mais il est aussi nécessaire de maintenir la cohérence *entre* les diverses représentations. Ainsi les modèles d'objets basés sur des ellipsoïdes permettent de détecter certains obstacles : il peuvent servir à améliorer la confiance sur l'étiquetage (voir à modifier l'étiquetage lui-même) des régions correspondantes dans la carte de régions. De même, le modèle B-spline peut permettre de vérifier que des régions étiquetées planes le sont réellement. Le maintien de la cohérence entre l'ensemble des représentations reste un problème ouvert qui a beaucoup d'intérêt : la qualité de chaque représentation serait améliorée ;
- Les erreurs de localisation ne sont tolérées que par les représentations "discrètes", *ie* basées sur des objets : la position de chacun des objets est connue avec une incertitude qui est mise à jour au fur et à mesure des procédures de recalage-fusion. Par contre, les représentations ayant un caractère "continu" (*ie* qui modélisent l'environnement sous forme "surfactive" - MNT, bitmap, carte de régions et modèle B-splines) ne tolèrent pas de variations : elles sont basées sur une structure rigide (généralement à deux dimensions), et requièrent donc que la position du robot soit précisément connue pour leur construction. Nous ne connaissons pas de contributions concernant la construction de tels modèles de terrain dont la structure peut évoluer lors de leur construction, à la manière d'une surface déformable : il s'agit aussi d'un problème ouvert.

Seconde Partie

Stratégies de navigation



Chapitre 5

Analyse du problème

Nous avons vu à la fin du second chapitre l'importance du planificateur de chemin (ou fonction navigation) en tant que processus de décision : à chaque étape de l'exécution de la mission "Aller à", c'est lui qui détermine le sous-but à rallier, le mode de déplacement à appliquer pour le rallier, et la tâche de perception à y effectuer.

Nous allons consacrer ce chapitre à bien préciser ce que nous devons résoudre : nous commençons par présenter les particularités de notre problème, et recensons les décisions (et les possibilités de choix associées) qui sont à la charge de la navigation. Nous présentons ensuite les diverses informations nécessaires à la prise de ces décisions.

5.1 Présentation

5.1.1 Problèmes comparables

À l'origine, le terme "navigation" s'applique bien évidemment au domaine maritime, et est aussi très utilisé en aéronautique. Par rapport à la navigation d'un robot mobile telle que nous l'avons définie dans le premier chapitre, il n'est dans ces cas pas besoin de modéliser géométriquement l'environnement, puisqu'il est toujours précisément connu (on dispose de cartes) ; mais il est par contre fondamental de s'y localiser précisément : ainsi la plupart des ouvrages qui traitent de la navigation appliquée à ces domaines se restreignent souvent à l'étude des moyens de localisation [Coutaud 78]. Si ces problèmes de localisation sont désormais résolus grâce à l'utilisation de techniques modernes (radio-navigation, positionnement assisté par satellite), il n'en reste pas moins qu'il est nécessaire de planifier les déplacements qui permettent de rallier le but fixé : on parle de "plan de vol" en aéronautique, et de "route" en navigation maritime.

Dans le cadre de la navigation aérienne, le problème est sensiblement différent du nôtre : l'environnement est modélisé de façon certaine (par l'intermédiaire de couloirs de navigation), et il n'y a quasiment pas d'incertitudes liées aux prévisions météorologiques : la vitesse de déplacement des avions est suffisamment grande devant la durée pour lesquelles ces prévisions sont fiables. Le plan de vol est donc déterminé *a priori*, en optimisant le temps

nécessaire grâce aux courants aériens qui sont connus ; et les perturbations météorologiques locales sont évitées en cours de vol, à la manière dont un robot évite des obstacles épars sur sa trajectoire. La difficulté ne réside ici que dans les contraintes dues à la présence de nombreux autres avions : on parle de “circulation” aérienne, mais c’est un autre sujet.

Les incertitudes liées aux prévisions météorologiques rendent le problème de la navigation des bateaux (et particulièrement du routage des voiliers dans les courses transocéaniques) plus intéressant pour nous : le temps nécessaire à rallier le but est cette fois-ci bien supérieur à la durée T pour laquelle ces prévisions sont fiables. Mais l’on dispose néanmoins de statistiques météorologiques sur l’ensemble de la zone à traverser : on connaît, pour les différentes périodes de l’année et les différentes zones de navigation possibles, la probabilité que le vent souffle dans une direction donnée avec une certaine vitesse moyenne (on peut assimiler ces informations avec nos données initiales concernant un tronçon d’itinéraire). Le problème du routage optimum est alors souvent résolu comme suit : à l’instant t , une route globale statistiquement optimale est déterminée pour rallier la position courante à l’arrivée, et les prévisions météorologiques sont utilisées pour déterminer le chemin local dont l’extrémité prévue à la date de fin de validité des prévisions ($t + T$) se rapproche le plus de cette route globale. Dans [Alquier 86] et [Wang 93], la détermination des routes locales et globales se fait par programmation dynamique sur une grille de points de passage préalablement définie.

5.1.2 Spécificité de notre problème

Une analogie plus en rapport à notre problème est celle d’un explorateur qui cherche à traverser une région inconnue afin d’atteindre un but plus ou moins précis : cet explorateur dispose de moyens d’orientation (sa boussole par exemple, mais aussi sa capacité à reconnaître des lieux), de moyens de perception à distance (ses yeux et sa longue-vue), et de moyens de locomotion (disons son cheval). Des rumeurs lui ont indiqué quelques points de passage possibles, ainsi que des régions réputées infranchissables. Bien entendu, de nombreux éléments indépendants de sa volonté contraignent son exploration : il doit notamment rentrer chez lui avant l’hiver et s’arrêter chaque nuit afin de se reposer.

Au fur et à mesure de son exploration, l’explorateur consigne la description des régions qu’il a traversées dans une carte (figure 5.1), et il doit très souvent effectuer des choix stratégiques. Au point où il arrive sur notre figure, il peut notamment se poser les questions suivantes : Faut-il tenter de traverser la forêt qui semble mener directement au but, au risque de se retrouver dans la zone hostile ? N’y a-t-il pas intérêt à revenir en arrière, pour monter sur les collines et observer la plaine ? N’existe-t-il pas un gué plus au nord, qui permettrait de traverser la rivière ?...

De manière plus sérieuse, cette analogie montre qu’il existe deux points particuliers qui distinguent notre problème de navigation des domaines de la navigation aérienne et maritime :

- L’environnement n’est pas totalement connu sur l’ensemble du tronçon : on ne dispose que de rares informations à l’initialisation de sa traversée, et les informations que l’on a pu acquérir en cours de traversée (principalement la carte de régions) demeurent *partielles et incertaines* ;

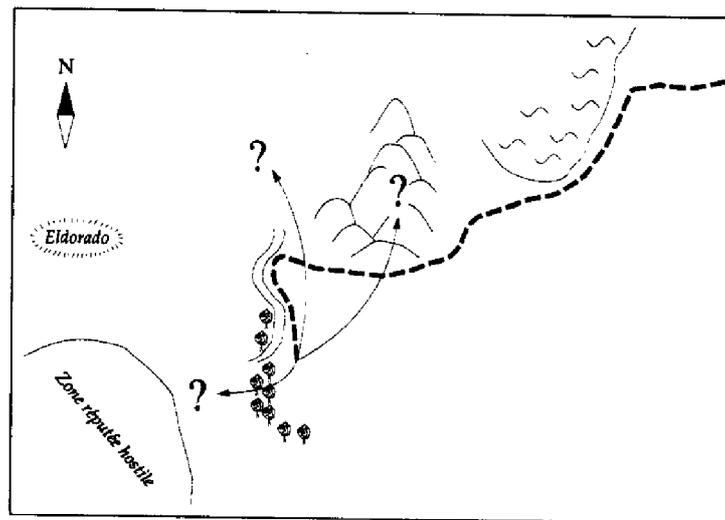


Figure 5.1 : Quelle décision prendriez-vous ?

- Si la finalité de la navigation est bien de rallier un but, donc de générer des déplacements, nous avons vu qu'il est pour cela nécessaire de modéliser l'environnement. Mais celui-ci n'étant pas connu *a priori*, une des actions dont nous devons décider est justement de la mise à jour de la connaissance que l'on en a. C'est donc un peu comme si le voilier, ne disposant pas de prévisions, devait se rendre en chaque endroit dont il veut connaître les caractéristiques météorologiques.

C'est ce second point qui rend la navigation difficile, car les actions de perception et de déplacement sont *fortement interdépendantes* : se déplacer nécessite d'avoir préalablement modélisé l'environnement, donc d'avoir effectué une ou plusieurs tâches de perception ; mais inversement il est souvent nécessaire de se déplacer pour pouvoir effectuer une tâche de perception.

5.2 Les actions à déterminer

5.2.1 Actions de déplacement

Déterminer une tâche de déplacement, c'est tout d'abord définir *où* aller (détermination du sous-but), mais aussi *comment* y aller (détermination du mode déplacement) :

- déterminer **comment** se déplacer consiste à choisir un mode de déplacement parmi les trois que nous avons retenus dans notre approche : en mode planifié sur terrain plan (2D), en mode planifié sur terrain accidenté (3D) ou en mode réflexe sur les terrains dégagés ;
- déterminer **où** se déplacer consiste à sélectionner un objectif intermédiaire (sous-but) situé *dans la zone couverte par la carte de régions*. Si les données initiales le permettent,

le mode réflexe peut éventuellement être appliqué sur une zone qui n'a pas encore été perçue par les capteurs du robot.

Une fois un sous-but et un mode de déplacement sélectionnés, la fonction navigation doit aussi préciser au planificateur de trajectoire le modèle de l'environnement sur lequel il va travailler, et particulièrement les limites que la trajectoire planifiée ne doit pas franchir.

5.2.2 Actions de perception

Nous avons vu dans le chapitre 2 que les représentations sont nécessaires aux deux principales fonctionnalités suivantes : *planifier* les diverses actions du robot, et *localiser* le robot dans l'environnement. Exécuter une tâche de perception, ce qui revient à mettre à jour une des représentations de l'environnement, répond donc à un des deux besoins fondamentaux suivants :

- **modéliser** l'environnement afin de pouvoir y planifier des déplacements ;
- **localiser** le robot dans cet environnement, afin de garantir sa sécurité, mais aussi la cohérence des représentations construites.

Déterminer laquelle de ces fonctions est nécessaire définit le "besoin perceptuel", et revient à répondre à la question : **quoi** percevoir ? Mais il faut aussi déterminer **comment** percevoir, c'est-à-dire répondre aux questions suivantes : Quel(s) capteur(s) utiliser ? À partir de quel point de vue ? Dans quelles conditions (modalités de fonctionnement) ? Quel traitement appliquer aux données acquises ?

La détermination de ces différents paramètres n'est pas triviale, et nous nous focaliserons essentiellement sur la détermination du point de vue, le choix du capteur et de ses modalités de fonctionnement étant directement déduit du besoin perceptuel à l'aide de règles prédéfinies : ainsi une tâche de recalage se fera toujours à la résolution maximale. Nous verrons néanmoins dans le chapitre suivant qu'il existe des moyens de déterminer certaines modalités de fonctionnement en fonction de l'état courant de l'environnement.

5.2.3 Autres actions

Il existe des décisions qui ne concernent ni le choix du sous-but à rallier et du planificateur de trajectoire, ni les tâches de perception dédiées à la génération des déplacements. Le plan de mission peut par exemple comporter des instructions telles que "acquérir un panorama dès que le robot est à une altitude supérieure à x mètres", ou "collecter des échantillons sur la première zone accidentée rencontrée dans telle zone". Mais si la navigation est un élément clé du niveau décisionnel du robot, elle n'est pas "tout" le niveau décisionnel, et d'autres données que les siennes doivent aussi être prises en compte : nous considérons donc que de telles actions ne sont pas à sa charge.

Par contre, la navigation peut échouer : il peut arriver qu'il soit impossible d'exécuter la mission dans les conditions spécifiées, parce que le tronçon courant s'avère impossible à traverser par exemple, ou parce que les déplacements s'exécutent de façon bien trop lente pour garantir que la mission s'exécute dans le temps imparti. Afin de permettre au niveau décisionnel d'analyser ces cas d'échecs, la fonctionnalité navigation doit lui retourner les informations nécessaires, et notamment les contraintes qui ne peuvent être respectées.

5.3 Éléments de décision

Nous avons regroupé les différentes informations nécessaires à la navigation en trois catégories distinctes, suivant leur origine : (i) les informations relatives au contexte de la mission à exécuter ; (ii) les informations relatives aux capacités du robot (déplacement et perception), et (iii) les informations relatives à l'environnement.

5.3.1 Informations concernant le contexte de la mission

Sans entrer dans un formalisme de langage de description de mission, nous pouvons néanmoins spécifier les éléments de la définition de mission qui concernent la navigation.

a) Définition du but global

Il s'agit bien entendu d'une information primordiale. Le but peut être spécifié par l'intermédiaire d'une position géométrique, définie précisément (point dans l'environnement) ou non (région de l'environnement, telle la frontière extrémité du tronçon courant par exemple) ; mais il peut aussi être spécifié par l'intermédiaire d'une description plus générale : première zone plane située à telle altitude, premier objet de volume supérieur à $x m^3$...

Ces moyens de spécifier un but requièrent de la part du robot une capacité de détection du but : elle est fournie par une simple consultation de la position du robot dans le premier cas (d'où l'importance d'une localisation suffisamment précise et fiable), et par des algorithmes de "reconnaissance" dans le second cas, algorithmes que nous considérons acquis.

b) Les critères à considérer

D'une manière générale, toute nouvelle action du robot doit être décidée en respectant les trois critères fondamentaux suivants :

1. **Le temps** : le robot gagne en efficacité en gagnant du temps ;
2. **L'énergie** : les ressources en énergies étant limitées, il est nécessaire de les économiser ;
3. et **l'incertitude sur la position du robot**, dont on a intérêt à ce qu'elle soit suffisamment faible pour permettre le succès de la tâche.

Chacun de ces trois critères peut être modulé : ainsi certaines missions peuvent être programmées sur une durée qui laisse beaucoup de "temps libre", ou commencer avec des ressources

énergétiques très largement suffisantes par exemple. C'est dans la définition de la mission que sont précisées les importances relatives de ces critères, et leur prise en compte par la fonction navigation se fera lors de la détermination du coût de chaque action envisagée.

La seule difficulté résulte alors du fait que ces critères peuvent être contradictoires : il peut arriver des situations où l'on peut rallier très rapidement le but global, mais au prix d'une très grande consommation d'énergie (si le chemin le plus court comporte de nombreuses pentes par exemple), ou réciproquement.

Remarque Signalons que ces critères peuvent évoluer en cours d'exécution de la mission (et ce pour mille raisons différentes : la consommation moyenne d'énergie peut s'avérer bien supérieure à celle prévue, le robot a mis beaucoup de temps à réaliser qu'un itinéraire est une impasse...). La gestion de ces critères est à la charge du niveau décisionnel, qui surveille et analyse le déroulement de la mission. En ce qui nous concerne, nous considérons que ces critères sont une donnée d'entrée de la fonction navigation à chaque étape du déplacement.

Notons enfin qu'il peut être nécessaire d'introduire une notion générale de "risque" dans les critères à considérer : un tel risque pourrait par exemple être relié aux zones de visibilité d'amers lointains ou aux zones de visibilité radio avec l'opérateur.

c) Les contraintes à respecter

La définition des diverses contraintes à respecter est importante, c'est notamment elle qui permet de définir la notion d'échec pour la fonction navigation : tout cas d'échec correspond en effet à la violation d'une de ces contraintes.

Il y a d'abord des contraintes qui sont "universelles", dans le sens où elles ne dépendent pas de la mission courante et où elles n'évoluent pas lors de son exécution :

- **La sécurité** : la sécurité du robot doit bien évidemment être garantie à tout moment. Cette contrainte fondamentale est d'abord considérée dans la conception des fonctions de modélisation (les non-détections sont proscrites, quitte à générer plus de fausses alarmes) ; elle est ensuite prise en compte par la fonction navigation, qui préférera les zones sûres aux zones risquées ; puis par les planificateurs de trajectoire, qui testent des contraintes de stabilité et de non-collision du véhicule ; et en dernier lieu par le niveau fonctionnel du robot, qui surveille l'exécution des trajectoires planifiées ;
- **Cohérence des représentations construites** : la gestion et la mise à jour des représentations construites sont à la charge de la fonction navigation (*via* la détermination des tâches de perception), et il est fondamental qu'elles soient cohérentes avec l'état réel de l'environnement. Nous avons vu que les fonctions de modélisation et la structure des modèles construits tolèrent les erreurs de modélisation, mais pas les erreurs de localisation. C'est la fonction navigation qui garantit que l'incertitude sur la position du robot reste suffisamment faible en contrôlant les actions de localisation ;

Il y a ensuite les contraintes spécifiques à chaque mission, dont la description est un élément du plan de mission :

- **Temps** : Un temps maximal peut être associé à l'exécution d'une mission ;
- **Énergie** : Une consommation d'énergie maximale peut aussi être imposée pour l'exécution d'une mission ;
- **Itinéraires** : La description géométrique du tronçon à traverser contraint l'évolution du robot. Cette contrainte est en fait prise en compte dans la carte de régions par le biais de la description initiale ;

Le plan de mission peut aussi être accompagné d'autres spécifications qui limitent les choix possibles pour chaque décision à prendre. De telles contraintes simplifient en fait le travail de la fonction navigation, en imposant par exemple un mode de déplacement particulier dans certaines zones du tronçon.

5.3.2 Informations concernant les capacités du robot

Pour déterminer les "meilleures" actions de déplacement et de perception à exécuter, il faut être capable de prévoir le résultat de chacune de ces actions possibles, afin d'en estimer l'intérêt dans le contexte de la mission à exécuter, et donc de faire un choix. Cette capacité de prédiction requiert la connaissance des modèles des possibilités du robot, qui seront appliqués sur les modèles de l'environnement. Nous présentons ici ces modèles d'action de manière générale, nous aurons l'occasion d'y revenir plus précisément dans le prochain chapitre.

a) Modèles des actions de déplacement

Les actions de déplacement sont les plus simples à prévoir, leurs principales conséquences étant de modifier la position du robot, de prendre du temps et de consommer de l'énergie.

Mais lors de l'exécution d'un déplacement, la position du robot est asservie sur les données fournies par ses capteurs proprioceptifs (codeurs odométriques, centrale inertielle, inclinomètres, accéléromètres...), dont les mesures ne sont pas parfaites, et qui génèrent donc une erreur cumulative sur les paramètres (X, Y, Z, θ) de la position du robot¹. Il s'agit d'un aspect qui doit faire partie du modèle des actions de déplacement du robot.

b) Modèles des actions de perception

Les modèles des actions de perception sont plus délicats à obtenir, principalement parce qu'ils intègrent *toute* la chaîne de traitement des données, depuis leur acquisition jusqu'à la production d'une représentation.

La notion de "capteur logique" [Henderson 85b, Henderson 85a] est particulièrement adaptée pour décrire une action de perception : ce modèle structurel définit un capteur par le type de ses entrées et de ses sorties, et permet d'organiser les différents capteurs d'un système multi-sensoriel en un réseau de transmission de données. Un capteur logique peut être constitué d'une succession de différents capteurs logiques, et ses propriétés sont héritées de

¹les angles de tangage et de roulis ont une erreur constante, non cumulative

celles de ses constituants : ainsi un ban stéréoscopique définit un "extracteur de points 3D", et notre procédure de classification définit un "classificateur de terrain".

Cette modélisation structurelle ne permet pas de prédire le résultat de l'activation d'un capteur logique : c'est le rôle du *modèle perceptuel*, que nous présenterons dans le chapitre suivant.

5.3.3 Informations concernant l'environnement

Il s'agit de la description initiale du tronçon d'itinéraire et des informations acquises par les capteurs du robot : nous avons vu dans la première partie du mémoire comment ces informations sont toutes deux représentées par la *carte de régions*, et n'y revenons donc pas ici. La fonction navigation ayant à sa charge le contrôle des actions de localisation extéroceptives, elle doit aussi avoir accès au modèle de localisation (*carte d'amers*).

La figure 5.2 présente de manière synthétique les différentes entrées et sorties de la fonction navigation.

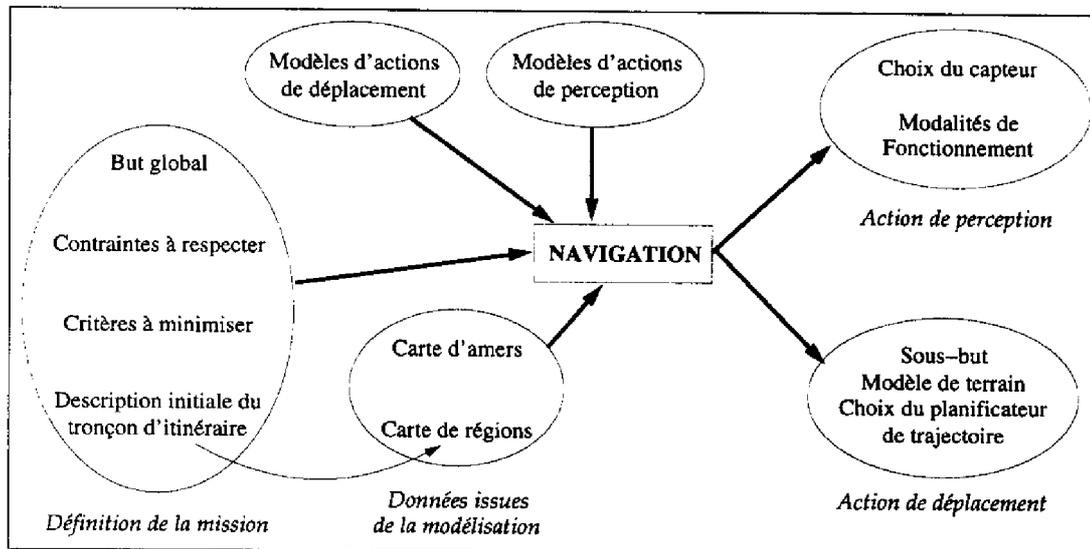


Figure 5.2 : Les entrées et sorties de la fonction navigation

Chapitre 6

Approche proposée

Le planificateur de chemin a donc la charge de déterminer les actions de déplacement et de perception du robot. Nous allons tout d'abord présenter dans ce chapitre les techniques nécessaires à la planification de chacune de ces actions indépendamment l'une de l'autre, en insistant particulièrement sur la planification de perception. Nous ébaucherons ensuite une approche que nous qualifions "d'exhaustive", dans le sens où elle utilise de manière explicite l'ensemble des informations qui sont à la disposition du robot pour résoudre le problème, et où elle planifie *en même temps* les actions de perception et de déplacement. Nous concluerons alors sur la difficulté et la complexité algorithmique de cette approche exhaustive, et proposerons une approche plus simple, dont l'efficacité a été montrée par des expérimentations. Sa présentation clôt le chapitre, et ses résultats sont analysés dans le chapitre suivant.

6.1 Planification d'une tâche de perception : généralités

La capacité de contrôler activement les moyens de perception d'un système ouvre de nombreuses possibilités, et permet notamment d'améliorer son degré d'autonomie [Bajcsy 88] : chercher à exhiber les éléments d'information nécessaires à la réalisation d'une tâche fixée (qui n'est pas forcément une tâche d'interprétation) est un comportement "intelligent", qui démontre la capacité d'un système à raisonner sur la tâche à accomplir et sur ses propres capacités, et ce en fonction de l'état de l'environnement. Le thème de la "perception active" est désormais considéré par de nombreux auteurs, mais est très vaste et recouvre en fait des aspects très variés (voir à ce sujet le numéro spécial d'International Journal of Computer Vision [IJCV 93], et une revue des principales contributions existantes dans [Christensen 94]). On distingue notamment les approches "réactives" des approches délibératives : ainsi on parle de "vision réactive" ou de "contrôle d'attention" lorsque les paramètres de contrôle d'un capteur sont déterminés en boucle fermée sur les percepts perçus, sans processus d'interprétation ou de planification (suivi de cible mobile et objectif autofocus par exemple), et on parle plutôt de "choix de point de vue" lorsque qu'une représentation de l'environnement (début d'interprétation) et des modèles des capacités perceptuelles sont explicitement utilisés.

Nous nous plaçons résolument dans le second cas : la tâche que le robot doit accomplir est clairement définie à long terme, et les actions de perception sont planifiées sur la base des représentations de l'environnement que l'on construit, après prédiction et évaluation de l'intérêt de leur activation.

6.1.1 Modélisation des actions de perception

Pour être capable de prédire le résultat de l'activation d'une tâche de perception, il est bien entendu nécessaire de connaître son comportement, c'est-à-dire de la *modéliser*. C'est le rôle du *modèle perceptuel*, qui est associé à chaque capteur logique défini dans le système. Nous décomposons le modèle perceptuel en quatre parties [Grandjean 91, Lacroix 94b] :

- *Modèle de spécificité* : issu de la définition du capteur logique, il spécifie les attributs de l'environnement que le capteur permet de percevoir (géométrie pour les capteurs de points 3D, classe de terrain pour notre fonction de classification...) ;
- *Modèle de détectabilité* : il indique les contraintes perceptuelles sous lesquelles le capteur retourne effectivement des informations : ainsi les attributs de l'environnement ne sont pas perçus s'ils sont occultés (contrainte de visibilité), s'ils sont trop éloignés (contrainte de champ de vue), s'ils concernent une très petite région de l'environnement (contrainte de résolution) ou s'ils sont perçus avec un très faible angle d'incidence par un télémètre laser (contrainte de spécularité). Ces contraintes définissent des régions dans l'espace des configurations du capteur au sein desquelles une primitive donnée de l'environnement est effectivement perceptible [Ikeuchi 87, Cowan 89] ;
- *Modèle d'incertitude* : il représente les incertitudes sur les informations retournées par le capteur logique. Il s'agit de la partie la plus importante du modèle perceptuel, mais aussi la plus délicate à obtenir : les incertitudes sur les données retournées par le capteur physique, déjà difficiles à modéliser précisément, sont propagées dans l'ensemble des traitements et deviennent alors difficiles à expliciter. Une analyse statistique des résultats du capteur logique permet néanmoins d'estimer ce modèle d'incertitude ;
- *Modèle d'état* : il spécifie l'influence des paramètres d'état du capteur (ou modalités de fonctionnement : zoom, champ de vue, résolution...) sur les modèles de détectabilité et d'incertitude.

On trouvera dans [Ikeuchi 87] une étude précise des modèles de différents capteurs appliqués à la perception de primitives géométriques précisément définies (segments, surfaces, sphères). Les modèles des actions de perception nécessaires à la navigation que nous cherchons à planifier (classification et recalage) sont présentés plus loin dans ce chapitre.

6.1.2 État de l'art

a) *Planification de perception pour la reconnaissance d'objets*

Dans ce type d'application, le problème se ramène à déterminer un point de vue (et éventuellement à choisir un capteur) qui permette de minimiser l'ambiguïté émise par un système d'interprétation pour une hypothèse d'objet. Cette ambiguïté concerne aussi bien la précision sur la position de cet objet que la nature même de cet objet.

Ikeuchi et Kanade [Ikeuchi 87] ont abordé ce problème, et notamment développé des modèles de détectabilité et d'incertitude de différents capteurs. Les auteurs ont déduit de ces modèles des stratégies de reconnaissance d'objet générées automatiquement [Ikeuchi 88b, Ikeuchi 88a].

Nous avons nous-mêmes abordé ce sujet dans le cadre du projet Esprit SKIDS [Lacroix 92, Lacroix 94b] : notre contribution a porté sur le développement d'une fonctionnalité de planification de perception dédiée à la confirmation par le système d'interprétation d'une hypothèse sur la présence d'un objet dans la scène. Planifier une tâche de perception consiste dans ce contexte à sélectionner un capteur, à choisir un point de vue s'il est mobile, et à déterminer ses modalités de fonctionnement les mieux adaptées à la tâche de confirmation.

Notre approche était basée sur une représentation particulière des objets de l'environnement, le "graphe d'aspects perçus", qui prend en compte les capacités des capteurs de la machine et les caractéristiques du système de reconnaissance. Ces graphes d'aspects perçus sont déterminés hors-ligne par une procédure de compilation des modèles des objets et des capteurs, qui permet de déterminer la meilleure tâche de perception pour chaque point de vue possible autour de l'objet, ainsi que les informations (primitives perceptuelles) qu'elle va permettre d'extraire et son utilité pour la tâche de confirmation.

- La détermination de la meilleure tâche de perception pour un point de vue donné (un aspect de l'objet) est réalisée heuristiquement en optimisant un rapport précision/volume des données ; elle définit une direction de visée, le champ de vue et la résolution angulaire des capteurs. Cette tâche optimale déterminée, son résultat est prédit en appliquant les modèles de détectabilité des capteurs, qui permettent de déterminer les primitives d'un objet effectivement perceptibles. Les modèles des objets sont polyédriques, et nous avons considéré les trois capteurs logiques suivants : extracteur de facettes 3D (à partir de données télémétriques), de segments 2D, et de régions colorées (à partir de caméras CCD).
- L'ensemble des tâches de perception déterminées est alors ordonné en fonction de leur utilité pour la tâche de confirmation. Ceci est réalisé grâce à la notion de "facteur discriminant", qui définit pour chaque objet les ensembles de primitives qui permettent à la fois une bonne interprétation et une bonne précision de son placement dans l'environnement. Un ensemble de facteurs discriminants est défini pour chaque objet par un expert qui connaît les performances du système d'interprétation des données, et la prédiction du résultat de chaque tâche définie dans le graphe d'aspects perçus permet alors d'exprimer leur utilité.

Une fois une hypothèse d'interprétation émise par le système de reconnaissance, les contraintes perceptuelles (visibilité et champ de vue) permettent d'extraire du graphe d'aspects perçus les tâches de perception effectivement réalisables, et une optimisation du rapport utilité/coût de chacune d'entre-elles permet alors de sélectionner la plus appropriée, ce qui

définit aussi un choix du capteur à utiliser. La méthode a été intégrée au système de reconnaissance multisensoriel et à un système de modélisation polygonale de l'environnement qui permet de déterminer les positions accessibles d'un capteur mobile. Elle a été expérimentée avec deux caméras fixes et une caméra montée sur le robot Hilare.

De nombreux autres auteurs se sont intéressés à ce problème particulier, que l'on peut voir comme une extension directe des problèmes d'interprétation de scène [Grimson 86, Cowan 88, Cowan 89, Hutchinson 88, Tarabanis 91].

D'une manière générale, retenons que ces méthodes sont basées sur des hypothèses d'interprétation, qui définissent *totalelement* le but perceptuel.

b) Approche par la théorie de la décision

En modélisant les capteurs par des fonctions stochastiques, qui prédisent la probabilité d'observer une certaine mesure en fonction des paramètres d'état du capteur et de l'estimée initiale de la valeur mesurée, des auteurs ont abordé le problème de l'estimation numérique des attributs d'un modèle [Durrant-Whyte 87, Durrant-Whyte 88, Hager 91]. Des applications à la recherche de points de vue (toujours basée sur une hypothèse initiale) utilisant de tels modèles figurent dans [Cameron 90, Wu 90]. Ces modèles sont essentiellement numériques, et ne comprennent pas la spécificité et le caractère partiel de la perception : ils sont considérés comme des appareils de mesure géométrique. Le formalisme de la théorie de la décision est toutefois très bien adapté à ce type de problème, et permet des développements rigoureux et efficaces.

c) Surveillance d'environnements connus

Surveiller un environnement initialement connu est une application classique de la robotique mobile, et a donné lieu à de nombreux développements formels dans des environnements 2D représentés par des polygones. La résolution du "problème du gardien de musée", qui consiste à déterminer le nombre minimal de points de vue qui permettent de percevoir un ensemble d'objectifs donné, a notamment été abordée dans [O'rourke 87] et [Moutarlier 91]. Les buts perceptuels étant connus, ce problème se rapproche dans un certain sens de ceux concernant le choix de point de vue pour la reconnaissance ; sa spécificité réside dans le nombre d'objectifs à percevoir, et dans le critère d'optimalité (nombre minimal de points de vue) à satisfaire. Les solutions proposées sont algorithmiquement complexes, et ne considèrent pas les incertitudes sur le modèle de l'environnement, ni les contraintes perceptuelles des capteurs.

d) Exploration d'environnements inconnus

On trouve de nombreuses contributions qui traitent de l'exploration d'un environnement initialement inconnu par un robot mobile. Par essence, ce problème nécessite la planification des tâches de perception qui permettent d'accroître la connaissance que le robot a de son environnement, mais il est souvent abordé en considérant que le robot perçoit une zone panoramique autour de sa position (ce qui est réaliste avec une ceinture de capteurs ultrasons). La plupart des contributions à ce problème concernent les environnements intérieurs, modélisés par des polygones [Rao 86, Lumelsky 90, Rao 91], des quad-trees [Zelinsky 92] ou des grilles [Elfes 91, Elfes 92].

Au LAAS, le cas du "robot curieux" a été abordé [Colly 89, Moutarlier 91] : l'environnement est initialement modélisé par un polygone sur la base des données percues par un télémètre laser effectuant une acquisition panoramique ; un objet est alors introduit dans l'environnement, qui est à nouveau perçu, et une stratégie visant à explorer les nouvelles zones occultées est utilisée pour déterminer un nouveau point de vue. Les contraintes de visibilité du capteur et d'accessibilité du robot sont considérées.

6.1.3 Conclusions

Cette rapide revue des contributions existantes à la planification de perception fait apparaître plusieurs points :

- Les actions de perception sont toujours déterminées dans le contexte d'une tâche particulière : reconnaissance d'objet, surveillance ou exploration (le cas de la planification d'une tâche de localisation sera examiné dans le paragraphe 6.4) ;
- La méthode générale consiste à déterminer tout d'abord le "but perceptuel", puis à rechercher la tâche qui permet de le percevoir ;
- Enfin, nous ne connaissons pas de contributions à la planification de perception dans des environnements extérieurs.

6.2 Détermination d'une tâche de recalage

Nous avons vu combien il est important que le robot connaisse avec précision sa position dans l'environnement, et qu'il est par conséquent nécessaire de planifier les acquisitions de données nécessaires au recalage de sa position. Nous présentons ici la technique que nous avons retenue pour prédire le résultat d'une tâche de recalage à partir d'une position donnée du véhicule et de la carte d'amers.

6.2.1 Modèle de la procédure de recalage

Le modèle d'une procédure de recalage extéroceptive est déterminé par le modèle de la fonction qui permet de percevoir un (ou plusieurs) amer(s) précédemment modélisé(s) (le capteur logique "extracteur d'amers"), et par le modèle de la fonction de recalage elle-même.

a) Modèle de perception des amers

Nous avons présenté dans le chapitre 4 les trois fonctionnalités d'extraction d'amers d'un environnement naturel développées au LAAS : extraction de poteaux et d'objets (rochers) à partir de données 3D brutes, et extraction de zones d'intérêt dans un MNT à partir de la représentation surfacique du terrain par des B-splines.

Chacune de ces trois instances du capteur logique "extracteur d'amers" doit être représentée par un modèle perceptuel composé des quatre champs que nous avons définis dans le paragraphe 6.1.1 : si le modèle de spécificité est immédiat, il n'en va pas de même des modèles de détectabilité et d'incertitude :

- Les contraintes de champ de vue et de résolution sont aisément vérifiées¹, mais la contrainte de visibilité est plus délicate : les positions du robot (donc du capteur) et de l'amer à percevoir étant connues avec une incertitude, et l'environnement étant lui-même incertain, le modèle de détectabilité devrait être une fonction aléatoire qui donne la probabilité que l'amer soit effectivement perçu (figure 6.1).

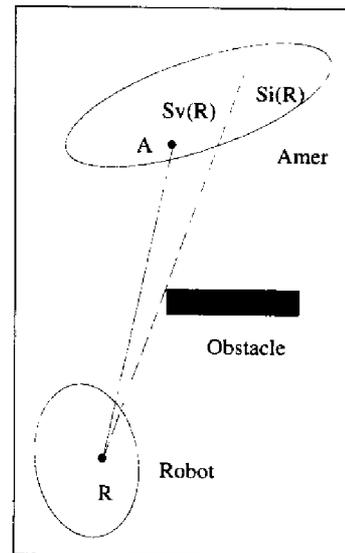
Figure 6.1 : Un modèle de détectabilité idéal en 2D : L'amer ponctuel et le robot sont positionnés avec une certaine incertitude. Si le robot est à la position R et l'amer à la position A comme sur la figure, alors l'amer est visible. En considérant l'incertitude sur la position de l'amer, la probabilité que l'amer soit visible depuis R est égale à la probabilité qu'il appartienne à la surface visible $S_v(R)$:

$$P(A = \text{vis} | R) = P(A \in S_v(R)) = \int_{x,y \in S_v(R)} P(A = x, y) dx dy$$

En introduisant l'incertitude sur la position R du robot, la probabilité pour que l'amer soit visible s'écrit :

$$P(A = \text{vis}) = \int_{x,y} P(A \in S_v(R)) P(R = x, y) dx dy$$

Ce modèle déjà complexe n'est même pas complet, puisqu'il ne prend pas en compte l'incertitude sur la position de l'obstacle (qui est l'incertitude sur l'élévation des pixels du MNT dans notre cas 3D), et qu'il considère l'amer comme étant ponctuel.



- Le modèle d'incertitude décrit la précision avec laquelle un amer perçu est localisé dans la perception courante. On peut ainsi estimer l'écart type sur l'altitude calculée en chaque pixel du MNT par une fonction quadratique de sa distance au capteur (figure 6.2) pour le cas où la fonction de recalage est basée sur une procédure de corrélation de zones modélisées par un MNT ; et de même on peut connaître l'incertitude avec laquelle un poteau est localisé dans une perception [Fillatreau 93a].

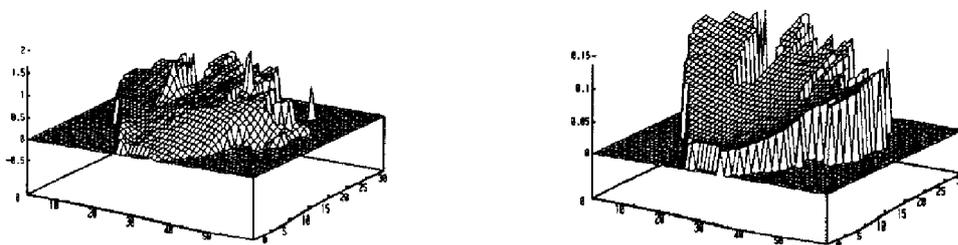


Figure 6.2 : MNT d'une perception et écarts types sur l'élévation [Nashashibi 93]

¹la contrainte de résolution est particulièrement importante pour l'extracteur de poteaux

b) *Modèle de la fonction de recalage*

Un modèle théorique de la fonction de recalage proprement dite nous donnerait une estimation de l'incertitude sur la position du robot après son application, grâce à une fonction du type :

$$(\mathcal{I}_{robot}^{Apr}, \{\mathcal{I}_{amers}^{Apr}\}) = \mathcal{F}(\mathcal{I}_{robot}^{Av}, \{\mathcal{I}_{amers}^{Av}\}, \{\mathcal{I}_{amers}^{local}\}),$$

où \mathcal{I}_{robot}^{Av} et les $\{\mathcal{I}_{amers}^{Av}\}$ sont les incertitudes initiales sur les positions du robot et des amers, les $\{\mathcal{I}_{amers}^{local}\}$ sont les incertitudes sur les positions des amers perçus dans la perception courante, et \mathcal{I}_{robot}^{Av} et les $\{\mathcal{I}_{amers}^{Av}\}$ sont les incertitudes initiales sur les positions du robot et des amers après la procédure de recalage. Le filtre de Kalman permet de déterminer une telle expression [Smith 86], qui n'est analytiquement pas simple, et ce d'autant plus si le nombre d'amers perçus est supérieur à 1.

6.2.2 Réalisation pratique

Les considérations précédentes montrent la difficulté d'obtenir un modèle précis de la procédure de recalage, particulièrement en terrain naturel. Pour les besoins de la fonction navigation, qui doit raisonner rapidement, nous avons défini un modèle plus simple, mais néanmoins fiable et réaliste

Ce modèle estime l'incertitude (écart type) sur la position du robot après recalage en fonction du nombre et de la distance à laquelle sont perçus les amers :

$$\mathcal{I}^{Apr} = \mathcal{F}(\{A_i, d_i\})\mathcal{I}^{Av}, \text{ avec } \mathcal{F}(\{A_i, d_i\}) = \prod_i f(A_i, d_i) \text{ et } f(A_i, d_i) = 1 - \sqrt{1 - \frac{d_i}{d_{max}}},$$

où \mathcal{I}^{Apr} et \mathcal{I}^{Av} sont respectivement les incertitudes sur la position du robot avant et après recalage, et d_{max} est la distance maximale à laquelle un amer est perceptible par le capteur.

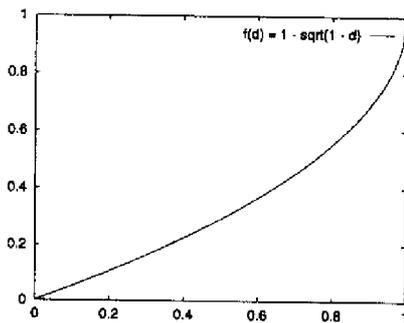


Figure 6.3 : Comportement de la fonction $f(d)$ pour un amer

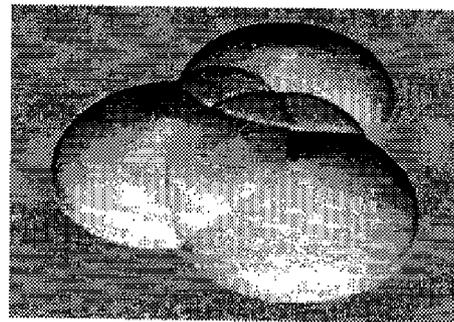


Figure 6.4 : Comportement de $1 - \mathcal{F}$ avec 3 amers

La fonction f prend ses valeurs entre 0 et 1 (figure 6.3), elle donne une réduction de l'incertitude sur la position du robot d'autant meilleure que la distance à laquelle est perçu

l'amer est faible. La figure 6.4 montre le comportement de la fonction $1 - \mathcal{F}$ dans un environnement dégagé d'obstacles, où sont présents trois amers.

Dans le même esprit de simplification, nous avons implémenté un test de visibilité basé sur l'algorithme de tracé de segment de Bresenham [Foley 90] (figure 6.5) : grâce à cet algorithme, on repère sur le bitmap les pixels qui joignent les positions du capteur et de l'amer. L'altitude de chacun de ces pixels est comparée avec l'altitude correspondante du rayon émis depuis le capteur vers l'amer : si la valeur ΔZ_i est inférieure à deux fois la valeur de l'écart type de l'altitude d'un pixel i (par sécurité), alors l'amer est supposé occulté ; sinon, l'amer est supposé visible, et la distance entre le capteur et l'amer est déterminée. Notons que cette procédure n'est applicable que pour des positions du capteur pour lesquelles le modèle est représenté par un bitmap (le MNT ou celui qui est issu de la fusion des images classifiées). Un résultat de la procédure est présenté figure 6.6.

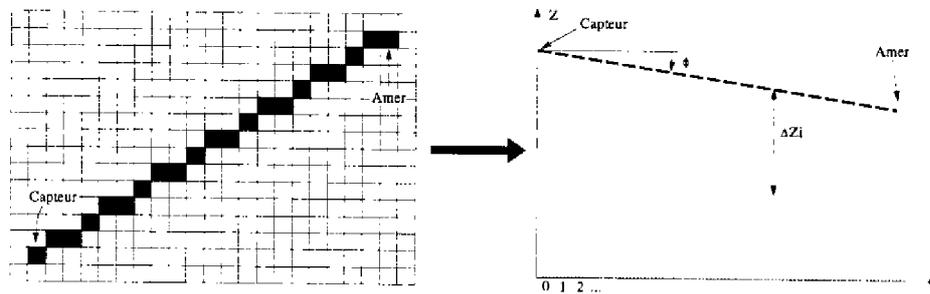


Figure 6.5 : Test de visibilité d'un amer

Cette procédure est bien sûr loin de prendre en compte tous les aspects liés à la probabilité de détection d'un amer que nous avons mentionnés figure 6.1, mais elle suffit à notre problème, où il faut estimer rapidement la faisabilité et le gain d'une tâche de recalage.

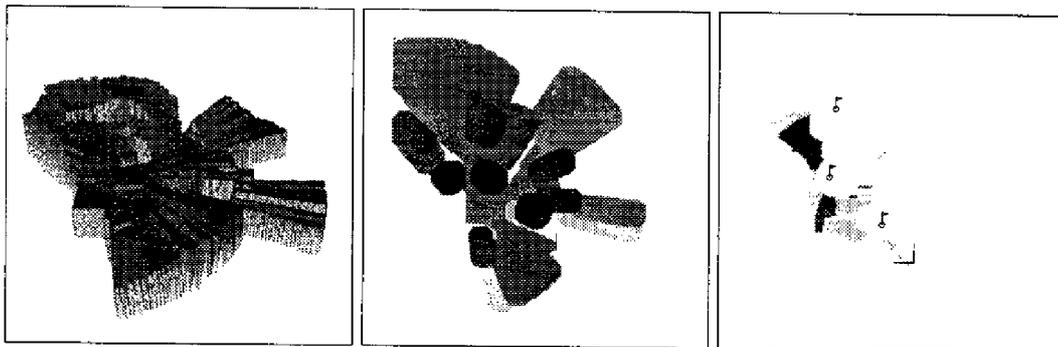


Figure 6.6 : Un modèle de terrain avec trois amers. De gauche à droite : altitudes, zones planes après grossissement, et zones de visibilité d'un amer (gris clair) et de deux amers (gris foncé) dans les régions étiquetées planes

6.3 Évaluation d'une tâche de classification

Par rapport à la planification d'une tâche de recalage, où l'évaluation du résultat d'une tâche prédite est triviale dès lors que l'on connaît un modèle de la fonction de recalage, planifier une tâche de modélisation *pour la navigation* est plus difficile. En effet, nous allons voir que l'on ne peut prédire qu'une estimation de la qualité d'une tâche de classification, et non son résultat, et qu'il est par conséquent délicat d'estimer son utilité dans le contexte de la tâche de navigation.

6.3.1 Modélisation de la procédure de classification

Le modèle de spécificité de notre fonction de classification découle de sa définition : elle perçoit la nature géométrique du terrain. Son modèle de détectabilité dépend quant à lui d'une part du modèle de détectabilité du capteur physique utilisé (qui permet de déterminer les conditions sous lesquelles des points 3D sont effectivement perçus), mais aussi de la procédure de classification. En effet, une zone de terrain plane a par exemple plus de chances d'être effectivement étiquetée plane si elle est proche du capteur que si elle en est loin. Il apparaît donc que le modèle de détectabilité de la classification est lié à son modèle d'incertitude, et nous parlons plutôt de "modèle d'erreur".

La figure 6.7 présente la moyenne des erreurs effectuées sur la classification bayésienne d'une dizaine d'images stéréoscopiques. Les erreurs augmentent en moyenne avec la distance des cellules au capteur, mais dépendent aussi de la classe décidée pour les cellules : les cellules étiquetées "obstacle" le sont avec une probabilité d'erreur plus grande que les cellules accidentées, qui sont elle-mêmes étiquetées avec une probabilité d'erreur plus grande que les cellules planes. Ce comportement de la probabilité d'erreur en fonction de la distance des cellules au capteur et de leur classe n'est bien entendu pas surprenant, puisque nous avons défini une loi de décision qui favorise les fausses alarmes afin d'éviter les non-détections ; de plus, la taille des cellules augmente avec leur distance au capteur, ce qui les rend susceptibles de correspondre à plusieurs classes de terrain. Remarquons que ces courbes justifient le modèle de confiance que nous avons adopté pour la procédure de classification des images laser (figure 6.8) : la notion de confiance est duale de celle de probabilité d'erreur.

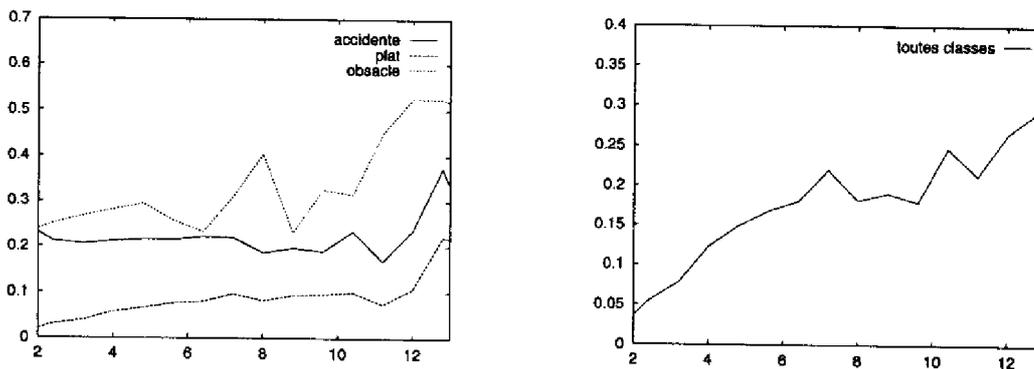


Figure 6.7 : Probabilité d'erreur sur l'étiquetage des images stéréo en fonction de la distance au capteur

Remarque : en toute rigueur, ce modèle dépend des modalités de fonctionnement du capteur

(particulièrement résolution angulaire en ce qui nous concerne) : on devrait écrire $P(\text{erreur} \mid d_c, u)$, où u représente l'état du capteur. Il nous aurait fallu pour cela faire un apprentissage bien plus conséquent, en observant tous les prototypes avec toutes les combinaisons de modalités possibles.

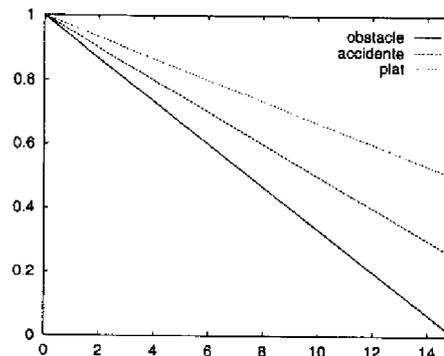


Figure 6.8 : Définition du modèle de confiance de l'étiquetage des images laser

6.3.2 Évaluation du résultat d'une tâche prédite

Ces modèles permettent d'estimer le gain d'information espéré par l'activation d'une procédure de classification en une position donnée : la comparaison de l'erreur (ou confiance) sur l'étiquetage de chaque pixel visible avec celle donnée par le modèle en donne une estimation. Une méthode tout à fait analogue est présentée dans [Elfes 91, Elfes 92].

Afin de limiter le temps de calcul, cette comparaison n'est pas effectuée sur l'ensemble des pixels couvert par le champ de vue du capteur, mais sur une discrétisation radiale et polaire (figure 6.9). Le gain d'information est estimé pour chaque "segment radial", et le secteur angulaire correspondant au champ de vue du capteur qui maximise la somme de ces gains est déterminé. La figure 6.10 montre les trois tâches de classification déterminées en trois positions différentes qui apportent le plus d'information.

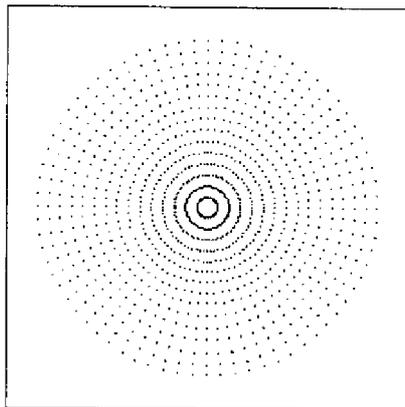


Figure 6.9 : Discrétisation choisie pour évaluer une tâche de classification

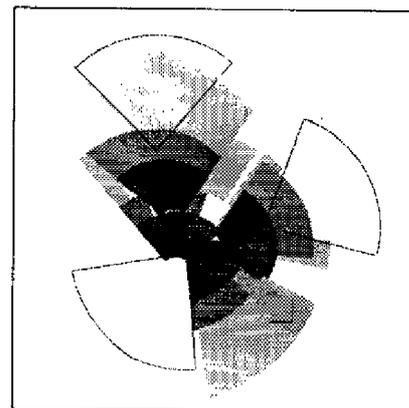


Figure 6.10 : Les trois "meilleures" tâches de classification déterminées en 3 endroits différents

Dans la figure 6.10, les niveaux de gris représentent la confiance sur l'étiquetage, chaque secteur correspondant à une tâche de perception couvre le "plus de blanc possible".

Il est donc aisé de déterminer la tâche de classification qui apporte le plus d'informations, mais cela a peu d'intérêt dans le contexte de la navigation, où il faut rallier un but global déterminé. Le problème n'est alors plus de déterminer la tâche qui apporte le plus d'informations dans l'absolu, mais le plus d'informations *utiles pour rallier le but*. Ainsi, la classification qui maximise le gain d'informations montrée dans la figure 6.11 présente à priori très peu d'intérêt s'il s'agit de rallier le but qui est en haut à droite : choisir de percevoir dans la direction opposée aurait été bien plus pertinent.

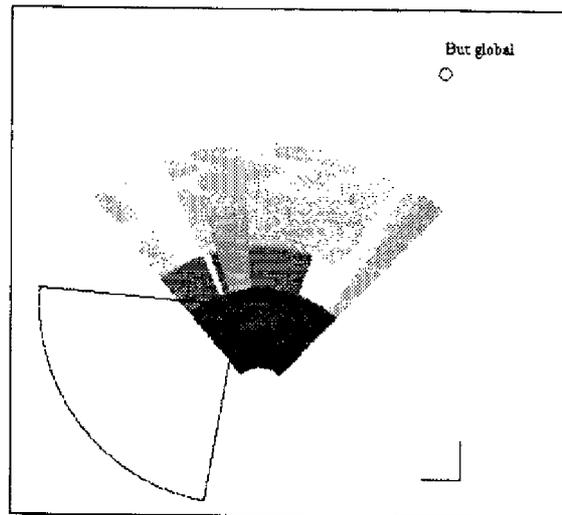


Figure 6.11 : Une classification qui maximise le gain d'information mais est de bien peu d'intérêt pour rallier le but fixé

C'est le problème d'interdépendance entre les actions de perception et de déplacement qui nous gêne ici : pour évaluer l'intérêt d'une tâche de classification, il nous faudrait être capable d'estimer le coût du chemin qu'elle permettrait d'exécuter jusqu'au but global. Or, la seule information que nous puissions prédire grâce au modèle de la fonction de classification est le gain d'information que l'on va vraisemblablement obtenir en percevant une région, et non ce que l'on va effectivement percevoir (nous ne disposons pas d'un modèle "boule de cristal"). Notons que l'on pourrait toutefois évaluer l'intérêt d'une tâche de perception en déterminant le coût du chemin qui joint la zone qu'elle modélise au but global.

6.4 Détermination d'un chemin

La détermination d'un chemin est bien évidemment un des problèmes les plus souvent abordé de la robotique mobile. Nous n'avons pas la prétention de faire ici une revue des nombreuses méthodes existantes, mais avant de présenter la technique que nous avons retenue (paragraphe 6.6), nous nous intéressons aux approches qui traitent des particularités de notre contexte applicatif :

- L'environnement n'est pas décrit en termes d'obstacles et de zones libres (description binaire), mais par plusieurs classes de terrain, qui présentent des caractéristiques différentes pour les déplacements (en particulier au niveau du coût) ;
- Le modèle d'environnement dont nous disposons est *partiel* et *incertain* : en particulier, le but global que l'on cherche à atteindre n'est pratiquement jamais inclus dans une zone modélisée (sauf en fin de mission), et le type des régions étiquetées est connu avec une incertitude ;
- Enfin, la nature géométrique et physique des environnements extérieurs rend difficile l'exécution fiable de trajectoires planifiées : la dérive en position du robot lorsqu'il se déplace est bien plus importante que pour des robots se déplaçant sur un sol plan et stable.

6.4.1 Dans des environnements "valués"

Le problème lié à la planification de chemin de coût minimum à travers des régions auxquelles sont associés différents coûts est très bien résolu lorsque l'on dispose d'un graphe. Il s'agit en effet d'une application pour laquelle les techniques classiques de recherche de chemin dans un graphe valué sont particulièrement bien adaptées. Mais pour que ces techniques retournent un chemin optimal, il est nécessaire que le graphe ait une "résolution" précise, et donc un très grand nombre de nœuds (comme une grille cartésienne très fine par exemple). Leur complexité algorithmique les rend alors coûteuses d'utilisation.

Des auteurs se sont intéressés à la résolution de ce problème lorsque l'environnement est décrit par une partition polygonale de régions. Dans le cas où seule une partition franchissable / infranchissable est considérée, un chemin peut être déterminé à l'aide d'un graphe de visibilité des sommets des polygones [Nilsson 69, Lozano-Perez 79, Moutarlier 91] (après grossissement des polygones obstacles) : cette méthode produit des chemins optimaux pour des robots circulaires dont le rayon de giration n'est pas borné ; elle a été améliorée pour permettre de considérer des modèles de robot plus réalistes. On parle du "problème des régions pondérées" [Mitchell 87] lorsque plus de deux coûts sont associés à la traversée des divers polygones : il existe alors des techniques basées sur une analogie avec les lois optiques de réfraction dans des milieux d'indices différents (loi de Descartes, issue du principe de Fermat) qui déterminent des chemins optimaux pour un robot point [Richbourg 86, Richbourg 87]. On trouvera notamment dans [Mitchell 88] et [Mitchell 91] une étude précise de ces techniques, pour lesquelles des adaptations de la loi de Descartes sont nécessaires : ainsi l'angle limite de réfraction d'un faisceau lumineux n'a pas de sens dans le cas d'un robot, et cette loi donne un chemin *localement* optimal (elle ne garantit pas un chemin globalement optimal lors de la traversée de plus de deux régions). Les algorithmes résultants ne sont pas simples, et de nombreux cas particuliers doivent être considérés.

6.4.2 Dans des environnements partiellement connus

Lorsque le coût est affecté aux régions de l'environnement avec une incertitude, la recherche de chemin est généralement effectuée en pondérant la métrique cartésienne par le degré

d'incertitude que l'on a sur l'environnement : ainsi dans [Timcenko 94] les obstacles sont représentés dans un bitmap par une probabilité d'occupation, et le chemin ralliant la position courante au but, déterminé grâce à l'algorithme A^* , est celui qui minimise le produit $d^* f(i)$, où d est la distance et $f(i) = 1/\sqrt{1-i^2}$, i étant la probabilité d'occupation. Une technique analogue est présentée dans [Oriolo 94].

A. Stentz [Stentz 94a] propose l'algorithme D^* , inspiré de A^* , qui a le grand avantage d'être extrêmement rapide lors de sa ré-exécution après qu'un supplément d'information ait été introduit dans l'environnement (grâce à une tâche de perception par exemple). D^* travaille sur une grille pour chaque élément de laquelle il maintient une estimation du coût pour rallier le but, ainsi qu'un pointeur sur la cellule suivante, qui représente la direction optimale à suivre. Après chaque nouvelle information nouvellement introduite, l'état des cellules environnantes est modifié *localement*, et un nouveau chemin optimal est déterminé en suivant les nouvelles directions optimales ainsi modifiées.

On trouve une autre application intéressante dans [Hu 91] : un robot devant effectuer de nombreux déplacements dispose d'une connaissance topologique initiale de l'environnement (un graphe défini par des lieux accessibles et des obstacles certains), qui ne varie pas. Il connaît aussi la probabilité que des zones libres soient occupées par un obstacle. Un chemin global est planifié dans le graphe en considérant ces probabilités, et au fur et à mesure de l'exécution de ce chemin, le robot met à jour cette connaissance probabiliste grâce à une procédure bayésienne, suivant qu'il découvre qu'une zone est effectivement obstruée ou non. Le robot affine ainsi la connaissance qu'il a de son environnement, et modifie les chemins planifiés en fonction des exécutions précédentes.

6.4.3 Avec des incertitudes sur le contrôle du véhicule

Ce n'est que récemment que le problème de la planification de trajectoires robustes par rapport aux incertitudes sur le déplacement d'un robot a été abordé [Takeda 94, Lazanas 92, Fualdes 95, Alami 94, Timcenko 94, Xu 94, Bouilly 95]. La dérive probable du véhicule est estimée en chaque point de la trajectoire pendant la phase de planification, ce qui a pour effet de réduire le domaine d'accessibilité du véhicule. L'aspect intéressant est la prise en compte de la possibilité de réduire l'incertitude la position du robot grâce à une tâche de recalage : le plan résultant contient alors la trajectoire à exécuter, ainsi que les tâches de recalage le long de cette trajectoire.

Ces travaux considèrent pour la plupart que l'environnement et les amers sont connus, mais il s'agit de véritables contributions à la planification *intégrée* de déplacement et de perception.

6.5 De la difficulté d'une approche exhaustive de la navigation

Considérons les configurations représentées dans la figure 6.12 : dans le premier cas (à gauche), le robot a tout intérêt à se déplacer vers le point "A" et à percevoir la région qui sépare les deux zones planes : par rapport à une stratégie qui consisterait à percevoir dans

la direction du but, cela produirait peut-être un chemin plus long, mais qui a beaucoup de chances d'être réalisable. Les second cas (à droite) illustre une configuration plus particulière : depuis sa position, le robot ne peut percevoir la région "X", parce qu'elle est inclinée de telle sorte qu'aucune réflexion n'est perçue par la télémètre par exemple². Par contre, cette région peut-être perçue depuis le point "A", située à l'extrémité d'un plateau entouré d'un dénivelé infranchissable : une solution idéale consisterait à se déplacer au point "A" pour percevoir la région "X", et éventuellement revenir à la position de départ si cette région s'avère traversable.

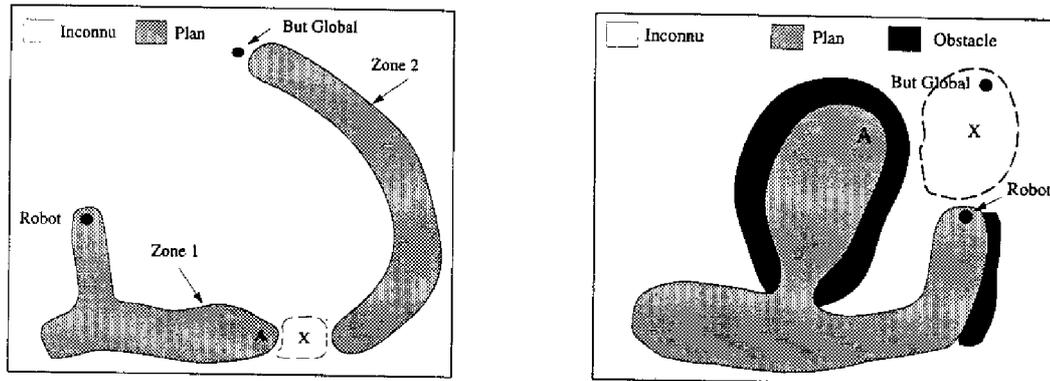


Figure 6.12 : Dans ces deux configurations particulières, la meilleure décision est de se rendre au point "A" pour percevoir la région "X"

Ces deux configurations particulières montrent que la principale difficulté de la fonction navigation est que les tâches de perception et de déplacement sont fortement interdépendantes : il est nécessaire de modéliser l'environnement pour pouvoir se déplacer, et inversement il est souvent nécessaire de se déplacer pour percevoir. On ne peut donc considérer la détermination de ces deux tâches de manière séparée.

Une solution exhaustive consisterait à effectuer une recherche de chemin dans le graphe, au cours de laquelle on déterminerait *pour chaque nœud rencontré* les différentes tâches de perception possibles. La détermination de chacune de ces tâches permettrait d'évaluer un nouvel état de la carte de régions, à partir duquel il faudrait terminer le chemin en cours de construction (figure 6.13).

Le problème d'une telle méthode est qu'il est délicat de chercher un chemin optimal dans la carte de région après sa modification par une tâche de perception prévue : nous avons vu dans le paragraphe 6.3 qu'on ne peut prévoir que la "qualité" d'une tâche de modélisation, et non son résultat. En outre, il est évident que la complexité de cette approche est rédhitoire : même en restreignant le nombre de nœuds pour lesquels on envisage une tâche de perception (aux frontières des zones planes et des zones contraintes par exemple), on aurait à évaluer le coût d'un très grand nombre de chemins différents, et ce d'autant plus si l'on planifie plusieurs tâches de perception au long d'un même chemin (ce qui reviendrait à générer d'autres sous-graphes des sous-graphes de la figure 6.13).

²Ceci n'a rien d'irréaliste, nous avons déjà rencontré une telle situation lors d'une expérimentation

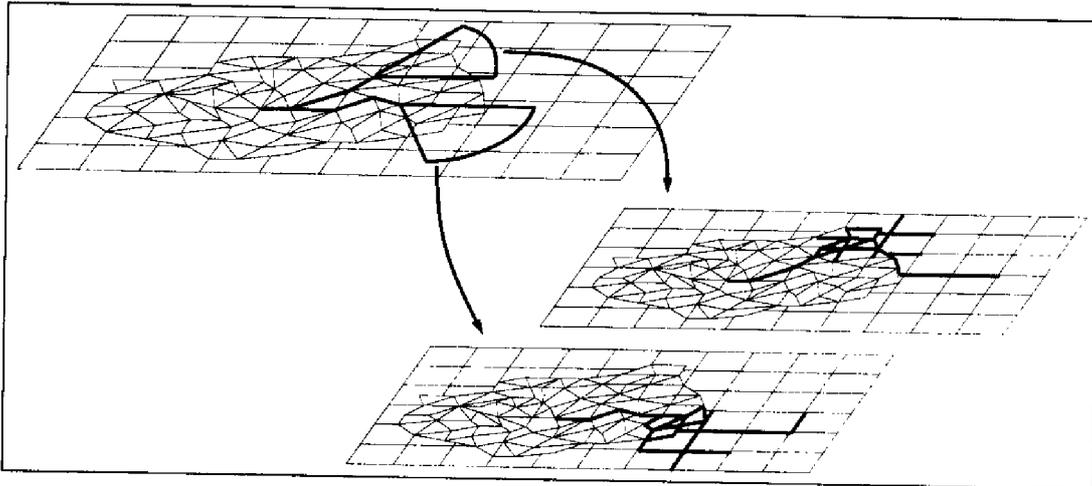


Figure 6.13 : Une approche exhaustive de la navigation : chaque tâche de perception envisagée génère un nouvel état du graphe, dans lequel il faut chercher un chemin optimal

6.6 Une approche heuristique

L'approche complète dont nous avons défini les grandes lignes dans le paragraphe précédent est complexe et coûteuse en temps, et ne satisferait pas les spécifications de rapidité que nous attendons de la fonction navigation. Nous proposons donc ici une approche heuristique bien plus rapide, que nous avons totalement implémentée et dont nous avons pu mesurer l'efficacité aussi bien en simulation que lors de nombreuses expériences réalisées avec le robot Adam.

6.6.1 Principe

Le principe de la méthode que nous proposons ici est de considérer de manière *implicite* les capacités perceptuelles du robot lors de la recherche d'un chemin dans le graphe des régions, grâce à une valuation du graphe qui prend en compte les incertitudes de modélisation. Ainsi l'algorithme de recherche de chemin minimal tolérera de traverser des régions étiquetées obstacle avec une faible confiance : elles peuvent en effet s'avérer traversables, et on sait qu'une tâche de modélisation permettra facilement de le vérifier. Cette approche donnera forcément des résultats moins optimaux que la méthode exhaustive, mais elle s'est avérée satisfaisante, tant des points de vue temps de calcul que de la qualité des résultats.

La principe est illustré par les figures 6.15 et 6.16 (page 103) : dans la figure 6.15, la valuation du graphe ne considère pas l'incertitude sur l'étiquetage des régions, et toutes les zones étiquetées obstacles sont interdites. Le chemin minimal reliant la position de départ au but global fait un retour arrière : la fonction navigation considère que le robot est dans une impasse. Mais ce n'est en réalité pas le cas : si les deux obstacles à l'entrée de l'impasse existent bien réellement (ils ont d'ailleurs été détectés avec une très forte confiance - ils sont rouge vif), il existe des passages possibles au fond de l'impasse : cette zone a été modélisée "obstacle", mais avec une faible confiance (elle a été perçue de loin, et est représentée en

rouge pâle). La figure 6.16 montre le chemin minimal retenu en considérant les incertitudes de modélisation : il traverse le fond de l'impasse, est bien plus court que le précédent, et correspond à une décision bien plus raisonnable. Bien entendu, il est hors de question de planifier une trajectoire à travers cette zone obstacle : seule la portion plane du chemin déterminé sera exécutée ; et l'exécution d'une tâche de perception à son extrémité permettra de lever le doute sur l'étiquetage de la zone obstacle, et ce d'autant mieux qu'elle la percevra de près.

On voit donc que ce principe de résolution rend la détermination de la tâche de perception subordonnée à la détermination d'un chemin minimal, et s'effectue en deux étapes :

1. Recherche du chemin minimal dans le graphe des régions (paragraphe 6.6.3) ;
2. Analyse du chemin minimal trouvé afin de déterminer le sous-but à rallier, le mode de planification à utiliser, et la tâche de perception à y effectuer (paragraphe 6.6.5) ;

6.6.2 Valuation du graphe

L'exemple précédent fait apparaître le caractère fondamental de la valuation du graphe : c'est par ce biais qu'est défini le critère d'optimalité d'un chemin, et la détermination du chemin optimal réalise *toutes* les décisions de la fonction navigation. Nous présentons ici comment est déterminé le coût associé à chaque action. Ce coût est représenté par un vecteur de dimension 2, correspondant aux 2 premiers critères à minimiser définis dans le chapitre 5 (temps et énergie). Nous introduirons le critère concernant l'incertitude sur la position du robot dans le paragraphe 6.6.4.

a) Coûts de déplacement nominaux

Les grandeurs fondamentales sur lesquelles nous nous basons pour déterminer les actions de déplacement sont bien entendu les dimensions géométriques définies dans le graphe des régions : distance cartésienne entre deux nœuds et variation d'altitude correspondant au franchissement d'un nœud.

Le coût en temps d'une tâche de déplacement est proportionnel à la distance parcourue, selon un facteur qui dépend bien sûr de la vitesse de déplacement du robot, mais qui prend aussi en compte le temps nécessaire pour planifier la trajectoire à suivre. C'est la vitesse d'exécution des déplacements, mais surtout le temps de planification qui rend les trajectoires sur terrain accidentées bien plus coûteuses que les trajectoires sur terrain plan.

Pour la détermination du coût en énergie d'une action de déplacement, nous considérons un modèle très simple : l'énergie consommée dépend de la longueur d'une trajectoire et des variations d'altitude *positives* le long de cette trajectoire. Il est pour nous inutile de déterminer un coût plus précis, la représentation de l'environnement que nous utilisons étant essentiellement qualitative (on trouvera un modèle de coût plus réaliste dans [Fagegaltier 94] ou [Gaw 86]). Pour toute action de déplacement, on a donc :

$$C_d = \begin{pmatrix} T_d \\ E_d \end{pmatrix} = \begin{pmatrix} \alpha(C_i)d_{traj} \\ e_1(C_i)d_{traj} + e_2(C_i)\Delta Z_{traj} \end{pmatrix}$$

où $\alpha(C_i)$ représente le temps unitaire de déplacement en fonction de la classe de terrain traversée, et $e_1(C_i)$ et $e_2(C_i)$ représentent les consommations d'énergie unitaires sur terrain plan et pentu.

Le coût nominal de déplacement est une combinaison linéaire de ces deux composantes du vecteur de coût, dont les coefficients sont imposés par la définition de la mission.

Cas des zones inconnues

La quasi-totalité des chemins trouvés traversera une partie du terrain non perçue, ne serait-ce que parce que le but global sera presque toujours dans une telle partie. Il est donc nécessaire d'évaluer le coût de traversée de ces zones : c'est le rôle de la carte de régions initiale. C'est grâce à la définition de ces coûts que la recherche de chemin est guidée vers le but global.

Ces coûts prennent en compte à la fois les coûts de déplacement et de perception, en considérant qu'une tâche de modélisation permet en moyenne d'effectuer une trajectoire d'une longueur pré-estimée, et qu'une tâche de recalage extéroceptive est nécessaire tous les x mètres.

Ainsi le coût de traversée d'une zone inconnue *supposée* plane sera supérieur à celui d'une zone *perçue* comme étant plane, et il en va de même pour les zones supposées accidentées dans la carte téléchargée. Quand aux zones interdites par la description initiale du terrain, elles ne constituent pas de régions (chapitre 4) et ne génèrent donc pas nœuds dans le graphe des régions : la fonction Navigation ne considérera jamais leur traversée.

b) Introduction de l'incertitude sur l'étiquetage des régions

Nous avons vu dans les chapitres précédents qu'à chaque région *perçue* est associée une confiance d'erreur c sur l'étiquetage. Les coûts de base présentés dans le paragraphe précédent sont modifiés par une fonction F dépendant de c , d'une manière différente suivant l'étiquetage de la région correspondante. La forme de ces fonctions de pondération est représentée de manière qualitative sur la figure 6.14, où C_{plat} , C_{acc} , et C_{inc} sont les coûts nominaux de traversée d'une région plane, accidentée, et inconnue.

Remarquons sur cette figure la présence de trois seuils sur la confiance de l'étiquetage S_{obst} , S_{acc} et S_{plat} , au delà desquels l'étiquetage est considéré comme sûr. Notons aussi que lorsque la confiance est nulle (on n'a aucune information sur l'étiquette), le coût de traversée est égal au coût nominal des régions inconnues. Enfin, il peut sembler au vu de cette figure que l'on préférera toujours traverser une zone accidentée étiquetée avec une faible confiance qu'une zone dont on est certain qu'elle est accidentée ; mais traverser une telle zone incertaine nécessite une tâche de modélisation préalable, dont le coût est pris en compte lors de la recherche du chemin minimal (paragraphe 6.6.3).

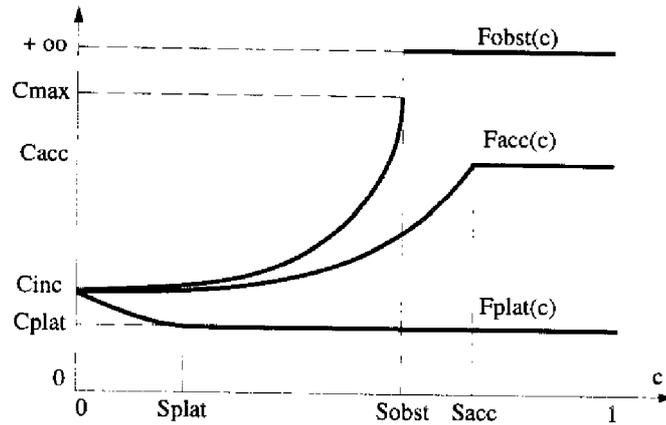


Figure 6.14 : Pondération des coûts nominaux en fonction de la confiance sur l'étiquetage

Cas des régions obstacle et accidentées

C'est une analyse expérimentale de la procédure de classification qui a permis de déterminer les seuils S_{acc} et S_{obst} sur la confiance au delà desquels il est certain que la région est convenablement étiquetée, et en deçà desquels la région peut être plane. Ainsi, le coût de traversée d'une région est égal à son coût nominal (C_{max} dans le cas des régions obstacles, voir figure) lorsque la confiance c est supérieure à ces seuils, et il est inférieur à ce coût nominal lorsque c est inférieure à ces seuils. On a donc :

$$C_{acc}(c) = F_{acc}(c)C_{acc}, \text{ avec } \begin{cases} F_{acc}(c) = 1 & \text{si } c \geq S_{acc} \\ \frac{C_{inc}}{C_{acc}} \leq F_{acc}(c) \leq 1 & \text{si } c < S_{acc} \end{cases}$$

$$C_{obst}(c) = \begin{cases} +\infty & \text{si } c \geq S_{obst} \\ \frac{C_{inc}}{C_{max}} \leq F_{obst}(c) \leq 1 & \text{si } c < S_{obst} \end{cases}$$

Cas des régions planes

Bien que la fonction d'étiquetage ne doive théoriquement pas produire de non-détections, nous introduisons un seuil sur la confiance S_{plat} au delà duquel il est possible qu'une région étiquetée plane ne le soit pas. Ceci conduira à préférer les régions dont on est certain qu'elles sont planes au détriment des régions généralement perçues loin du capteur. On a donc :

$$C_{plat}(c) = F_{plat}(c)C_{plat}, \text{ avec } \begin{cases} F_{plat}(c) = 1 & \text{si } c \geq S_{plat} \\ 1 \leq F_{plat}(c) \leq \frac{C_{inc}}{C_{plat}} & \text{si } c < S_{plat} \end{cases}$$

La définition des trois fonctions de pondération est bien sûr très importante. Nous avons implémenté une fonction de la forme d'une exponentielle croissante pour les pondérations des coûts des régions étiquetées obstacle et accidentée, et une simple pente pour la pondération

des régions planes. D'autres formes de fonction pourraient néanmoins convenir : il y a là un travail d'expertise à effectuer rigoureusement, qui dépend beaucoup du comportement de la fonction de classification (*ie* de son modèle de confiance).

En ce qui nous concerne, ce sont des tests sur des modèles bâtis sur des données réelles, ainsi que des expérimentations qui nous ont aidé à définir ces fonctions. Une telle approche empirique donne des résultats satisfaisants, mais il est clair qu'il est nécessaire de tester un très grand nombre de configurations de terrain pour arriver à un résultat fiable.

6.6.3 Recherche du chemin optimal

Une fois définie la valuation du graphe, la recherche d'un chemin optimal est réalisée par un algorithme A^* , pour lequel l'heuristique de guidage est la distance cartésienne du nœud courant au nœud but, multipliée par le plus faible coût possible (C_{plat}). Les nœuds du graphe existent avant l'exécution de l'algorithme (ils ont été déterminés pendant la phase de structuration du bitmap), les arcs entre les nœuds sont développés en cours de recherche.

Remarque : Notons qu'un algorithme de type A_ϵ donnerait des résultats tout à fait satisfaisants, pour un nombre de nœuds développés plus faible : il peut en effet paraître inutile de déterminer le chemin optimal dans une représentation qui est de nature qualitative (tant en ce qui concerne sa géométrie qu'en ce qui concerne l'étiquetage des régions) : un chemin optimal à ϵ près serait suffisant.

En cours de recherche, une tâche de modélisation et le coût correspondant sont associés au passage d'une région étiquetée de façon certaine à une région étiquetée avec une confiance inférieure aux "seuils de certitude" S . Le coût d'une tâche de classification est essentiellement lié à son temps d'exécution, de l'activation du capteur à la mise à jour du modèle global. En toute rigueur, ce temps est proportionnel au volume de données acquises, et dépend donc de la résolution et du champ de vue du capteur : par souci de simplification, nous considérons que ces modalités de fonctionnement sont fixées. En considérant aussi un coût en terme d'énergie consommée pour la tâche de classification (bien qu'elle soit vraisemblablement négligeable devant l'énergie consommée par un déplacement), nous avons un vecteur de coût :

$$C_{classif} = \begin{pmatrix} T_{classif} \\ E_{classif} \end{pmatrix},$$

où le temps et l'énergie $T_{classif}$ et $E_{classif}$ consommés par la classification sont des valeurs constantes.

Nous considérons au plus *une seule* tâche de modélisation par chemin examiné : le nœud où est planifiée cette tâche de modélisation définira le sous-but à rallier (en ce qui concerne les régions étiquetées "inconnu", nous avons vu que leur coût intégrait le coût des actions de modélisation).

Les figures 6.17 et 6.18 (page 103) présentent le résultat de la recherche de chemin dans deux modèles de terrains construits sur la base d'images lasers acquises sur le site de Geroms.

6.6.4 Introduction du modèle d'erreur de déplacement

Nous avons vu dans le paragraphe 6.2 qu'il est possible d'estimer rapidement le résultat de l'activation d'une fonction de recalage en une position donnée ; par ailleurs, on est capable d'estimer (de majorer) la dérive du robot lorsqu'il se déplace, en fonction du type de terrain sur lequel il évolue.

Il devient alors possible de planifier des chemins en générant *en cours de recherche dans le graphe* une (ou plusieurs) tâches de perception dédiées au recalage. Nous avons implémenté ceci en considérant que le coût d'un chemin est l'intégrale de la courbe de la figure 6.19 (les gains en localisation correspondent à la réalisation d'une tâche de recalage, à laquelle est associée un certain coût forfaitaire similaire à celui de la procédure de classification). Cela revient à introduire une troisième composante au vecteur de coût, mais le coût total à optimiser n'est cette fois plus une combinaison linéaire des composantes du vecteur. Le choix de minimiser cette intégrale et non une combinaison linéaire des trois coûts rend le critère de localisation très important, et consiste à dire que plus la position du robot est incertaine, plus il est risqué de perdre encore de la précision. Nous avons fait ce choix en considérant implicitement un modèle de la fonction de recalage dont la précision dépend fortement de l'incertitude sur la position estimée.

Remarquons qu'en cours de recherche, plusieurs tâches de recalage peuvent être associées à un chemin : toutefois seule la première sera considérée lors de l'exécution du chemin, c'est elle qui définit le sous-but à rallier.

Nous avons implémenté une recherche de chemin qui minimise un tel coût. Elle s'effectue avec une perte de performance supportable (de l'ordre de 50%), due au fait que le facteur de branchement du graphe est augmenté d'une unité (introduction d'un fils supplémentaire pour chaque nœud, qui correspond à la réalisation d'une tâche de recalage), et que l'évaluation d'une tâche de recalage requiert bien entendu plus de calculs que l'évaluation d'un déplacement le long d'un arc. Le chemin résultant est "déviant" vers les amers (figure 6.20).

6.6.5 Analyse du chemin trouvé

Une fois le chemin optimal déterminé, il est analysé de façon à déterminer les éléments suivants :

- Le sous-but : il correspond à la position à laquelle doit s'effectuer la première tâche de perception à exécuter (modélisation ou recalage). Il définit l'objectif qui sera transmis au planificateur de trajectoire. La détermination du sous-but est la réponse aux questions "*où aller ?*" et "*quoi percevoir ?*" posées dans le chapitre 5.
- La tâche de perception à effectuer au sous-but : c'est le choix de modéliser ou de localiser, et la détermination de l'axe de visée. C'est la réponse à la question "*Comment percevoir ?*"
- Le mode de planification à appliquer pour rallier le sous-but : c'est la réponse à la question "*Comment aller ?*"



Figure 6.15 : *Chemin trouvé sans considérer les incertitudes sur la modélisation*

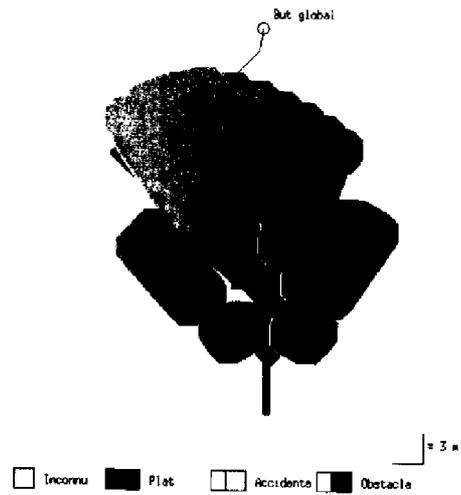


Figure 6.16 : *Chemin trouvé en considérant les incertitudes sur la modélisation*

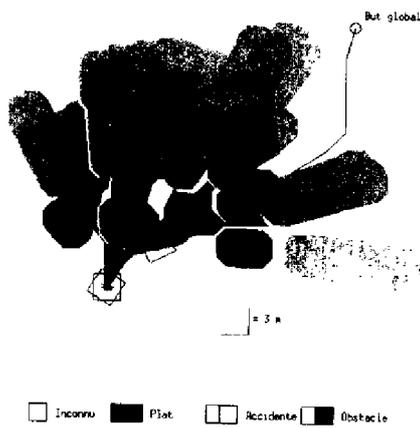


Figure 6.17 : *Traverser la zone étiquetée obstacle est moins coûteux que la contourner*

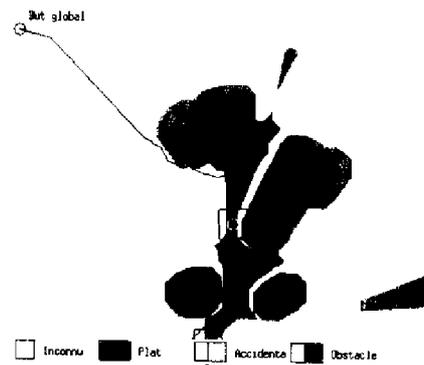


Figure 6.18 : *Traverser la zone accidentée s'avère moins coûteux que la contourner*

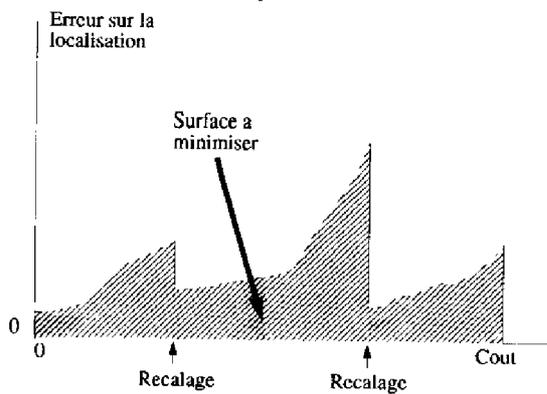


Figure 6.19 : *Introduction de la dérive pour évaluer le coût*

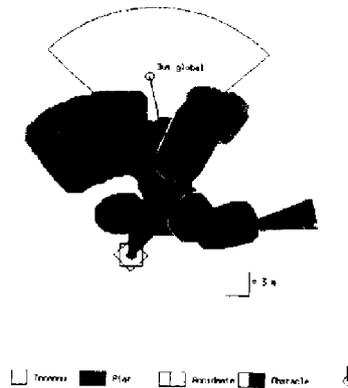


Figure 6.20 : *Le chemin résultant est dévié vers lamer*

La technique utilisée pour déterminer chacun de ces trois éléments est simple : le chemin résultat est scindé en deux chemins : celui qui va du départ au nœud qui définit la tâche de perception, et celui qui va de ce nœud au but global. Si la tâche de perception est une tâche de modélisation, le sous-but est le nœud du premier chemin qui est le plus proche du nœud de la tâche de perception, mais à une distance supérieure à la distance aveugle du capteur. Si la tâche est une tâche de recalage, le sous-but est le nœud correspondant (la distance aveugle est considérée *via* le modèle de détection des amers). La seconde partie du chemin est utilisée pour déterminer les paramètres de la tâche de perception (paragraphe 6.6.5)

a) Contrôle des planificateurs :

Les classes des régions traversées pour aller de la position initiale à ce sous-but définissent le type de planificateur à utiliser. Par souci de simplification, nous considérons qu'à chaque déplacement doit être associé un unique mode de planification : la présence d'une seule région accidentée parmi les régions à traverser provoque donc l'activation du planificateur 3D, et le planificateur 2D n'est sélectionné que si toutes les régions à traverser sont planes.

Une fois le planificateur déterminé, la fonction navigation doit extraire des différentes représentations le modèle correspondant, afin de guider le planificateur le long du chemin (qui a été déterminé selon un critère d'optimalité auquel le planificateur n'a pas forcément accès), et d'optimiser son temps d'exécution (la complexité algorithmique des planificateurs de trajectoires est toujours liée à la taille du modèle utilisé).

Cela est aisé dans le cas du planificateur 2D : les grossissements effectués sur la carte de régions garantissent qu'il existe une trajectoire réalisable à l'intérieur des régions traversées par le chemin. Cependant, le planificateur 2D [Dacre-Wright 93a] prend en compte des contraintes que la navigation ignore : ainsi Adam n'exécute à la vitesse maximum (0.25 m/s !) que les segments de droite dont la longueur est supérieure à 1.20 m. C'est pourquoi le modèle de l'environnement passé au planificateur 2D est défini par le rectangle exinscrit aux régions que la navigation a décidé de traverser (figures 6.21 à 6.24), dans lequel seules les régions étiquetées "plan" sont traversables.

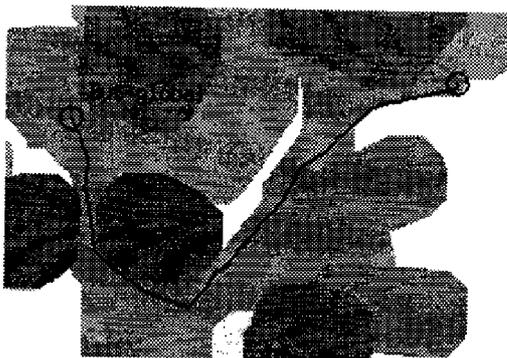


Figure 6.21 : Résultat de la navigation



Figure 6.22 : Les régions correspondantes

C'est par contre plus délicat dans le cas où le planificateur 3D a été sélectionné : rien ne garantit qu'il existe effectivement une trajectoire 3D qui suive fidèlement le chemin planifié. Il faut dans ce cas transmettre au planificateur une portion de la carte topographique qui englobe les régions à traverser, afin de lui laisser une marge de manoeuvre.



Figure 6.23 : Le modèle de terrain effectivement passé au planificateur de trajectoires 2D

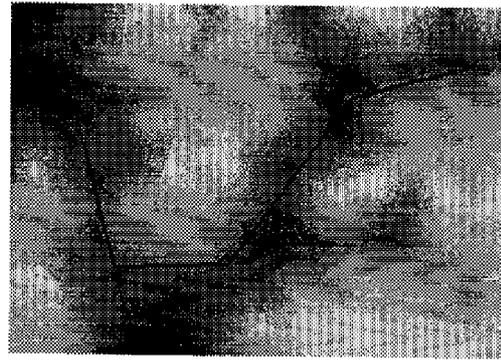


Figure 6.24 : Trajectoire planifiée (en gris : les distances aux obstacles) [Dacre-Wright 93a]

Remarque : Dans le cas où le planificateur 3D ne trouve pas de trajectoire ralliant le sous-but dans les régions sélectionnées, c'est que les informations dont dispose la fonction Navigation ne sont pas "fiables", et il est donc nécessaire de les réviser (il s'agit d'un cas de figure classique dans le cadre de la résolution hiérarchisée d'un problème).

b) Contrôle de la tâche de perception

Le sous-but détermine le besoin perceptuel à remplir (modélisation ou recalage), et la position où il faut exécuter la tâche de perception. Le dernier paramètre important qui définit la tâche de perception est l'angle de visée.

Dans le cas où la tâche de perception retenue est une tâche de recalage, cet axe de visée est défini par la position de (ou des) amer(s) correspondant au meilleur recalage possible à la position considérée. Les éventuelles autres modalités de fonctionnement du capteur (champ de vue, résolution) peuvent être aussi déterminées en fonction du nombre et de la position de ces amers.

Si le sous-but correspond à une tâche de modélisation, l'axe de visée est déterminé en considérant que le chemin qui a été planifié par la fonction Navigation est "pertinent" : les régions à traverser au delà du sous-but ont été sélectionnées car elles sont étiquetées avec une faible confiance, et une tâche de modélisation permettra donc facilement de lever le doute sur leur étiquetage. C'est grâce à la seconde partie du chemin (qui rallie le nœud où a été défini la tâche de modélisation au but global) que l'angle de visée (et le champ de vue s'il est paramétrable) est déterminé. Les régions correspondant à ce chemin situées à une distance visible par le capteur en sont extraites, et la direction de visée est déduite de leur définition : c'est la direction qui maximise la couverture de ces régions par le champ de vue du capteur.

6.6.6 Détection d'échecs

La recherche de chemin détermine un résultat qui optimise une combinaison de critères, mais ce résultat doit aussi satisfaire chacune des contraintes définies dans le chapitre 5.

- La sécurité du robot et l'obligation de rester au sein de l'itinéraire courant sont intrinsèquement pris en compte dans la détermination du graphe à partir de la carte de régions : les régions étiquetées obstacles de façon certaine ou interdites par la description de l'itinéraire ne génèrent pas de nœuds, et on ne considérera donc jamais leur traversée ;
- Les contraintes portant sur le temps et l'énergie sont considérées en cours de recherche : tout chemin qui ne les respecte pas n'est pas considéré. Si aucun chemin ralliant le but ne les respecte, aucun chemin n'est retourné : la fonction navigation est en situation d'échec. C'est alors à un niveau de décision supérieur d'analyser les raisons de cet échec, et de décider des moyens de le résoudre (en décidant de relâcher des contraintes ou en modifiant les critères à minimiser par exemple). Le fonction navigation est alors réactivée avec ces nouvelles données ;
- En ce qui concerne la contrainte qui définit un seuil à ne pas dépasser sur l'incertitude de la position du robot, elle est testée en cours de recherche : un coût infini est associé à tout déplacement qui aurait pour effet de dépasser ce seuil, et conduit par conséquent l'algorithme de recherche à effectuer un retour arrière, afin de reprendre un autre chemin. Si l'algorithme de recherche ne retourne pas de chemin satisfaisant cette contrainte, la fonction navigation retourne un échec.



Chapitre 7

Résultats expérimentaux

L'ensemble des travaux présentés dans ce manuscrit ont été développés et implémentés avec le souci constant d'être validés par des expérimentations en site réel. Nous considérons en effet que les enseignements tirés de situations réalistes sont d'une grande richesse et permettent d'améliorer considérablement la robustesse et la fiabilité des méthodes : de telles situations sont fréquemment imprévisibles, et font apparaître de nombreux problèmes qui n'ont pas été envisagés lors de la conception des logiciels. Les travaux ont donc été intégrés dans le cadre du projet interne EDEN [Chatila 93a, Lacroix 94a, Lacroix 95a] ; ainsi une première version des logiciels a été testée au printemps 1993 sur le site d'expérimentation du LAAS : elle a permis à Adam d'effectuer de manière autonome des déplacements de l'ordre d'une trentaine de mètres. Depuis l'automne 1994, une nouvelle version des logiciels a été implémentée et expérimentée de très nombreuses fois sur le site Geroms, où Adam a pu construire et gérer une carte de l'environnement de plusieurs milliers de mètres carrés et effectuer des missions de plusieurs dizaines de mètres, en se sortant notamment d'impasses.

Ce chapitre présente l'intégration de nos travaux au sein du projet EDEN, ainsi que les résultats de quelques expérimentations. Il s'achève par la présentation d'un logiciel de simulation qui a permis d'évaluer le comportement de notre méthode à long terme.

7.1 Le projet EDEN

7.1.1 Présentation générale

En participant au sein du groupement RISP au projet VAP, le LAAS a commencé à définir et étudier différents aspects liés à la robotique d'exploration planétaire. L'arrivée fin 1992 du robot Adam dans les locaux du laboratoire nous a permis de définir le projet EDEN, dont le but est d'implémenter et d'intégrer ces différents aspects, et d'expérimenter la réalisation autonome de la tâche de navigation "canonique". Hormis les travaux qui font l'objet de ce mémoire, les thèmes qui ont été abordés et implémentés dans le cadre d'EDEN sont les suivants :

- Traitement d'image : stéréovision par corrélation [Lasserre 95], segmentation d'images couleur et identification des régions segmentées ;
- Modélisation de l'environnement sur la base de données tridimensionnelles : construction incrémentale d'une carte d'élévation [Nashashibi 93], extraction et modélisation d'objets d'une scène naturelle (rochers) à l'aide d'ellipsoïdes ou de super-quadriques [Betge-Brezetz 94] ;
- Localisation extéroceptive (recalage) : par corrélation sur la base d'une carte d'élévation [Nashashibi 93], par appariements d'objets [Betge-Brezetz 95] ou de pics extraits d'une représentation d'une carte d'élévation à l'aide de B-splines [Fillatreau 93b] ;
- Planification de trajectoires sur terrain accidenté et sur terrain plan [Dacre Wright 93b];
- Architecture logicielle embarquée : intégration des fonctionnalités embarquées au sein de modules [Fleury 94], communiquant entre eux suivant un protocole client/serveur (CsLib [Chatila 90]) ;
- Structure décisionnelle : gestion des tâches et contrôle de leur activité [Alami 93a] ;
- Mode de déplacement "réflexe" : détection et évitement d'obstacles pendant le mouvement [Milian 94]

Chacun de ces thèmes concerne la partie "robot" de l'architecture globale proposée au LAAS ; mais des travaux qui concernent la station de contrôle et son interaction avec le robot ont aussi pu être menés dans le contexte d'EDEN (télé-programmation [Perret 94] et planification de mission [Chatila 95]).

7.1.2 Architecture fonctionnelle du robot

a) Implémentation du niveau décisionnel : PRS

Pour l'implémentation des composants du niveau décisionnel embarqué sur le robot (interpréteur de tâches, affineur de tâches et superviseur - voir le chapitre 1), nous avons retenu le système de raisonnement procédural PRS [Ingrand 92, Chatila 95]. PRS est un environnement qui permet le développement et l'exécution de *procédures*, qui nous est apparu particulièrement adapté aux contraintes de l'architecture de contrôle d'un robot : la représentation procédurale des différentes tâches et actions à exécuter par le robot est en effet aisée. PRS interprète les tâches spécifiées par la station de contrôle (dans notre cas la tâche "Aller à [But]") en déroulant les procédures associées (figure 7.1 et paragraphe 7.1.3), déclenche l'activation (éventuellement en parallèle) des différentes actions requises en "connectant" entre eux les différents modules du niveau fonctionnel, et surveille le bon déroulement de leur exécution.

Notons que dans l'implémentation actuelle d'EDEN, le processus d'affinement de la tâche de navigation (planification de chemin : choix de sous-but et de tâche de perception) est relégué au niveau fonctionnel.

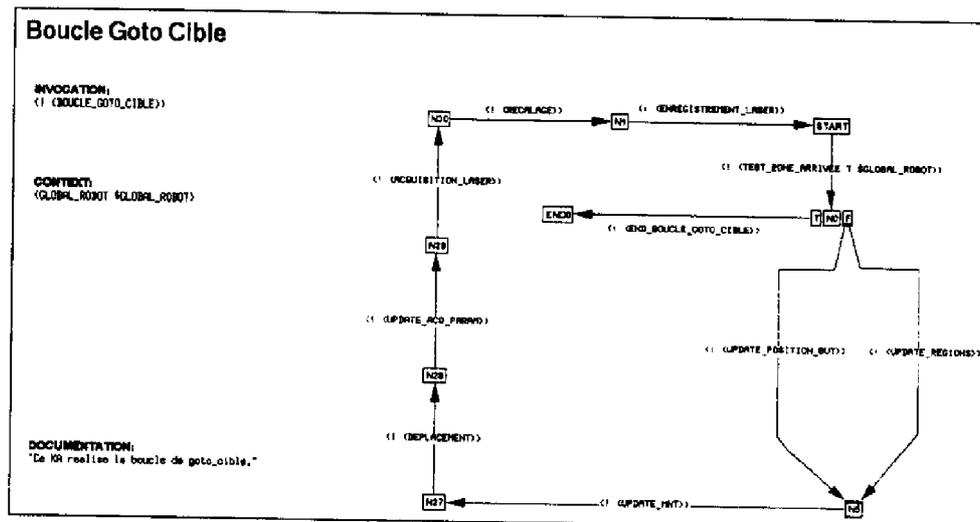


Figure 7.1 : La procédure PRS correspondant à la tâche ‘‘Aller à [But]’’

b) Le niveau fonctionnel

Les différents modules développés dans le cadre d’EDEN sont représentés dans la figure 7.2 : outre le module **Détection de but**, on distingue trois modules de modélisation de l’environnement (**Régions**, **MNT** et **Localisation**) et trois modules de planification (**Navigation**, **Planif 2D** et **Planif 3D**). Notons que les différentes entrées et sorties du module **Navigation** correspondent bien à celles que nous avons spécifiées dans le chapitre 5 (se reporter à la figure 5.2 de la page 82)

7.1.3 Scénario d’expérimentation

Le scénario général d’expérimentation que nous avons défini est une réalisation de la tâche de navigation telle que nous l’avons présentée dans le chapitre 1, avec deux niveaux de planification¹ (chemins et trajectoires) et trois modes de déplacement (réflexe, 2D et 3D).

La procédure d’exécution de la boucle correspondant à la procédure de la figure 7.1 qui permet de réaliser la tâche ‘‘Aller à [But]’’ est la suivante (les modules de la figure 7.2 correspondants à chaque fonction sont indiqués en gras) :

1. Initialisation :

- Spécification au robot des données initiales concernant l’environnement de la mission : frontières de la zone d’évolution et éventuellement zones pour lesquelles une certaine classe de terrain est connue (une cible hexagonale dans l’implémentation actuelle d’EDEN) ;
- Spécification de la mission à réaliser : critères à minimiser, contraintes à respecter, et description du but à atteindre ;

¹Les terrains d’expérimentation sont bien trop petits pour expérimenter le niveau de planification d’itinéraires

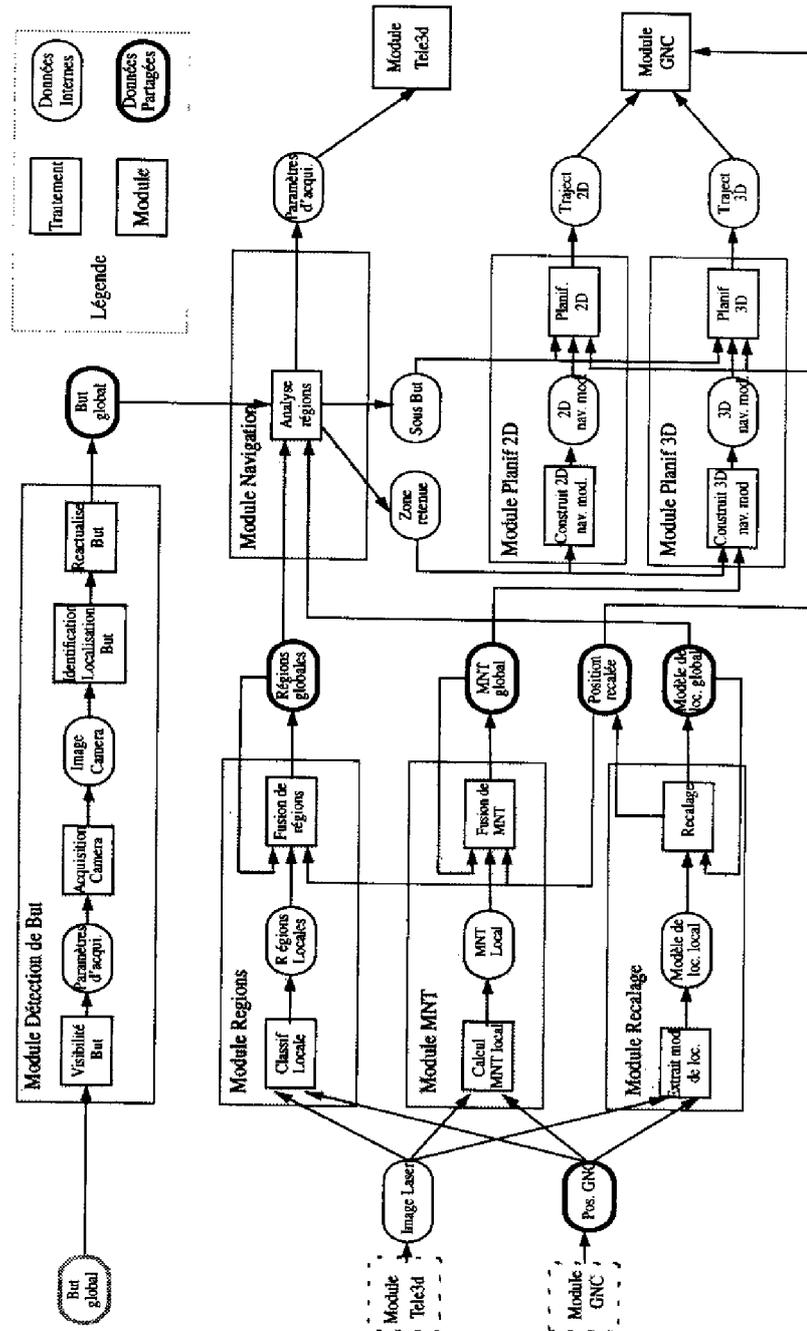


Figure 7.2 : La couche fonctionnelle d'Adam : les flux de données

- Détection et re-localisation éventuelle du but (**Détection de but**) : test de visibilité, acquisition d'une image vidéo, segmentation, recherche du modèle de la cible dans l'image segmentée et rectification éventuelle de la position du but global en cas de succès de la détection ;
- Acquisition de données tridimensionnelles (**Télé-3D**), selon des paramètres fixés par

- défaut ou déterminés par l'activation précédente de la fonction navigation ;
4. Mise à jour du modèle de régions (**Régions**) : classification de l'image acquise, fusion de l'image classifiée dans le modèle de régions global, et structuration du modèle global ;
 5. Si l'activation précédente du modèle navigation l'indique, construction de la carte d'élévation (**MNT**) et recalage de la position du robot (**Recalage**) ;
 6. Planification de chemins (**Navigation**) : recherche du sous-but, détermination du mode de déplacement et de la zone de travail des planificateurs de trajectoire, et détermination des prochains paramètres d'acquisition et des éventuelles représentations de l'environnement à mettre à jour (carte d'élévation et modèle de localisation) ;
 7. Planification de trajectoire (**Planif 2D** ou **Planif 3D**, suivant le résultat de la fonction navigation) ;
 8. Exécution de trajectoire (**GNC**), éventuellement en mode réactif ;
 9. Retour en 2

7.2 Les expériences de navigation

L'expérimentation EDEN n'a pas encore été totalement intégrée : des expérimentations ont toutefois été effectuées avec le planificateur 3D [Nashashibi 94], et le module de navigation a été expérimenté avec le mode de planification 2D : les régions étiquetées accidentées sont alors considérées comme infranchissables. Les logiciels de construction incrémentale de la carte de région et de recherche de sous-but ont pu être testés de très nombreuses fois², et ont montré leur robustesse.

7.2.1 Les terrains d'expérimentation

Les premières expérimentations ont été réalisées au LAAS, sur un terrain de 20x35 mètres dans lequel ont été disposés des rochers, et qui comporte quelques pentes (zones accidentées). D'autres expérimentations ont ensuite été réalisées au CNES, sur le site d'expérimentation Geroms : la grande taille de ce site nous a permis de faire exécuter à Adam des trajectoires résultant de l'enchaînement d'une dizaine de perceptions. Geroms est composé d'une fosse et d'une grande "colline", le terrain présente à la fois des pentes douces (quelques degrés) et raides (jusqu'à plus de 50 degrés d'inclinaison), ainsi que de nombreux obstacles isolés, de taille variable et répartis un peu partout (figure 7.3).

²plusieurs centaines d'images laser ont été classifiées, et depuis l'automne 1994 une vingtaine d'expériences de navigation complètes ont été réalisées sur le site Geroms

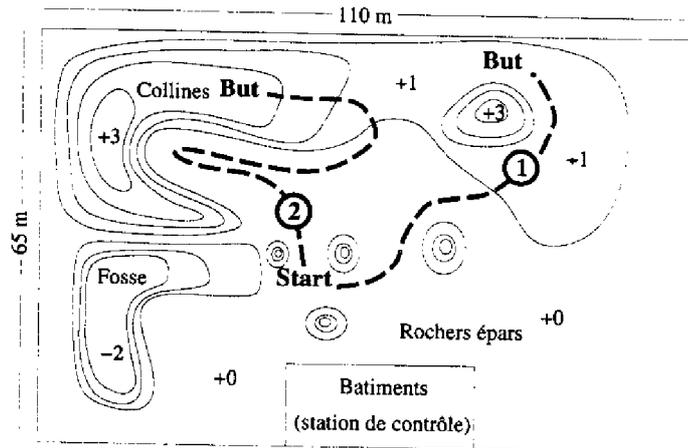


Figure 7.3 : Aperçu de l'allure du site d'expérimentation Geroms (les chiffres indiquent l'élévation du terrain en mètres). Y sont représentées les trajectoires exécutées lors de deux expérimentations présentées dans les pages suivantes

7.2.2 Le robot

La structure mécanique du robot Adam³ a été construite à St Petersburg (Russie) dans les établissements VNII - Transmach : il s'agit d'un châssis à six roues non orientables commandables indépendamment (Adam tourne à la manière d'un véhicule à chenilles) et montées sur des systèmes de suspension passives. Adam pèse près de 800 kg, et avance à la vitesse maximum de 0.28 m/s.

Adam est actuellement prêté au LAAS : il a été acquis par Matra Marconi Space et Framatome lors du projet Euréka AMR. C'est dans le cadre de ce projet qu'Adam a été équipé d'une centrale inertielle 3 axes, et d'un rack VME sur lequel sont montées deux cartes 68030 (fonctionnant avec le système d'exploitation temps réel VxWorks), et qu'ont été développés les logiciels de commande des actionneurs, de contrôle d'exécution des mouvements et de localisation proprioceptive (réalisée par la fusion des informations provenant des codeurs odométriques et d'une centrale inertielle). Ces logiciels ont été "encapsulés" au sein du module GNC dans l'architecture fonctionnelle d'Adam.

Arrivé au LAAS, Adam a été équipé du télémètre laser présenté dans le chapitre 3, de deux bancs stéréoscopiques (l'un orienté vers le bas, dédié à la détection d'obstacles en mouvement, et l'autre dédié aux fonctions de modélisation) et d'un second rack VME sur lequel sont montées deux cartes 68040 et trois cartes Datacube de traitement d'images vidéo.

Seuls les modules GNC, Télé-3D et Détection de but sont compilés sous VxWorks et embarqués à bord du robot : pour des raisons de confort de développement, de possibilités graphiques et de puissance de calcul, tous les autres modules ainsi que PRS sont exécutés sur des stations de travail Unix (Sun SparcStation 10, et Silicon Graphics pour le module Planif 3D) qui communiquent avec Adam par l'intermédiaire d'un câble Ethernet. Notons que les logiciels de communication rendent transparent l'implémentation des modules (Unix

³Advanced Demonstrator for Autonomy and Mobility

ou VxWorks), et que les logiciels tournant sous Unix sont embarquables.

7.2.3 Environnement logiciel

L'expérimentation EDEN est réalisée grâce à trois environnements de développement réalisés au LAAS :

- CsLib : Il s'agit d'un logiciel de communication client/serveur temps-réel qui connecte entre eux les différents modules développés, ainsi que PRS ;
- Calife : les fonctionnalités de traitement des données sensorielles (modélisation, localisation extéroceptive... et en particulier l'ensemble des algorithmes présentés dans ce mémoire) sont développées et exécutées sous Calife, logiciel réalisé au LAAS qui comprend de nombreuses primitives de manipulation de structures de données pour la perception ;
- C-PRS : le contrôle d'exécution des différents modules est implémenté en C-PRS⁴, système de raisonnement procédural.

7.2.4 Quelques résultats

Les figures 7.4 à 7.6 montrent le résultat d'une expérimentation assez simple : le but est très facilement atteint au bout de la sixième perception (il s'agit de la première trajectoire représentée dans la figure 7.3).

⁴C-PRS est un produit d'ACS Technology

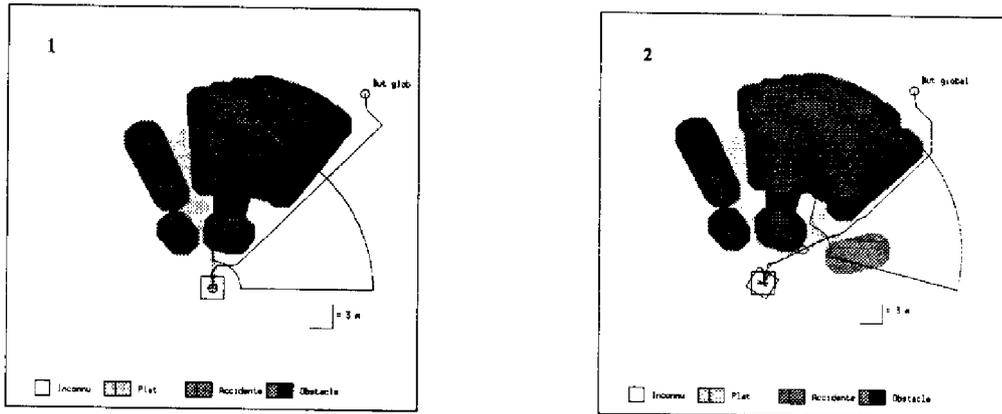


Figure 7.4 : Le robot préfère regarder à droite plutôt que de s'engager entre les deux obstacles...

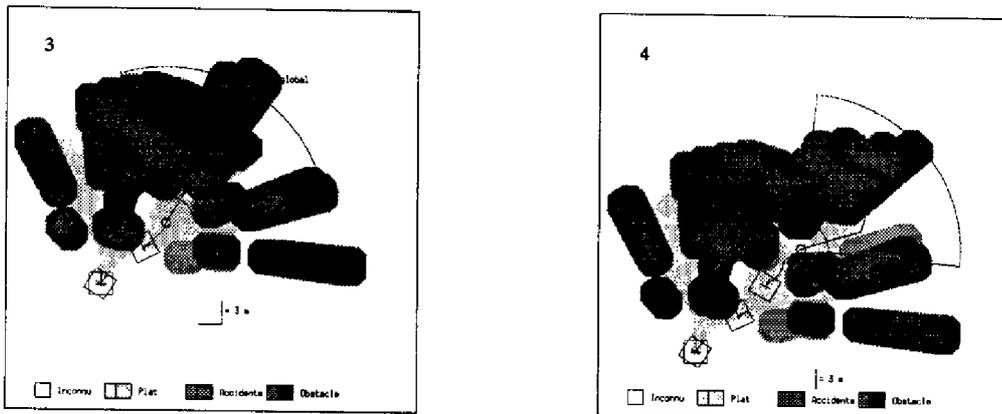


Figure 7.5 : ... ce qui semble être un bon choix...

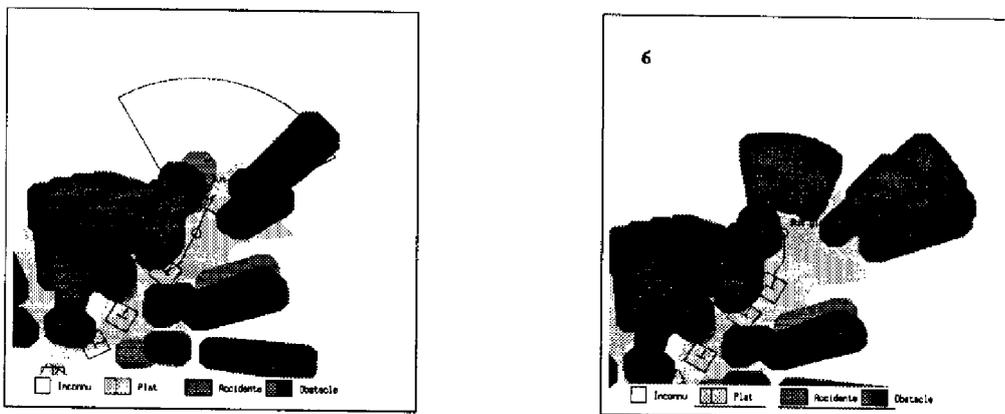


Figure 7.6 : ... le but est atteint après la sixième perception.

Une expérimentation plus intéressante est illustrée par les figures 7.7 à 7.9 (page suivante - il s'agit de la seconde trajectoire représentée figure 7.3) : le but est situé en haut de la colline, et en tentant de le rallier le plus directement possible, le robot se fourvoie dans une impasse de laquelle il ne se sortira qu'après quelques hésitations :

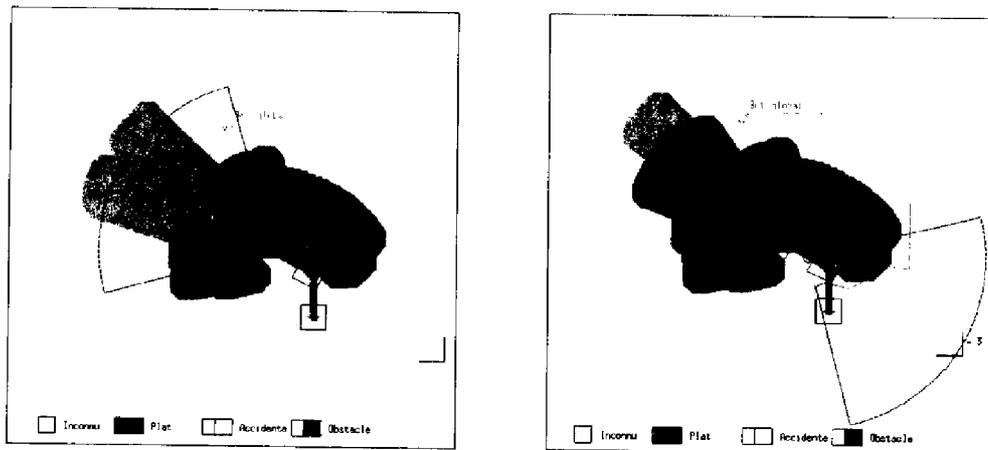


Figure 7.7 : Le robot commence par tenter de contourner les obstacles par la gauche : à la 3ème perception, il décide de faire demi-tour...

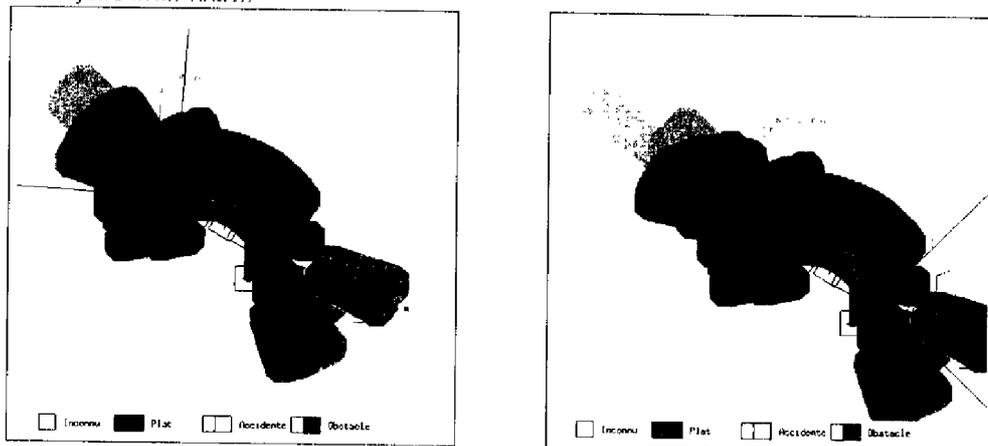


Figure 7.8 : ...mais cette nouvelle voie n'est pas aisée et l'éloigne considérablement du but : après être aller vérifier qu'il est impossible de passer par la gauche, il décide à nouveau de s'engager vers la droite...

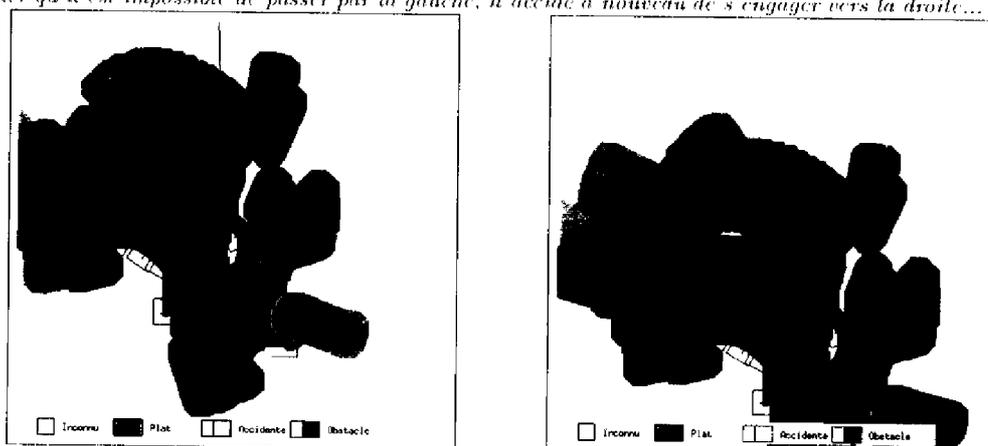
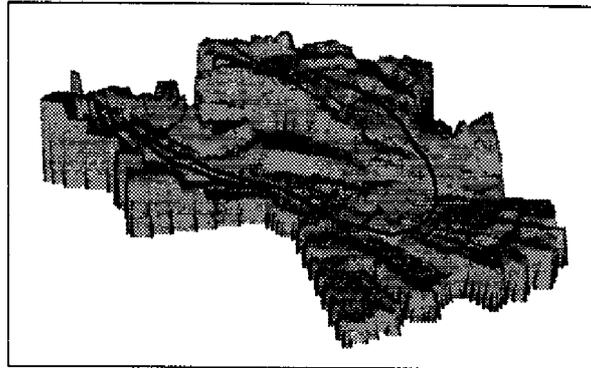


Figure 7.9 : ... et le but final est atteint à la 10ème perception

La figure 7.10 montre quelques caractéristiques du modèle de terrain construit à l'issue de cette expérimentation, ainsi qu'une vue des élévations calculées et de la trajectoire parcourue.

Figure 7.10 : Vue des élévations calculées et caractéristiques du modèle de terrain construit :

- 1400 m² perçus en 10 fois
- 10 positions : 74 m parcourus
- Rectangle ex-inscrit à la zone perçue : 44 par 53 m
- Variations d'altitude : de 0.0 à 3.66 m
- 4700 régions différentes
- Occupation mémoire :
 - régions : 310 Ko
 - bitmap : 3.0 Mo



7.2.5 Performances

Les performances des logiciels de modélisation sont celles que nous avons indiquées dans les chapitres correspondants : chaque image laser est classifiée en moins de 0.2 seconde⁵, et la mise à jour du modèle de régions s'effectue en moins de 5 secondes.

Notre implémentation de l'algorithme A* est capable de développer environ 500 noeuds par seconde (figure 7.11), mais le nombre de noeuds qu'il est nécessaire de développer afin de déterminer le chemin optimal est très variable, suivant la configuration du modèle sur lequel la fonction Navigation travaille. Durant les expérimentations, le temps d'exécution de la fonction Navigation atteint au pire cas une dizaine de secondes.

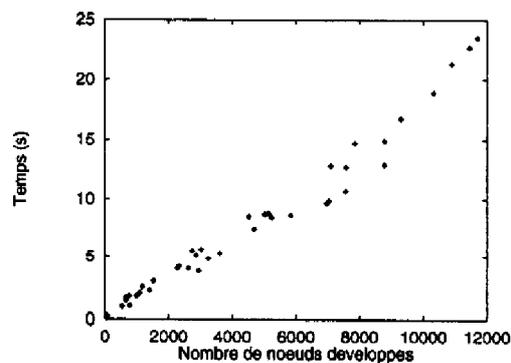


Figure 7.11 : Performances de l'implémentation de l'algorithme A*

⁵Les paramètres d'acquisition des images laser sont les suivants : champ de vue en azimuth et en site respectivement de 90° et de 30°, incréments de balayage de 1° suivant ces deux axes

7.2.6 Conclusion et enseignements

La réalisation d'expérimentations sur un site réel a montré la robustesse de nos algorithmes de construction incrémentale de modèle de terrain et la pertinence des décisions effectuées par la fonction Navigation. Elles ont permis de valider les différents choix que nous avons effectués :

- La procédure de classification des données s'est avérée robuste et très efficace : elle permet à moindre coût de fournir au système une représentation sur laquelle il peut raisonner et prendre des décisions ;
- Le choix de représenter le modèle géométrique précis aux environs du robot (bitmap glissant) s'est avéré judicieux : aucun problème n'est apparu à cause de la discrétisation introduite, et surtout nous avons pu construire des dizaines de modèles sans qu'aucune erreur logicielle n'apparaisse ;
- Enfin, la fonction Navigation a bien rempli son rôle : elle est capable de déterminer en quelques secondes un sous-but à atteindre (ce qui aurait pris beaucoup plus de temps avec un planificateur de trajectoire), et surtout de déterminer la prochaine zone du terrain à percevoir.

Ces constats encourageants nous incitent à persister dans l'approche de la navigation autonome en terrain naturel que nous défendons : il reste en effet de nombreux travaux à réaliser afin d'expérimenter un robot qui présente toutes les capacités qu'une mission d'exploration planétaire exige. Outre l'intégration effective des différents modes de déplacement, que notre méthode de navigation supportera facilement, il reste aussi à gérer l'ensemble des différentes représentations construites par le système, et à faire coopérer différentes méthodes de localisation extéroceptive, et à les expérimenter intensivement. C'est ce dernier point qui nous paraît le plus fondamental : les expérimentations ont montré combien un robot peut facilement se perdre en ne se localisant que sur la base de ses capteurs proprioceptifs.

7.3 Résultats en Simulation

7.3.1 Présentation

Bien que nous disposions d'une plate-forme expérimentale, il nous a paru intéressant de développer un logiciel qui permette de simuler l'acquisition de données tridimensionnelles : cela nous a notamment permis de mettre au point nos algorithmes de gestion de la carte de régions et de navigation, et d'avoir la possibilité de les tester sur une très grande échelle.

a) Les modèles de terrain utilisés

Nous avons utilisé des cartes d'élévation de régions du Monde en guise de modèle de terrain (figure 7.12) : un simple changement d'échelle (différent suivant l'axe vertical et les axes horizontaux) permet de les rendre compatibles avec la taille du robot.

Figure 7.12 : La baie de San Francisco : Carte d'élévation originale

Caractéristiques réelles :

- 1 pixel \leftrightarrow 3 secondes d'arcs (soit environ 92 mètres)
- Surface couverte : 110 par 110 km
- Altitude de 0 à 976 mètres

Caractéristiques simulées :

- 1 pixel \leftrightarrow 0.15 m
- Surface couverte : 180 par 180 mètres
- Altitude de 0 à 5.80 mètres



b) Simulation d'une tâche de perception

Il est aisé de simuler l'acquisition d'une image laser lorsque l'on dispose une description d'un terrain sous la forme d'une carte d'élévation : grâce à une technique de lancer de rayon tout a fait similaire à celle que nous utilisons pour évaluer la visibilité d'un amer (voir figure 6.5 page 90), il est possible de déterminer la distance mesurée par un télémètre.

Il est bien entendu extrêmement difficile de simuler de manière réaliste ce que percevrait un véritable télémètre sur le terrain⁶ : il faudrait connaître le modèle de détectabilité du capteur (conditions pour qu'une mesure soit effectivement retournée : angle de spécularité du faisceau, nature de la surface perçue...), son modèle d'incertitude et son modèle d'erreur (bruit sur la valeur retournée et artefacts, qui dépendent eux aussi de l'angle d'incidence du faisceau laser et de la nature de la surface perçue). Nous avons vu dans le chapitre 3 combien ces modèles sont complexes, mais nous n'avons pas besoin d'une simulation aussi fidèle : d'une part nous ne cherchons pas à valider la procédure de classification des données acquises, qui a pu être testée sur de très nombreuses images réelles, et d'autre part la nature même de cette procédure fait que son application sur les données simulées donne des résultats "réalistes". Il suffit pour cela d'appliquer le modèle de la fonction de confiance de l'étiquetage des cellules (figure 6.8 page 6.8) pour les images classifiées.

7.3.2 Résultats

Les figures 7.13 et 7.14 montrent le modèle construit lors de la réalisation de la tâche de navigation entre les points spécifiés sur la figure 7.12 : le robot a tout d'abord tenté de rallier le but le plus directement possible, et a tenté de passer par le bas (point 1). Arrivé au point 2, il a dû rebrousser chemin (remarquons l'introduction de données initiales - la zone interdite - pour empêcher le robot de sortir de la zone couverte par le modèle de terrain), et a tenté de passer par le point 3, qui s'est aussi avéré être une impasse. Il est donc reparti vers le nord ouest (point 4), et a finalement atteint le but après avoir tenté à plusieurs reprises de couper au plus court (point 5).

⁶Et il est encore plus difficile de simuler l'acquisition d'une image vidéo.

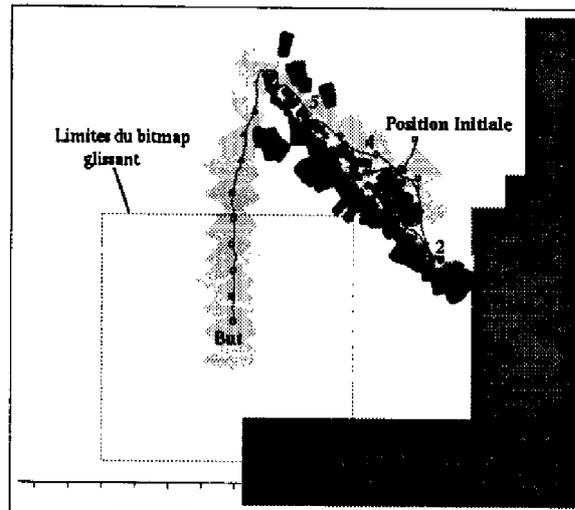
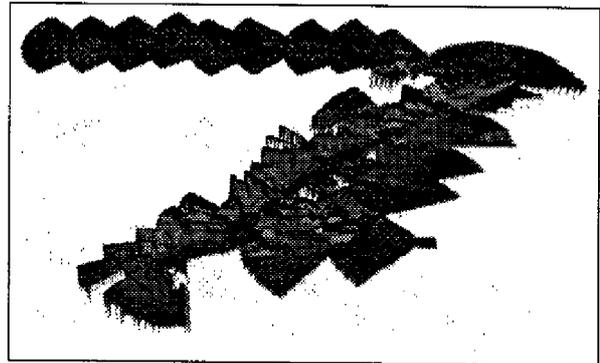


Figure 7.13 : Modèle construit par le robot : le but est ralié après 29 déplacements

Figure 7.14 : Vue des élévations calculées et caractéristiques du modèle de terrain construit en simulation :

- 3200 m² perçus en 29 fois
- 29 positions : 235 m parcourus
- Rectangle ex-inscrit à la zone perçue : 100 par 85 m
- Variations d'altitude : de 0.0 à 3.66 m
- 7750 régions différentes
- Occupation mémoire :
 - régions : 480 Ko
 - bitmap : 3.0 Mo



Chapitre 8

Conclusion générale

Nous avons présenté dans ce mémoire une approche de la navigation d'un robot mobile autonome en environnement naturel. Cette approche fait apparaître le besoin d'une composante décisionnelle essentielle pour doter le robot d'un comportement cohérent et efficace : la *navigation*, qui a la charge de planifier les tâches de déplacement et de perception du robot. Nos travaux ont porté sur le développement et l'expérimentation d'algorithmes dédiés à la navigation, et il en ressort deux principales contributions :

- Une modélisation originale d'un environnement naturel à partir des données tridimensionnelles fournies par les capteurs du robot : une technique de classification probabiliste permet d'analyser rapidement les données et produit une description "sémantique" de la zone perçue, en termes de régions liées à la navigabilité du robot (planes, accidentées, obstacles...). Les données acquises de différents points de vue sont fusionnées en un modèle global, dans lequel figurent les incertitudes résultant des caractéristiques du capteur (niveaux de confiance sur l'étiquetage des régions de terrain), et qui est structuré en un graphe de connexité de régions. Des algorithmes de gestion de ce modèle global permettent de le conserver en mémoire sur l'ensemble de la zone concernant la mission donnée au robot, ce qui présente un grand intérêt ;
- Un système de planification des déplacements du robot qui intègre la planification de ses actions de perception : à notre connaissance, ce problème fondamental a peu été abordé. Nous avons proposé une solution heuristique qui permet de planifier les actions de modélisation et de localisation extéroceptive, tout en déterminant le mode de déplacement à utiliser le mieux adapté. La méthode a été expérimentée de nombreuses fois et a montré un grand intérêt pratique.

Ces travaux ont apporté quelques éléments de résolution aux problèmes soulevés par la navigation autonome en environnement naturel, mais ont aussi mis en évidence le besoin de continuer des recherches dans ce contexte :

- En ce qui concerne la modélisation de l'environnement, nous avons vu que de nombreuses représentations différentes du terrain sont nécessaires pour mener à bien une

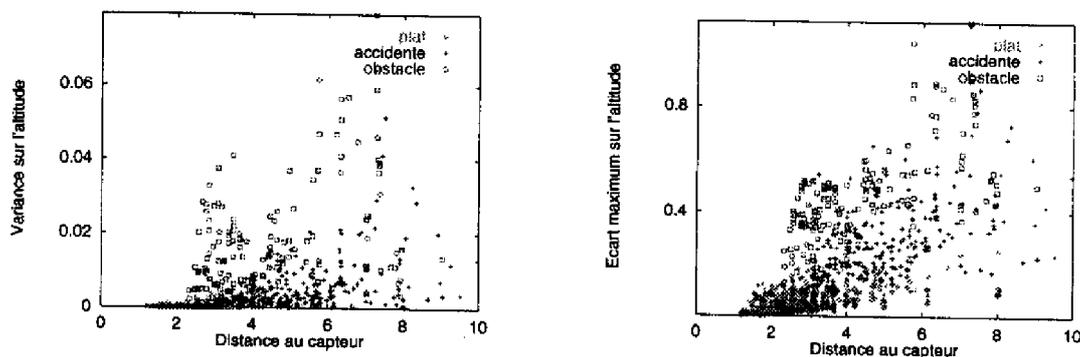
tâche de navigation autonome, et qu'il est par conséquent nécessaire de maintenir la cohérence entre ces différentes représentations. Cela reste un problème ouvert qui a beaucoup d'intérêt : la qualité de chaque représentation serait améliorée, et la détection des incohérences pourrait permettre de contrôler les activités perceptuelles du robot ;

- Les stratégies de perception sont un thème qui reste aussi très ouvert : la prise en compte de modèles probabilistes des capteurs permettrait de planifier les tâches de perception dans le cadre de la théorie de la décision. Le problème a déjà ainsi été abordé dans des environnements intérieurs, où la perception se ramène souvent à l'estimation numérique des attributs de primitives géométriques simples. Dans des environnements naturels qui sont difficilement modélisables à l'aide de primitives géométriques, le problème est plus complexe et mérite des efforts particuliers.

Annexe A

Quelques projections de l'espace des prototypes

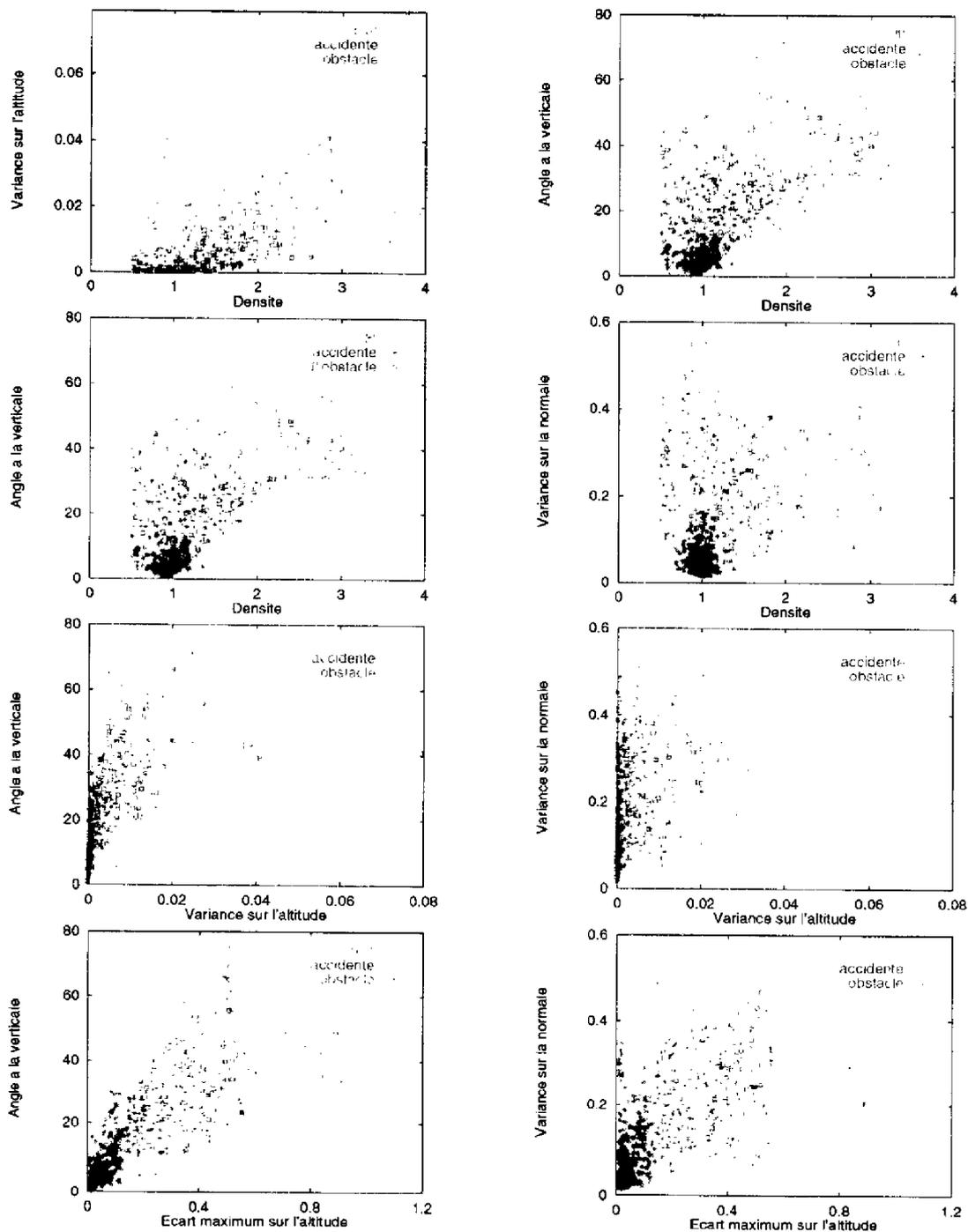
L'espace des prototypes est de dimension 6, et il est par conséquent impossible de le visualiser en une fois. Nous présentons ici quelques projections de cet espace suivant deux attributs (il y en a 15 en tout, nous n'avons retenu que les plus intéressantes). Nous n'avons retenu comme prototypes que les cellules dont la densité relative est supérieure à 0.5 : les autres sont considérées comme non perçues, et seront en fait scindées pendant la procédure de classification.



Ces deux projections montrent clairement un défaut de nos images prototypes : aucune d'entre-elles ne contient de cellules obstacles ou accidentées à une distance inférieure à 2.50 m du capteur. Cela n'est pas trop grave dans notre cas, où *toutes* les paires stéréoscopiques dont on dispose ont cette caractéristique. Remarquons cependant que la variance sur l'altitude croît avec la distance des cellules au capteur (les données sont alors plus bruitées et les cellules couvrent une zone plus grande), de même pour l'écart maximum sur l'altitude (cela est aussi dû à l'augmentation de la surface des cellules avec la distance).

Les projections suivantes montrent que les attributs de densité, de variance et d'écart maximum sur l'altitude, d'angle avec la verticale sont très discriminants pour la classe "plat". Ils le sont par contre beaucoup moins pour les deux autres classes.





En conclusion, il apparaît qu'il existe des corrélations entre les différents attributs de nos prototypes, mais qu'elles sont "floues" et donc délicates à exprimer. Enfin, il est clair que si les attributs sont assez discriminants pour la classe plan, il existe de nombreuses zones de l'espace des prototypes où sont aussi bien présentes des cellules obstacles que des cellules accidentées.

Références bibliographiques

- [Alami 93a] R. Alami & R. Chatila. *Gestion des tâches et Structure Décisionnelle*. Groupement risp, projet vap. rapport laas n. 92030, LAAS - CNRS, 1993.
- [Alami 93b] R. Alami, R. Chatila & B. Espiau. *Designing an Intelligent Control Architecture for Autonomous Robots*. In Third International Symposium on Experimental Robotics, Kyoto, Japan, Oct. 28-30, 1993.
- [Alami 94] R. Alami & T. Simeon. *Planning Robust Motion Strategies for a Mobile Robot*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Alquier 86] M. Alquier. *Navigation et Intelligence Artificielle*. Pour la Science, Juin 1986.
- [Andrews 72] H.C. Andrews. *Mathematical techniques in pattern recognition*. Wiley-Interscience, 1972.
- [Asada 88] M. Asada. *Building a 3D World Model for a Mobile Robot from Sensory Data*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), 1988.
- [Ayache 88] N. Ayache & C. Hansen. *Rectification of images for binocular and trinocular stereovision*. In 9th IEEE International Conference on Pattern Recognition, Rome (Italy), pages 11-16, November 1988.
- [Bajcsy 88] R. Bajcsy. *Active Perception*. Proceedings of the IEEE, vol. 76, no. 8, Aug. 1988.
- [Ballard 94] P. Ballard & F. Vacherand. *The Manhattan Method : A Fast Cartesian Elevation Map Reconstruction from Range Data*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Bares 89] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons & W. Whittaker. *AMBLER: An Autonomous Rover for Planetary Exploration*. IEEE Computer, June 1989. Robotics Institute, Carnegie Mellon University.
- [Basye 89] K. Basye & T. Dean. *Map Learning with Indistinguishable Locations*. In NASA Conference on Space Telerobotics, Pasadena (California), 1989.
- [Betge-Brezetz 94] S. Betge-Brezetz, R. Chatila & M.Devy. *Natural Scene Understanding for Mobile Robot Navigation*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Betge-Brezetz 95] S. Betge-Brezetz, R. Chatila & M.Devy. *Object-Based Modelling and Localization in Natural Environments*. In IEEE International Conference on Robotics and Automation, Nagoya, Japan, 1995.

- [Bouilly 95] B. Bouilly, T. Simon & R. Alami. *A Numerical Technique for Planning Motion Strategies for a Mobile Robot in Presence of Uncertainties*. In To be published IEEE International Conference on Robotics and Automation, Nagoya, (Japan), 1995.
- [Cameron 90] A. Cameron & H. Durrant-White. *A Bayesian Approach to Optimal Sensor Placement*. The International Journal of Robotic Research, vol. 9, no. 5, 1990.
- [Chatila 85] R. Chatila. *Mobile Robot navigation: space modeling and decisional processes*. In Robotics Research : The Third International Symposium, Gouvieux (France), 1985.
- [Chatila 90] R. Chatila & R. Ferraz De Camargo. *Open Architecture Design and Inter-task/Intermodule Communication for an Autonomous Mobile Robot*. In IEEE International Workshop On Intelligent Robots and Systems, Tsuchiura, Japan, July 1990.
- [Chatila 93a] R. Chatila, S. Fleury, M. Herrb, S. Lacroix & C. Proust. *Autonomous Navigation in Natural Environment*. In Third International Symposium on Experimental Robotics, Kyoto, Japan, Oct. 28-30, 1993.
- [Chatila 93b] R. Chatila, B. Dacre Wright, M. Devy, P. Fillatreau, S. Lacroix, F. Nishashibi, P. Pignon & T. Siméon. *Génération de déplacements autonomes*. Groupement risp, projet iares. rapport laas n. 93272, LAAS - CNRS, 1993.
- [Chatila 95] R. Chatila & F. Ingrand. *Mission Planning and Execution Control for Intervention Robots*. In Proceedings of the Indersea Robotics and Intelligent Control, Lisboa, Portugal, 1995.
- [Christensen 94] H.I. Christensen & E. Granum. *Active Vision - A Survey*. In 11th European Conference on Artificial Intelligence (ECAI '94), Amsterdam (Netherlands), 1994.
- [Colly 89] J. Colly, M. Devy, M. Ghallab & L.H. Pampagnin. *Sensory Control for 3D Environment Modeling*. In IARP 1st Workshop on Multi-Sensor Fusion and Environment Modeling, Toulouse (France), October 1989.
- [Costard 92] F. Costard & Al. *A Reference Martian Mission for a Long Range Rover*. In Misions, Technologies and Design of Planetary Mobile Vehicles, CNES, Toulouse, France, Sept. 1992.
- [Coutaud 78] G. Coutaud & L. Andlauer. *Les techniques de la navigation aérienne*. Collection "Que sais-je ?" - Presses Universitaires de France, 1978.
- [Cowan 88] C.K. Cowan. *Model-based Synthesis of Sensor Location*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), pages 900-905, April 1988.
- [Cowan 89] C.K. Cowan & A. Bergman. *Determining the Camera and Light Source Location for a Visual Task*. In IEEE International Conference on Robotics and Automation, Scottsdale, (USA), pages 509-514, April 1989.
- [Dacre-Wright 93a] B. Dacre-Wright. *Planification de Trajectoire pour un Robot Mobile sur Terrain Accidenté*. PhD thesis, Thèse de l'Université Paul Sabatier, Toulouse (France), 1993.
- [Dacre Wright 93b] B. Dacre Wright & T. Siméon. *Free space representation for a mobile robot moving on a rough terrain*. IEEE International Conference on Robotics and Automation, Atlanta (USA), 2-6 Mai 1993, pp.37-43, 1993.
- [Daily 88a] 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*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), 1988.

- [Daily 88b] M.J. Daily, J.G. Harris & K. Reiser. *An Operational Perception System for Cross-Country Navigation*. In IEEE Conference on Computer Vision and Pattern Recognition, Ann Arbor, Michigan (USA), 1988.
- [Degallaix 91] B. Degallaix, R. Alami, R. Chatila & V. Rigaud. *An Architecture for Task Level Programming and Supervision of an Intervention Robot. Application to the NADIA Project*. In 4th International Symposium on Offshore, Robotics and Artificial Intelligence (ORIA), Marseille (France), December 1991.
- [Dunlay 88] R.T. Dunlay. *Obstacle Avoidance Perception Processing for the Autonomous Land Vehicle*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), 1988.
- [Durrant-Whyte 87] H.F. Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Robot Systems*. Kluwer Academic Publ., Boston, MA, 1987.
- [Durrant-Whyte 88] H. F. Durrant-Whyte. *Sensors Models and Multisensors Integration*. International Journal of Robotics Research, vol. 7, pages 97-113, 1988.
- [Elfes 86] A. Elfes. *A Sonar-Based Mapping and Navigation System*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), 1986.
- [Elfes 89] A. Elfes. *Using Occupancy Grids for Mobile Robot Perception and Navigation*. In IEEE Transactions on Computers, 1989.
- [Elfes 91] A. Elfes. *Dynamic Control of Robot Perception Using Stochastic Spatial Models*. In International Workshop on Information Processing in Mobile Robots, March 1991.
- [Elfes 92] A. Elfes. *Dynamic Control of Robot Perception Using Multi-Property Inference Grids*. In IEEE International Conference on Robotics and Automation, Nice, (France), 1992.
- [Fagegaltier 94] L. Fagegaltier. *Planification Multichemin pour les Déplacements d'un Véhicule Tout-Terrain*. PhD thesis, Thèse de l'Université Montpellier II Sciences et Techniques du Languedoc, 1994.
- [Fillatreau 93a] P. Fillatreau & M. Devy. *Localization of an Autonomous Mobile Robot from 3D Depth Images using heterogeneous Features*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '93), Yokohama, , Japan), July 1993.
- [Fillatreau 93b] P. Fillatreau, M. Devy & R. Prajoux. *Modelling of Unstructured Terrain and Feature Extraction using B-spline Surfaces*. In '93 International Conference on Advanced Robotics (ICAR), Tokyo (Japan), November 1993.
- [Fleury 94] S. Fleury, M. Herrb & R. Chatila. *Design of a Modular Architecture for Autonomous Robot*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Foley 90] J. Foley, A. Van Dam, S. Feiner & J. Hughes. *Computer graphics : Principles and practice (2nd edition)*. Addison-Wesley, 1990.
- [Forhan 94] J.M. Forhan. *Apprentissage de l'Environnement d'un Robot Mobile par Classification Bayésienne et Réseau de Neurones*. Rapport technique, Rapport de fin d'études de l'Ecole Supérieure de l'Aéronautique et de l'Espace, 1994.
- [Fualdes 95] T. D. Fualdes & C. J. Barrouil. *Perception Planning for an Unmanned Ground Vehicle*. Robotics and Autonomous Systems, vol. 11, 1995.
- [Garcia 91] F. Garcia & R. Mampey. *Mobile Robot Planning by Reasoning both at Itinerary and Path Levels*. In Fifth International Conference on Advanced Robotics, Pisa, Italy, 1991.

- [Gat 92] E. Gat. *Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots*. In 10th National Conference on Artificial Intelligence (AAAI '88), San Jose, Ca (USA), 1992.
- [Gat 94] 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, no. 4, August 1994.
- [Gaw 86] D. Gaw & A. Meystel. *Minimum-Time Navigation of an Unmanned Mobile Robot in a 2-1/2D World with Obstacles*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), 1986.
- [G.Giralt 92] G.Giralt, R.Chatila & R.Alami. *The remote-operated autonomous-robot concept: from deep sea to outerspace applications*. International Planetary Rovers Symposium, Lab ge (France), 28-30 Septembre 1992, 17p., 1992.
- [Ghallab 89] M. Ghallab & A. Mounir Alaoui. *Managing Efficiently Temporal Relations Through Indexed Spanning Trees*. In 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, Michigan (USA), pages 1297-1303, 1989.
- [Giralt 88] G. Giralt & R. Paul. *An Automatic Task-Planning, Sensor- Controlled Robot-system for Autonomous Underwater Manipulation Tasks*. In 2nd Int. Workshop on Subsea Robotics, Tokyo, Japan, November 1988.
- [Giralt 91] G. Giralt & J.L. Lacombe. *France national developments in mobile robots for hostile or difficult work environments*. International Symposium on Advanced Robot Technology ('91 ISART), Tokyo (Japon), March 1991.
- [Giralt 92] G. Giralt & L. Boissier. *The French Planetary Rover VAP : Concept and Current Developments*. In IEEE International Conference on Intelligent Robots and Systems (IROS '92), Raleigh, NC (USA), 1992.
- [Giralt 93] G. Giralt, R. Chatila & R. Alami. *Remote Intervention, Robot Autonomy, And Teleprogramming: Generic Concepts And Real-World Application Cases*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '93), Yokohama, (Japan), pages 314-320, July 1993.
- [Goto 87] Y. Goto & A. Stensz. *Mobile Robot Navigation : the CMU System*. IEEE Expert, vol. 2, no. 4, Winter 1987.
- [Grandjean 91] P. Grandjean. *Perception Multisensorielle et Interprétation de Scènes*. PhD thesis, Thèse de l'Université Paul Sabatier, Toulouse (France), 1991.
- [Grandjean 93] P. Grandjean & L. Matthies. *Perception Control for Obstacle Detection by a Cross-Country Rover*. In IEEE International Conference on Robotics and Automation, Atlanta, Georgia, 1993.
- [Grimson 86] W.E.L. Grimson. *Disambiguating Sensory Interpretations Using Minimal Sets of Sensory Data*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), 1986.
- [Hager 91] G. Hager & M. Mintz. *Computational Methods for Task-Directed Sensor Data Fusion and Sensor Planning*. The International Journal of Robotic Research, vol. 10, no. 4, 1991.
- [Hebert 86] M. Hebert & Al. *Outdoor Scene Analysis Using Range Data*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), pages 1426,1432, April 1986.
- [Hebert 89a] M. Hebert. *Building and Navigating Maps of Road Scenes Using an Active Sensor*. In IEEE International Conference on Robotics and Automation, Scottsdale, (USA), 1989.

- [Hebert 89b] M. Hebert, C. Caillas, E. Krotkov, I.S. Kweon & T. Kanade. *Terrain Mapping for a Roving Planetary Explorer*. In IEEE International Conference on Robotics and Automation, Scottsdale, (USA), pages 997-1002, 1989.
- [Hebert 91] M. Hebert & E. Krotkov. *3D Measurements from Imaging Laser Radars : How Good are They ?* In IEEE International Workshop on Intelligent Robots and Systems (IROS '91), Osaka (Japan), 1991.
- [Hebert 94] M. Hebert. *Pixel-based Range Processing for Autonomous Driving*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Henderson 85a] T.C. Henderson, C. Hansen & B. Bhanu. *A Framework for Distributed Sensing and Control*. In 9th International Joint Conference on Artificial Intelligence (IJCAI), Los Angeles, California (USA), 1985.
- [Henderson 85b] T.C. Henderson, C. Hansen & B. Bhanu. *The Specification of Distributed Sensing and Control*. Journal of Robotic Systems, vol. 2, no. 4, pages 387 - 396, 1985.
- [Hoffman 91] R. Hoffman & E. Krotkov. *Perception of Rugged Terrains for a Walking Robot : True Confessions and New Directions*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '91), Osaka (Japan), 1991.
- [Hotz 91] B. Hotz. *Etude de techniques de stéréovision par corrélation - Application au programme Véhicule Autonome Planétaire (V.A.P.)*. Rapport de DEA en Robotique et Vision Artificielle, Centre National d'Etudes Spatiales (Toulouse), September 1991.
- [Hu 91] H. Hu, M. Brady & P. Robert. *Coping with Uncertainty in Control and Planning for a Mobile Robot*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '91), Osaka (Japan), 1991.
- [Hutchinson 88] S.A. Hutchinson, R.L. Cromwell & A.C. Kak. *Planning Sensing Strategies in a Robot Work Cell with Multi-Sensor Capabilities*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), pages 1068-1075, April 1988.
- [IJCV 93] IJCV. *Promising Directions in Active Vision*. International Journal of Computer Vision - Special Issue on Active Vision, vol. 11, no. 2, 1993.
- [Ikeuchi 87] K. Ikeuchi & T. Kanade. *Modeling Sensor Performance for Model-Based Vision*. In Robotics Research : The Fourth International Symposium, Santa Cruz (USA), pages 255-263, 1987.
- [Ikeuchi 88a] K. Ikeuchi & T. Kanade. *Applying Sensor Models to Automatic Generation of Object Recognition Programs*. In Second International Conference on Computer Vision, Tampa, Florida (USA), pages 228-237, 1988.
- [Ikeuchi 88b] K. Ikeuchi & T. Kanade. *Automatic Generation of Object Recognition Programs*. Proceedings of the IEEE, vol. 76, no. 8, pages 1016-1035, August 1988.
- [Ingrand 92] F. F. Ingrand, M. P. Georgeff & A. S. Rao. *An Architecture for Real-Time Reasoning and System Control*. IEEE Expert, Knowledge-Based Diagnosis in Process Engineering, vol. 7, no. 6, pages 34-44, December 1992.
- [Krotkov 91] E. Krotkov, J. Bares, T. Kanade, T. Mitchell, R. Simmons & R. Whittaker. *Ambler: A Six-Legged Planetary Rover*. In '91 International Conference on Advanced Robotics (ICAR), Pisa (Italy), pages 717-722, June 1991.
- [Krotkov 92] E. Krotkov & R. Simmons. *Performance of a Six-Legged Planetary Rover : Power, Positioning and Autonomous Walking*. In IEEE International Conference on Robotics and Automation, Nice, (France), 1992.

- [Krotkov 94] E. Krotkov, M. Hebert, M. Buffa, F. Cozman & L. Robert. *Stereo Fringing and Position Estimation for Autonomous Planetary Rovers*. In IARP 2nd Workshop on Robotics in Space, Montreal, Canada, 1994.
- [Kuipers 83] B. J. Kuipers. *Modeling Human Knowledge of Routes : Partial Knowledge and Individual Variation*. In 3rd National Conference on Artificial Intelligence (AAAI '83), Washington (USA), 1983.
- [Kuipers 88] B. J. Kuipers & Y. T. Byun. *A Robust, Qualitative Method for Robot Spatial Learning*. In 7th National Conference on Artificial Intelligence (AAAI '88), Saint Paul (USA), 1988.
- [Kuipers 91] B. Kuipers & Y. T. Byun. *A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations*. Journal of Robotics and Autonomous Systems, vol. 8, no. 1-2, 1991.
- [Kweon 88] I.S. Kweon, M. Hebert & T. Kanade. *Sensor Fusion of Range and Reflectance Data for Outdoor Scene Analysis*. In NASA Workshop on Space Operations, Automation, and Robotics, Dayton, Ohio (USA), pages 373-382, 1988.
- [Kweon 91] I.S. Kweon & T. Kanade. *Extracting Topographic Features for Outdoor Mobile Robots*. In IEEE International Conference on Robotics and Automation, Sacramento, (USA), 1991.
- [Kweon 92] I.S. Kweon & T. Kanade. *High-Resolution Terrain Map from Multiple Sensor Data*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, 1992.
- [Lacombe 91] J. L. Lacombe & T. Blais. *Programme AMR de Robots Mobiles Avancés. Solutions retenues pour le développement des modules de perception et d'interface homme-machine du démonstrateur ADAM*. In 4th International Symposium on Offshore, Robotics and Artificial Intelligence, Marseille (France), 1991.
- [Lacroix 92] S. Lacroix, P. Grandjean & M. Ghallab. *Perception Planning for a Multi-sensor Interpretation Machine*. In IEEE International Conference on Robotics and Automation, Nice, (France), May 1992.
- [Lacroix 94a] S. Lacroix, R. Chatila, S. Fleury, M. Herrb & T. Simeon. *Autonomous Navigation in Outdoor Environment : Adaptive Approach and Experiment*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Lacroix 94b] S. Lacroix, P. Grandjean & M. Ghallab. *Active Scene Interpretation with a Multisensory Perception Machine*. Submitted to the Journal of Artificial Intelligence Research, Décembre 1994.
- [Lacroix 95a] S. Lacroix & R. Chatila. *Motion and Perception Strategies for Outdoor Mobile Robot Navigation in Unknown Environments*. In Submitted to the Fourth International Symposium on Experimental Robotics, Stanford, Ca, USA, 1995.
- [Lacroix 95b] S. Lacroix, T. Siméon & R. Chatila. *Stratégies de Déplacement et de Perception pour la Navigation*. Projet iares - sous système gda. rapport laas n. 94512, LAAS - CNRS, 1995.
- [Langer 94] D. Langer, J. K. Rosenblatt & M. Hebert. *An Integrated System for Autonomous Off-Road Navigation*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Lasserre 95] P. Lasserre & P. Grandjean. *Stereo Vision Improvements*. In submitted to the '95 International Conference on Advanced Robotics, Barcelona, Spain, 1995.

- [Laurette 91] R. Laurette, R. A. de Saint Vincent (Matra-Espace) R. Alami, C. Chatila & V. Pérébaskine (LAAS/CNRS). *Supervision and Control for the AMR Intervention Robot*. In 5th International Conference on Advanced Robotics (ICAR '91), Pisa (Italy), 1991.
- [Lazanas 92] A. Lazanas & J.C. Latombe. *Landmark-Based Robot Navigation*. In 10th National Conference on Artificial Intelligence (AAAI '88), San Jose, Ca (USA), 1992.
- [Leonard 90] J.J. Leonard, H.F. Durrant-White & I.J. Cox. *Dynamic Map Building for an Autonomous Robot*. International Journal of Robotics Research, vol. 11, no. 4, 1990.
- [Levitt 90] T.S. Levitt & D.T. Lawton. *Qualitative Navigation for Mobile Robots*. Artificial Intelligence, vol. 44, no. 3, 1990.
- [Linden 86] T.A. Linden, J.P. Marsh & D.L. Dove. *Architecture and Early Experience with Planning for the ALV*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), 1986.
- [Lozano-Perez 79] T. Lozano-Perez & M. Wesley. *An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles*. Communications of the ACM, vol. 22, no. 10, October 1979.
- [Lumelsky 90] V.J. Lumelsky, S. Mukhopadhyay & K. Sun. *Dynamic Path Planning in Sensor-Based Terrain Acquisition*. IEEE Transactions on Robotics and Automation, vol. 6, no. 4, 1990.
- [Matthies 87] L. Matthies & S.A. Shafer. *Error Modeling in Stereo Navigation*. IEEE Journal of Robotics and Automation, vol. 3, no. 3, 1987.
- [Matthies 88] L. H. Matthies & A. Elfes. *Integration of Sonar and Stereo Range Data Using a Grid-Based Representation*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), 1988.
- [Matthies 92] L. H. Matthies. *Stereo Vision for Planetary Rovers : Stochastic Modeling to Near Real Time Implementation*. International Journal of Computer Vision, vol. 8, no. 1, pages 71-91, 1992.
- [Maurette 92] M. Maurette. *CNES research for orbital and planetary robots*. In Toulouse (France) CEPADUES Ed., éditeur, Intelligence Artificielle, Robotique et Automatique, Appliquées à l'espace, pages 43-52, September 1992.
- [Metea 87] M. B. Metea & J. J-P. Tsai. *Route Planning for Intelligent Autonomous Land Vehicles using Hierarchical Terrain Representation*. In IEEE International Conference on Robotics and Automation, Raleigh (USA), 1987.
- [Milian 94] T. Milian. ... Rapport technique, Rapport de fin d'études de l'Ecole Supérieure de l'Aéronautique et de l'Espace, 1994.
- [Miller 92] D.P. Miller, R.S. Desai, E. Gat, R. Ivlev & J. Loch. *Reactive Navigation through Rough Terrain : Experimental Results*. In 10th National Conference on Artificial Intelligence (AAAI '88), San Jose, Ca (USA), 1992.
- [Mitchell 87] J.S.B. Mitchell & C.H. Papadimitriou. *The Weighted Region Problem*. In ACM Third Annual Symposium on Computational Geometry, Waterloo, Ontario (USA), 1987.
- [Mitchell 88] J.S.B. Mitchell. *An Algorithmic Approach to some Problems in Terrain Navigation*. Artificial Intelligence, vol. 37, no. ?, 1988.

- [Mitchell 91] J.S.B. Mitchell & C.H. Papadimitriou. *The Weighted Region Problem : Finding Shortest Paths Through a Weighted Planar Subdivision*. Journal of the Association for Computing Machinery, vol. 38, no. 1, 1991.
- [Moravec 85] H.P. Moravec & A. Elfes. *High Resolution Maps from Wide Angle Sonar*. In IEEE International Conference on Robotics and Automation, St Louis (USA), 1985.
- [Moutarlier 91] P. Moutarlier. *Modélisation Autonome de l'Environnement par un Robot Mobile*. PhD thesis, Thèse de l'Université Paul Sabatier, Toulouse (France), 1991.
- [Nashashibi 92] F. Nashashibi, M. Devy & P. Fillatreau. *Indoor Scene Terrain Modeling using Multiple Range Images for Autonomous Mobile Robots*. In IEEE International Conference on Robotics and Automation, Nice, (France), pages 40–46, May 1992.
- [Nashashibi 93] F. Nashashibi. *Perception et Modélisation de l'Environnement Tridimensionnel pour la Navigation Autonome d'un Robot Mobile*. Thèse de l'Université Paul Sabatier, Toulouse (France) 93032, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), February 1993.
- [Nashashibi 94] F. Nashashibi, P. Fillatreau, B. Dacre-Wright & T. Simeon. *3D Autonomous Navigation in a Natural Environment*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Nilsson 69] M. Nilsson. *A mobile automaton: an application of artificial intelligence techniques*. In proc. of the first IJCAI, pages 509–520, 1969.
- [Olin 91] K.E. Olin & D. T. Tseng. *Autonomous Cross-Country Navigation*. IEEE Expert, vol. 6, no. 4, August 1991.
- [Oriolo 94] G. Oriolo, G. Ulivi & M. Venditelli. *Motion Planning with Uncertainties : Navigation on Fuzzy Maps*. In Fourth IFAC Symposium on Robot Control, Capri, Italy, 1994.
- [O'rourke 87] J. O'rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.
- [Orteu 91] J.J. Orteu. *Application de la Vision par Ordinateur l'Automatisation de l'Abattage dans les Mines*. PhD thesis, Thèse de l'Université Paul Sabatier, Toulouse (France), 1991.
- [Pagnot 95] R. Pagnot & P. Grandjean. *Fast Cross-Country Navigation on Fair Terrains*. In Submitted to IEEE International Conference on Robotics and Automation, 1995.
- [Parodi 86] A. M. Parodi, J. J. Nitao & L. S. McTamane. *An Intelligent System for an Autonomous Vehicle*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), 1986.
- [Perret 94] J. Perret, C. Proust, A. Alami & R. Chatila. *How to Tele-program a Remote Intelligent Robot*. In IEEE International Conference on Intelligent Robots and Systems (IROS '94), Munich (Germany), 1994.
- [Pignon 91] P. Pignon. *Structuration de l'Espace pour une Planification Hiérarchisée des Trajectoires de Robots Mobiles*. PhD thesis, Thèse de l'Université Paul Sabatier, Toulouse (France), 1991.
- [Postaire 87] J. G. Postaire. *De l'image à la décision - analyse des images numériques et théorie de la décision*. DUNOD informatique, 1987.
- [Rao 86] N.S.V. Rao, S.S. Iyengar, C.C. Jorgensen & C.R. Weisbin. *Robot Navigation in an Unexplored Terrain*. Journal of Robotic Systems, vol. 3, no. 4, 1986.

- [Rao 91] N.S.V. Rao, N. Stoltzfus & S.S. Iyengar. *A Retraction Method for Learned Navigation in Unknown Terrains for a Circular Robot*. IEEE Transactions on Robotics and Automation, vol. 7, no. 5, 1991.
- [Richard 94] F. Richard, H. Chochon, O. Faugeras, P. Grandjean & T. Siméon. *Autonomous navigation for a planetary rover*. In 2nd International Symposium on Missions, Technologies and Design of Planetary Rovers, Moscow (Russia), also in International Symposium on Artificial Intelligence, Robotics, and Automation for Space. Pasadena, Ca., 1994.
- [Richbourg 86] R.F. Richbourg. *Solving a Two Dimensional Path Planning Problem Using Topographical Knowledge of the Environment and Capability Constraints*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), 1986.
- [Richbourg 87] R.F. Richbourg, N.C. Rowe, M.J. Zyda & R.B. Mac Ghee. *Solving global, two-dimensional routing problems using Snell's law and A* search*. In IEEE International Conference on Robotics and Automation, Raleigh (USA), April 1987.
- [Shafer 86] S. A. Shafer, A. S. Stenz & C. E. Thorpe. *An architecture for sensor fusion in a mobile robot*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), pages 2002-2011, 1986.
- [Siméon 93] T. Siméon & B. Dacre Wright. *A Practical Motion Planner for All-terrain Mobile Robots*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '93) Japan, July 1993.
- [Simmons 91] R. Simmons & E. Krotkov. *An Integrated Walking System for the Ambler Planetary Rover*. In IEEE International Conference on Robotics and Automation, Sacramento, (USA), 1991.
- [Simmons 93] R. Simmons & E. Krotkov. *Autonomous Planetary Exploration : From Ambler to APEX*. In '93 International Conference on Advanced Robotics, Tokyo, Japan, pages 429-434, 1993.
- [Singh 91] S. Singh & P. Keller. *Obstacle Detection for High Speed Autonomous Navigation*. In IEEE International Conference on Robotics and Automation, Sacramento, (USA), 1991.
- [Smith 86] R.S. Smith & M. Gini. *Robot Tracking and Control Issues in an Intelligent Error Recovery System*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), April 1986.
- [Stenz 94a] A. Stenz. *Optimal and Efficient Path Planning for Partially-Known Environments*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Stenz 94b] A. Stenz & M. Hebert. *A Complete Navigation System for Goal Acquisition in Unknown Environments*. Rapport technique CMU-RI-TR-94-07, Robotics Institute, Carnegie Mellon University, 1994.
- [Takeda 94] H. Takeda & J.C. Latombe. *Sensory Uncertainty Field for Robot Navigation*. In IEEE International Conference on Robotics and Automation, Nice, (France), 1994.
- [Tarabanis 91] K. Tarabanis, R.Y. Tsai & P.K.Allen. *Automated Sensor Planning for Robotic Vision Tasks*. In IEEE International Conference on Robotics and Automation, Sacramento, (USA), pages 76-82, 1991.
- [Thorpe 88] 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, no. 3, pages 362-373, May 1988.

- [Thorpe 91a] C. Thorpe, M. Hebert, T. Kanade & S. Shafer. *Toward Autonomous Driving : the CMU Navlab. Part I : Perception*. IEEE Expert, vol. 6, no. 4, August 1991.
- [Thorpe 91b] C. Thorpe, M. Hebert, T. Kanade & S. Shafer. *Toward Autonomous Driving : the CMU Navlab. Part II : Architecture and Systems*. IEEE Expert, vol. 6, no. 4, August 1991.
- [Timcenko 94] A. Timcenko & P. Allen. *Probability-Driven Motion Planning for Mobile Robots*. In IEEE International Conference on Robotics and Automation, San Diego, California, 1994.
- [Wang 93] C. T. Wang & P. Chrétienne. *Routage des Voiliers et Programmation Dynamique*. RAIRO - Recherche Opérationnelle, vol. 27, no. 1, 1993.
- [Weisbin 92] C.R. Weisbin, M. Montenerlo & W. Whittaker. *Evolving Directions in NASA's Planetary Rover Requirements and Technology*. In Missions, Technologies and Design of Planetary Mobile Vehicules. Centre National d'Etudes Spatiales, France, Sept 1992.
- [Wilcox 92] B. Wilcox, L. Matthies, D. Gennery, B. Cooper, T. Nguyen, T. Litwin, A. Mishkin & H. Stone. *Robotic Vehicles for Planetary Exploration*. In IEEE International Conference on Robotics and Automation, Nice, (France), 1992.
- [Wu 90] H.L. Wu & A. Cameron. *A Bayesian Decision Theoretic Approach for Adaptive Goal-Directed Sensing*. In Third International Conference on Computer Vision, Osaka (Japan), pages 563-567, 1990.
- [Xu 94] H. Xu & J. Vandorpe. *Perception Planning in Mobile Robot Navigation*. In IEEE International Conference on Multisensir Fusion and Integration for Intelligent Systems, Las Vegas (USA), 1994.
- [Zelinsky 92] A. Zelinsky. *A Mobile Robot Exploration Algorithm*. IEEE Transaction on Robotics and Automation, vol. 8, no. 6, Dec. 1992.

Thèse de Simon LACROIX

Stratégies de Perception et de Déplacement pour la Navigation d'un Robot Mobile Autonome en Environnement Naturel

Cette thèse porte sur la détermination autonome de stratégies de navigation pour un robot mobile évoluant dans un environnement extérieur initialement inconnu et non structuré. Ces stratégies concernent le choix de buts intermédiaires à rallier pour atteindre un but final, d'un mode de déplacement à appliquer, et de la prochaine tâche de perception à effectuer. Le mémoire est composé de deux parties : la première présente les algorithmes développés afin de construire une représentation topologique de l'environnement adaptée à la prise des diverses décisions stratégiques, et la seconde concerne les prises de décision proprement dites.

Étant données la complexité et la diversité d'un environnement extérieur naturel d'une part, et les grandes incertitudes que l'on a sur les données fournies par les capteurs d'autre part, nous avons favorisé pour la construction de la représentation topologique de l'environnement une technique de classification probabiliste. Les données tridimensionnelles (issues d'un télémètre laser ou d'un système de stéréovision) sont ainsi rapidement analysées de manière à produire une description de la zone perçue en termes de régions liées à la navigabilité du robot. Les données acquises de différents points de vue sont fusionnées en un modèle global, dans lequel figurent les incertitudes résultant des caractéristiques du capteur, et qui est structuré en un graphe de connexité de régions.

C'est sur la base de ce modèle global, des modèles des capacités de déplacement et de perception du robot, et de la définition de la mission à réaliser (critères portant sur le temps et l'énergie à minimiser), que sont effectués les choix stratégiques. Une analyse du problème montre que sa difficulté provient essentiellement de sa complexité algorithmique et du caractère incertain des modèles de l'environnement et des capteurs. Une approche réaliste est présentée : elle consiste à déterminer le chemin au sein du graphe minimisant un coût, lequel prend en compte, outre les différents critères à minimiser, les incertitudes liées au modèle de l'environnement. Des résultats obtenus lors d'expérimentations sur un robot mobile réel sont présentés et analysés tout au long du mémoire.

Mots clés : Robots Mobiles, Navigation autonome, Modélisation d'environnements naturels, Stratégies de déplacement, Stratégies de perception.

Perception and Motion Strategies for Mobile Robot Autonomous Navigation in a Natural Environment

This thesis is about the determination of navigation strategies for an autonomous mobile robot roving in an unstructured and initially unknown outdoor environment. These strategies concern the selection of intermediary goals to reach, of a motion mode to apply, and of the next perception task to execute. The work is divided in two parts : the first one presents the algorithms that produce a topological representation of the environment adapted to the strategic decision, and the second one concerns the strategic decisions.

Given the diversity and the complexity of natural outdoor environments, and the uncertainties on the data provided by the sensors, we favored a probabilistic classification technique to build the topological representation. The 3D data (produced by a stereovision algorithm or a laser range finder) are analyzed in order to produce a description of the perceived zone in terms of regions related to the robot navigation capabilities. The data acquired from different positions are fused in a global model, which is structured in a region connection graph.

The strategic decision are taken on the basis of this global model, of the models of the robot motion and perception capabilities, and of the mission description (criteria related to time and energy to minimize). An analysis of the problem shows that its difficulty arises from its algorithmic complexity and from the uncertain characteristics of the environment and sensors models. A realistic approach is presented : it relies on the determination of an optimal path in the connection graph that takes into account the uncertainties of the environment model. Experimental results on a real robot are presented and analyzed along the thesis.

Keywords : Mobile Robots, Autonomous Navigation Outdoor environment modelisation, Motion strategies, Perception Strategies.
