



**HAL**  
open science

# Localisation et modélisation tridimensionnelles pour un robot mobile autonome tout terrain

Philippe Fillatreau

► **To cite this version:**

Philippe Fillatreau. Localisation et modélisation tridimensionnelles pour un robot mobile autonome tout terrain. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 1994. Français. NNT: . tel-00261834

**HAL Id: tel-00261834**

**<https://theses.hal.science/tel-00261834>**

Submitted on 10 Mar 2008

**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.

# THESE

présentée au

Laboratoire d'Automatique et d'Analyse des Systèmes du CNRS

en vue de l'obtention du

Doctorat de l'Université Paul Sabatier de Toulouse

Spécialité: Robotique

par

**Philippe FILLATREAU**

Ingénieur ENSEEIHT

---

## LOCALISATION ET MODELISATION TRIDIMENSIONNELLES POUR UN ROBOT MOBILE AUTONOME TOUT TERRAIN

---

Soutenue le 19 Janvier 1994 devant le jury composé de :

MM.	Georges	GIRALT	}	Président
	Roland	PRAJOUX		Directeur de thèse
	James	CROWLEY	}	Rapporteurs
	Jean-Louis	LACOMBE		
	Bertrand	ZAVIDOVIQUE		
	Serge	CASTAN	}	Examineurs
	Michel	DEVY		
	Michel	TROUBLÉ		

Rapport LAAS N° 94108

Cette thèse a été préparée au Laboratoire d'Automatique et d'Analyse des Systèmes du CNRS  
7, avenue du Colonel Roche 31 077 Toulouse Cedex

**A mes parents Jacques et Carmen,**

**A Nicole,**

**A mes frères  
Jean-Luc, José, Vincent et Camille,**

**A toute ma famille.**

# Avant-Propos

Ce mémoire de thèse présente la part la plus importante des travaux que j'ai effectués au Laboratoire d'Automatique et d'Analyse des Systèmes (L.A.A.S.) du C.N.R.S., au sein du groupe Robotique et Intelligence Artificielle (RIA).

Je remercie Monsieur Alain Costes, Directeur du L.A.A.S., de m'avoir accueilli dans son laboratoire.

Je remercie également Monsieur Georges Giralt, responsable du groupe RIA, de m'avoir permis de travailler dans son groupe de recherche et de mener ce travail jusqu'au bout, et d'avoir honoré ma thèse par la présidence de mon Jury d'examen.

Je tiens à remercier Monsieur Roland Prajoux, mon directeur de recherche, pour l'intérêt constant porté à mon travail, pour m'avoir fait bénéficier de ses conseils et de sa compétence, et pour la confiance et la liberté qu'il m'a accordées.

Cette thèse doit beaucoup à Monsieur Michel Devy, Ingénieur de Recherche au C.N.R.S., qui a guidé mes premiers pas (et les suivants ...) dans le monde de la Perception, et dont la compétence, les conseils et encouragements m'ont été très précieux. Je lui exprime ici toute ma reconnaissance.

Je remercie particulièrement Messieurs James Crowley, Professeur à l'Institut National Polytechnique de Grenoble, Jean-Louis Lacombe, Directeur Technique Adjoint à Matra Hachette, et Bertrand Zavidovique, Professeur à l'Université Paris XI, qui ont fait un très grand honneur à cette thèse en acceptant d'en être les rapporteurs et de siéger dans mon jury de thèse, et par l'intérêt qu'ils ont porté à ce travail. Leurs remarques ont été très précieuses pour l'amélioration du manuscrit. Je remercie Monsieur Serge Castan, Professeur à l'Université Paul Sabatier, pour avoir également accepté de participer à mon jury.

Ce travail de thèse s'est déroulé pour l'essentiel dans le cadre d'une convention CIFRE unissant le LAAS et Framatome. Je remercie la société Framatome (et en particulier Messieurs Clar, Ollier et Delevallée) d'avoir rendu ce travail possible en m'accordant sa confiance. Je remercie plus particulièrement Monsieur Omer Mercier, Chef du Département Développement Robotique à Framatome Lyon, pour l'intérêt qu'il a toujours porté à ce travail. Je suis très reconnaissant à Monsieur Michel Troublé, Ingénieur Roboticien à Framatome Paris, chargé du suivi technique de mon travail, d'avoir accepté de siéger à mon jury de thèse. Je le remercie également pour sa gentillesse, son soutien et sa confiance tout au long de ces années.

Merci à Matthieu Herrb, Ingénieur de Recherche au CNRS, dont les compétences en informatique m'ont été précieuses, et pour son aide durant les "manips". Merci aussi à Guy Vialaret, Gérard Bauzil et Christian Lemaire qui font des robots Hilare des plateformes expérimentales disponibles et évolutives.

Merci à Michel Courdesses, Professeur à l'Université Paul Sabatier, pour ses conseils, dans ma recherche bibliographique CAO notamment, et sa gentillesse. Je remercie chaleureusement Christophe Rabut, Maître de Conférences à l'Institut National des Sciences Appliquées de Toulouse (Centre de Mathématiques) dont l'expérience concernant les splines et les encouragements m'ont été précieux.

Je tiens aussi à remercier Jackie et Nicole pour leur gentillesse et leur disponibilité.

Je voudrais également remercier tous ceux qui ont contribué, par le travail d'équipe que nous avons mené, à l'aboutissement de ce travail. Merci en particulier à Benoît Dacre Wright, Christophe Proust pour sa disponibilité et son aide lors des "manips", et l'ensemble des équipes EDEN et AMR, avec qui discuter, développer, debugger et expérimenter a été un réel plaisir.

J'exprime tout particulièrement ma reconnaissance à Fawzi Nashashibi, avec qui j'ai travaillé dans une parfaite osmose durant ces années. Pour notre collaboration scientifique, pour ton soutien et ton amitié, tope là, Fawzi!

Un grand merci également à tous ceux qui m'ont apporté leur aide pour la réalisation du manuscrit ou la préparation de la soutenance de ce travail: Thierry Vidal pour les photos d'écrans (et les lumières), Maher Khatib pour le montage du film, Florence Marois et Patricia Lasserre.

Merci également aux services de documentation du LAAS, qui m'ont été si souvent précieux pour la réalisation et le tirage de ce mémoire.

Merci encore à tout ceux qui ont contribué à la chaleureuse ambiance qui règne dans le groupe RIA: les "oiseaux de nuit" (Titou, Bernard, "Montagne Noire" Hervé, Maher, Juju, Ericd) qui ont égayé les longues soirées de travail, mais aussi mes voisins de bureau ("RMP" Juju, "2 nanosecondes" Jean-José, et "Fangio Spaghetti" Hanna), tous les Rochelais, Simon, Nicola, Bruno, André, Maurice, Mourad, Alain, Jean-Paul, Sara, Philippe(s), Patricia, Monique, Flo, Marc(s), Malik, Raja, Félix, Rachid(s), Christophe, Jean-Philippe, Luc-Henri, Sylvie, Vianney, Thierry T., Shooji, Joël, Pierrick, Vincent, Olivier, Stéphane, Paul, Jean-Yves, Victor, Flavien, Jean-Claude, Luis, Sepanta et tous les autres. Merci aussi à tous mes amis de Toulouse et d'ailleurs.

Enfin et surtout, merci, pour leur amour, leur soutien sans faille et leur patience, à Nicole, à mes frères Jean-Luc, José, Vincent et Camille, et à mes parents Jacques et Carmen. Vous m'avez tout donné, et vous m'avez appris l'essentiel !

\* \* \*

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Introduction</b>	<b>1</b>
<b>I Contexte du travail, problématique et choix</b>	<b>5</b>
<b>Chapitre 1 Le contexte du travail</b>	<b>7</b>
1.1 Généralités . . . . .	7
1.2 La perception d'un robot mobile autonome . . . . .	7
1.3 La localisation . . . . .	10
1.4 Le contexte applicatif . . . . .	12
1.4.1 Le projet AMR . . . . .	13
1.4.2 Le projet VAP . . . . .	15
1.4.3 L'expérimentation EDEN . . . . .	16
1.4.4 Les moyens expérimentaux . . . . .	18
<b>Chapitre 2 Les Capteurs pour la Localisation/Modélisation</b>	<b>21</b>
2.1 Les différents capteurs utilisables . . . . .	21
2.1.1 Les capteurs pour la trajectométrie . . . . .	21
2.1.2 Les capteurs extéroceptifs . . . . .	23
2.2 Les capteurs inertiels . . . . .	26
2.2.1 Les capteurs "élémentaires" . . . . .	26
2.2.2 Les systèmes envisageables en robotique mobile . . . . .	28
2.2.3 La centrale équipant le robot ADAM . . . . .	31
2.3 Les capteurs laser . . . . .	31
2.3.1 Principe de fonctionnement des télémètres laser . . . . .	31
2.3.2 Les problèmes de calibrage . . . . .	32
2.3.3 Le calibrage intrinsèque: télémètre HILARE II / ADAM . . . . .	33
2.3.4 Le calibrage extrinsèque Robot-Capteur . . . . .	34

<b>Chapitre 3 Les Modèles</b>	<b>37</b>
3.1 Revue de méthodes de modélisation 3D . . . . .	37
3.1.1 La modélisation de formes structurées . . . . .	37
3.1.2 La modélisation de formes non structurées ou de terrain . . . . .	40
3.2 Les représentations choisies . . . . .	49
3.2.1 Le modèle de localisation en milieu structuré . . . . .	49
3.2.2 Le modèle de terrain . . . . .	49
<b>Conclusion de la partie I</b>	<b>53</b>
<b>II Quand le milieu est semi-structuré</b>	<b>55</b>
<b>Introduction à la partie II</b>	<b>57</b>
<b>Chapitre 4 Outils pour l'appariement et la manipulation de données im-</b> <b>précises</b>	<b>59</b>
4.1 L'appariement . . . . .	59
4.2 Imprécisions et fusion numérique . . . . .	61
<b>Chapitre 5 Localisation en Milieu Semi-Structuré</b>	<b>67</b>
5.1 Recalage 2D sur lignes verticales . . . . .	67
5.1.1 Extraction de primitives hétérogènes: le modèle instantané de locali- sation . . . . .	68
5.1.2 Estimation du mouvement et mise à jour du modèle de localisation . .	69
5.1.3 Résultats expérimentaux . . . . .	73
5.1.4 Extension possible: estimation de la verticale . . . . .	78
5.2 Recalage 2D sur obstacles verticaux . . . . .	79
5.2.1 Modèle instantané de localisation . . . . .	79
5.2.2 Estimation du mouvement et mise à jour du modèle local de localisation	80
5.2.3 Résultats expérimentaux . . . . .	81
5.3 Coopération entre les deux méthodes . . . . .	81
5.3.1 Intégration des deux méthodes de localisation . . . . .	83
5.3.2 Choix des amers et recalage du robot . . . . .	83
<b>Conclusion de la partie II</b>	<b>89</b>
<b>III Quand le milieu est non structuré</b>	<b>93</b>
<b>Introduction à la partie III</b>	<b>95</b>

<b>Chapitre 6 Le Modèle du Terrain</b>	<b>97</b>
6.1 B-splines: le formalisme adopté . . . . .	97
6.1.1 B-splines en dimension 1, courbes B-splines . . . . .	97
6.1.2 Surfaces B-splines de “produit tensoriel” . . . . .	100
6.1.3 Interpolation ou approximation ? . . . . .	104
6.2 Modèle hiérarchique du terrain . . . . .	106
6.2.1 Réseau de carreaux, multi-résolution et approche hiérarchique . . . . .	106
6.2.2 Choix du degré des B-splines . . . . .	108
6.2.3 Construction du modèle hiérarchique . . . . .	108
6.2.4 Résultats expérimentaux . . . . .	114
6.3 Modélisation incrémentale . . . . .	118
6.3.1 Localité . . . . .	119
6.3.2 Deux approches au problème de modélisation incrémentale . . . . .	122
6.3.3 Fusion “données instantanées à spline locale” . . . . .	122
6.3.4 Fusion “spline locale à spline instantanée” . . . . .	123
6.4 Conclusion et orientations futures . . . . .	126
<b>Chapitre 7 Localisation en Milieu Non-Structuré</b>	<b>127</b>
7.1 Recalage en milieu non-structuré: état de l’art . . . . .	127
7.2 Quels invariants 3D ? . . . . .	132
7.3 Extraction de caractéristiques de terrain . . . . .	133
7.3.1 Pics d’élévation . . . . .	133
7.3.2 Régions de forte courbure . . . . .	138
7.4 Recalage: approche adoptée . . . . .	142
7.4.1 Recalage par mise en correspondance de points . . . . .	142
7.4.2 Recalage par coopération avec un appariement iconique . . . . .	143
7.5 Résultats expérimentaux . . . . .	146
7.6 Conclusion et orientations futures . . . . .	152
<b>Conclusion</b>	<b>155</b>
<b>Annexes</b>	<b>159</b>
<b>Annexe A Calibrage intrinsèque du télémètre HILARE II / ADAM</b>	<b>161</b>



<b>Annexe B Calibrage extrinsèque robot-capteur par l'approche de Tsai et Lenz</b>	<b>165</b>
B.1 La procédure de calibrage . . . . .	165
B.2 Dispositif expérimental et résultats . . . . .	166
<b>Annexe C Le filtre de Kalman généralisé: l'outil YAFF</b>	<b>168</b>
<b>Annexe D Extrémum d'une courbe ou surface B-splines</b>	<b>170</b>
D.1 Cas du degré 2 . . . . .	170
D.2 Cas du degré 1 . . . . .	171
<b>Annexe E Courbure en un point d'une surface B-splines</b>	<b>173</b>
<b>Références bibliographiques</b>	<b>175</b>

# Introduction Générale

Notre travail se situe dans le domaine de la robotique mobile d'intervention [Giralt 93]; nous considérons la classe des robots qui doivent effectuer un certain nombre de tâches dans des environnements hostiles à l'homme. De tels robots doivent être technologiquement capables d'affronter les contraintes liées à l'environnement (milieu sous-marin, espace, ...), mais de nombreuses autres contraintes, spécifiques des applications visées, doivent également être prises en compte. Elles sont liées aux communications (temps de transmission, bande passante faible, ... – pour un véhicule planétaire notamment –), aux tâches à effectuer (leur durée ou leur non-répétabilité), à la connaissance de l'environnement (connaissance partielle ou nulle, environnement susceptible de changer, ...).

Les applications concernées peuvent être divisées en deux grands domaines:

1. Celles qui se déroulent dans des environnements conçus pour coopérer avec la machine ou le robot, et qui peuvent être connus a priori avec une assez grande précision: station orbitale, site sous-marin, ...
2. Celles qui se déroulent dans des environnements inconnus ou partiellement connus, qui peuvent être non coopératifs, éventuellement naturels ou non structurés. Citons par exemple les applications suivantes: exploration sous-marine (surveillance de site, intervention d'un véhicule autonome inhabité), robotique d'intervention pour la sécurité civile, robots mobiles pour l'exploration planétaire.

Dans ce travail, nous nous situons dans le deuxième domaine. La connaissance disponible a priori est souvent d'un niveau de description géométrique assez grossier et fournit surtout des informations intéressantes sur la topologie des lieux. Dans le cas d'un robot de sécurité civile, destiné à intervenir sur un site sinistré (de type usine par exemple), cette connaissance peut être constituée par un plan d'ensemble de l'usine, donnant l'emplacement et les dimensions approximatives des différentes pièces, des couloirs, des portes, ... Mais la configuration des lieux peut avoir été modifiée par le sinistre. Dans le cas d'un robot mobile d'exploration planétaire, la connaissance disponible a priori sur l'environnement peut être donnée sous la forme de cartes de terrain issues d'images orbiteur, de faible résolution en général.

Cette connaissance a priori sur l'environnement permet de dessiner les grandes lignes d'une mission: position approximative du but, sous-buts à atteindre, amers à identifier, ... Mais l'information géométrique qu'elle renferme est souvent trop pauvre pour suffire à elle seule

à garantir une navigation sûre du robot, et il faudra donc *modéliser* de manière plus fine l'environnement. D'autre part, les communications entre le robot et sa station de contrôle peuvent être difficiles, ou momentanément coupées.

Le robot mobile doit donc être capable, dans certains cas, d'évoluer de manière autonome dans un environnement inconnu ou partiellement connu. Au fur et à mesure qu'il se déplace et explore son environnement, il doit pouvoir acquérir, organiser, ou remettre à jour des informations sur cet environnement. Pour construire incrémentalement, au cours du déplacement, le modèle du monde, il faut connaître avec une précision qui dépend de la tâche et du contexte, la position à laquelle est effectuée chaque nouvelle acquisition sensorielle.

Le robot doit donc être notamment à la fois capable de *percevoir* son environnement et de se *localiser*, c'est à dire d'avoir une estimation de sa position, dans celui-ci. C'est dans le contexte de la *perception*, de la *modélisation* de l'environnement et de la *localisation* que se situe notre étude.

Ces fonctionnalités doivent permettre au robot de *s'adapter* aux différentes situations qu'il peut rencontrer. Un robot mobile d'intervention doit pouvoir affronter des environnements très variés, qui peuvent être plus ou moins structurés, ou complètement non structurés.

Le mémoire s'organise en trois parties.

Dans la partie I, nous présentons le contexte scientifique, mais aussi applicatif, de notre travail. Les fonctionnalités dont nous souhaitons doter le robot sont évoquées: dans ce travail, nous nous sommes intéressés tout particulièrement au problème de la localisation d'un robot mobile tout terrain dans un environnement tridimensionnel, et à celui de la modélisation de cet environnement. Les grands projets dans le cadre desquels s'inscrit ce travail sont présentés. Un état de l'art concernant d'une part les principaux capteurs utilisables en robotique mobile, et d'autre part la modélisation de formes structurées ou non-structurées, est présenté. Une comparaison essentiellement qualitative des différents capteurs et modèles est effectuée. Nous justifions ainsi le choix des capteurs utilisés, ainsi que des primitives géométriques sur l'utilisation desquelles les techniques de modélisation ou de localisation présentées par la suite s'appuient.

Dans la partie II, nous nous intéressons au module de recalage d'un robot mobile dans un environnement *semi-structuré*, c'est à dire dans lequel peuvent figurer un certain nombre d'amers de localisation de forme géométrique simple. Le recalage est alors effectué sur la base d'amers de forme structurée, mais éventuellement *naturels*. Le modèle de localisation met en oeuvre des *primitives hétérogènes*, pour prendre en compte la diversité des amers utilisables: lignes verticales de discontinuité de profondeur, arbres, colonnes, murs, ... La contribution de cette partie se situe surtout au niveau de l'introduction de nouvelles formes de primitives géométriques et l'intégration de plusieurs sortes de primitives dans un même modèle. Une stratégie de choix des amers est proposée. Les méthodes proposées ont été validées par des expérimentations de déplacement du robot mobile Hilare II dans un environnement semi-structuré.

---

Dans la partie III, nous nous intéressons au cas où le robot évolue dans un environnement complètement non-structuré. Nous présentons une méthode de modélisation géométrique incrémentale d'un terrain accidenté reposant sur l'utilisation d'un arbre hiérarchique de *primitives non planes* (B-splines de produit tensoriel). Nous montrons comment la construction de ce modèle peut être *dédiée à différents types de tâches* à accomplir, et en particulier la *localisation*. Dans le cadre de celle-ci, le modèle proposé permet, par une approche multi-résolution, une *focalisation* de la recherche de caractéristiques de terrain utilisables pour le recalage. Ce dernier est réalisé grâce à une exploitation efficace de plusieurs représentations du terrain pour effectuer des appariements iconiques. On aboutit à des temps de calcul assez rapides par rapport aux méthodes classiques de recalage tridimensionnel dans un environnement non structuré. Les méthodes développées ont fait l'objet d'une validation expérimentale sur le robot mobile tout terrain ADAM, lors d'expérimentations de déplacement en environnement naturel, non structuré.



## **Partie I**

# **Contexte du travail, problématique et choix**



# Chapitre 1

## Le contexte du travail

### 1.1 Généralités

Un robot mobile autonome doit être capable de *percevoir* son environnement. En fonction de la tâche à réaliser, il doit le *modéliser* à partir des informations acquises, c'est à dire en construire une représentation utilisable par les composants décisionnels du système. Pour pouvoir construire incrémentalement des modèles cohérents de l'environnement, enfin, le robot doit être capable de se *localiser*.

Pour cela, il doit être équipé de capteurs que l'on peut fonctionnellement classer suivant les deux familles suivantes:

- Les capteurs *proprioceptifs* permettent de prendre des mesures internes au robot. Parmi ceux-ci, la sous-famille des *capteurs de déplacement* est particulièrement intéressante dans notre contexte. Ils fournissent une estimation du *mouvement* au cours des déplacements du robot. Dans tout le mémoire, nous utiliserons l'expression "capteur proprioceptif" comme raccourci de "capteur proprioceptif de déplacement".
- Les capteurs *extéroceptifs* permettent d'acquérir des informations sur l'environnement extérieur du robot.

C'est dans le contexte de la perception, de la modélisation et de la localisation que se situe notre étude.

Nous allons dans la suite de ce chapitre nous intéresser plus particulièrement au module de perception d'un robot mobile, puis mettre en lumière les problèmes auxquels notre travail s'est intéressé.

### 1.2 La perception d'un robot mobile autonome

Les buts du sous-système de perception d'un robot mobile autonome qui rentrent dans le cadre du problème qui nous intéresse, sont de construire les représentations internes



(modèles) de l'environnement, à partir desquelles il sera possible: d'extraire des primitives géométriques 3D utiles pour les mouvements asservis du robot ou le recalage de sa position, d'identifier des amers dans l'environnement (porte, coin de mur dans un environnement d'intérieur ou d'abord de bâtiments, rocher ou arbre particulier en environnement naturel), et d'analyser le terrain pour construire un modèle de navigation, ... Durant la mission, les fonctions de modélisation vont permettre de construire incrémentalement les modèles de l'environnement, par agrégation (ou plutôt *fusion*) des modèles successifs élaborés à partir des données sensorielles correspondant aux différentes acquisitions. Nous distinguons suivant des critères spatio-temporels:

- **Les modèles instantanés:** Ces modèles sont construits à partir des données sensorielles courantes (correspondant à une acquisition sensorielle donnée).
- **Les modèles locaux:** Ils résultent de la fusion de plusieurs modèles instantanés acquis dans un même lieu topologique (par exemple, une même pièce, un même couloir à l'intérieur de bâtiments).
- **Les modèles globaux:** Ils résultent de l'agrégation de tous les modèles locaux construits durant une mission donnée. Lors d'une intervention dans une usine sinistrée par exemple, il sera constitué de l'ensemble des pièces, couloirs, halls, ... formant l'intérieur des bâtiments. Le modèle obtenu est un graphe, dont les noeuds sont les modèles locaux construits au cours de la mission, et dont chaque arc correspond à la transformation géométrique 3D relative entre les repères locaux correspondant aux deux noeuds reliés par cet arc, ainsi que, éventuellement, à une relation d'adjacence.

Soulignons que cette terminologie a été adoptée dans le groupe RIA dans le contexte du projet AMR (cf. paragraphe 1.4.1). Par souci d'homogénéité, nous l'avons conservée pour l'ensemble du travail présenté dans ce mémoire. Elle n'est pas toujours conforme à celles rencontrées dans d'autres milieux.

Le système de repères dans lesquels seront exprimés les différents modèles manipulés est composé des repères suivants:

- $R_C$ , repère cartésien lié au capteur;
- $R_R$ , repère cartésien lié au robot, dont l'origine, classiquement, est prise au point de référence des mesures du mouvement (entre les roues motrices pour HILARE); la transformation entre  $R_R$  et  $R_C$  est décrite par la matrice homogène de passage  $T_{RC}$ , évaluée par une procédure de calibrage dite *extrinsèque* (cf. paragraphe 2.3.4).
- $R_{R_r}$ , repère robot redressé, connaissant l'attitude du robot.
- $R_L$ , repère local dans lequel un modèle local est exprimé; chacun des repères locaux est propre au lieu topologique dans lequel le robot se déplace.

- $R_G$ , repère global, dans lequel sont exprimés les modèles locaux construits par le robot et les connaissances a priori sur l'environnement. L'origine de ce repère doit être parfaitement connue au départ de la mission. Elle peut être liée à une entité ou un lieu topologique connus, ou au point de départ de la mission.

Les modèles constituent une représentation de l'environnement qui doit être structurée en fonction des trois grandes tâches à satisfaire:

- **Localisation:** Les capteurs proprioceptifs dédiés à la trajectométrie (odomètres, inclinomètres, centrale inertielle ...) fournissent au cours du mouvement du robot une estimation de sa position.

Le robot est localisé dans son environnement tridimensionnel à l'aide de six paramètres: trois de position ( $X, Y, Z$ ), et trois d'orientation ( $\theta, \phi, \psi$ ) qui désignent respectivement le *lacet*, le *tangage* et le *roulis*. Si  $P_L$  et  $P_R$  désignent un même point 3D exprimé par ses coordonnées homogènes respectivement dans le repère local et dans le repère robot, le changement de repère s'exprime à l'aide d'une matrice homogène par:

$$P_L = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} P_R$$

avec  $R = R_\theta \times R_\phi \times R_\psi$  et  $T = (X \ Y \ Z)^T$ .

Les capteurs proprioceptifs commettent des erreurs cumulatives (cf. 2.1.1). Un recalage extéroceptif est donc nécessaire. Le robot doit être capable de détecter dans son environnement des éléments suffisamment caractéristiques pour lui servir de points de repère (dans la suite, nous appellerons de telles entités *amers* ou *caractéristiques de recalage*). La mise en correspondance d'entités 3D perçues plusieurs fois au cours du mouvement permet d'estimer le déplacement du robot dans le repère de l'environnement, et donc de corriger l'estimée de position fournie par les capteurs proprioceptifs. Les capteurs proprioceptifs et extéroceptifs jouent ici un rôle complémentaire du point de vue de la localisation du robot. L'estimation trajectométrique permet de diminuer la complexité de la phase de mise en correspondance d'amers; par ailleurs, plus cette estimée est bonne, et plus le risque d'aboutir à des appariements erronés d'amers est faible. Le *modèle de localisation* contient l'ensemble de ces entités de recalage, connues a priori (*balises*), ou découvertes au cours de la mission. La fonction *localisation* peut être activée sur requête du système de Supervision, si l'incertitude sur l'estimée courante de la position du robot devient trop grande ou si des amers de localisation peuvent être perçus.

- **Navigation:** Cette fonctionnalité est insérée dans la boucle générale exécutée par le robot pour explorer de manière autonome son environnement: acquisition sensorielle / modélisation de l'environnement / génération de trajectoire / exécution de trajectoire.

Il faut ici construire un modèle de l'environnement adapté aux caractéristiques du robot (sa taille ou encombrement, ses capacités de locomotion, ...).

- **Interprétation:** Ce type de tâche peut être activée de manière asynchrone par le système de supervision du robot, lorsqu'on veut détecter et reconnaître dans la scène une entité prédéfinie (dont on connaît souvent la position et une description approximatives), ayant une quelconque pertinence avec la mission accomplie.

Dans ce travail, nous nous sommes intéressés tout particulièrement au problème de la localisation du robot, dans un environnement semi-structuré ou complètement non-structuré. Ce problème est indissociable de celui de la modélisation de cet environnement: d'une part il faut déterminer comment modéliser les amers de localisation détectés dans l'environnement, d'autre part une connaissance précise de la position du robot dans son environnement est nécessaire à une construction incrémentale cohérente des modèles de cet environnement. Nous avons principalement travaillé sur la construction incrémentale de modèles géométriques de l'environnement, quelquefois entièrement dédiés au recalage de la position du robot.

Dans cette étude, nous faisons l'hypothèse raisonnable que l'environnement est *statique*, et a les propriétés des *solides indéformables*. Nous supposons également que le robot évolue dans un environnement suffisamment peu étendu pour que la référence de verticale puisse être considérée comme constante (*planète plate*). Par ailleurs, les acquisitions sensorielles à l'aide de capteurs extéroceptifs sont effectuées *à l'arrêt*. Cette hypothèse permet d'améliorer grandement la connaissance de la verticale lors des acquisitions (cf. chapitre 2), et simplifie l'acquisition d'images. En effet, le temps d'acquisition de données extéroceptives est tel, que faire une acquisition sensorielle en mouvement nécessiterait, pour pouvoir exprimer l'ensemble des mesures dans un même repère, de corriger les différents points de mesure, en fonction de la position du robot à laquelle chacun d'eux aurait été acquis. L'acquisition de données en mouvement pose également le problème de la correction des perturbations des mesures dues aux vibrations qui apparaissent au cours du déplacement.

Dans le paragraphe suivant, nous présentons la méthodologie habituelle pour réaliser la fonction *localisation*, et montrons combien *localisation* et *modélisation* sont fortement liées.

## 1.3 La localisation

### A/ Méthodologie générale

La méthodologie habituellement utilisée dans les systèmes de recalage de la position d'un robot mobile [Ayache 87a] [Moutarlier 91] met en oeuvre les grandes étapes suivantes, lorsqu'une acquisition sensorielle vient d'être effectuée:

1. Construction d'un modèle instantané de localisation: une technique de segmentation des données sensorielles appropriée au type d'amers que l'on veut détecter extrait les caractéristiques recherchées.

2. Mise en correspondance entre les caractéristiques contenues dans le modèle instantané et celles contenues dans le modèle local de localisation. On obtient un ensemble d'appariements.
3. Réestimation de la position et de l'orientation du robot (i.e. *recalage*) à partir des appariements obtenus.
4. Mise à jour du modèle local de localisation; les caractéristiques déjà existantes et de nouveau perçues sont mises à jour, et celles qui n'avaient jamais été perçues auparavant (caractéristiques du modèle instantané non appariées avec des caractéristiques du modèle local) sont incorporées dans le modèle local.

La plupart des méthodes de localisation que nous avons développées et que nous présentons dans les parties II et III impliquent la construction incrémentale d'un modèle de l'environnement contenant les représentations de l'ensemble des amers utilisés pour le recalage de la position du robot. L'estimation du mouvement de celui-ci et la fusion des primitives contenues dans les modèles de localisation se font au moyen d'un outil de filtrage numérique basé sur le *Filtre de Kalman Etendu (FKE)*, développé au LAAS par Moutarlier et Chatila [Moutarlier 89b]. Etant donné un ensemble d'appariements  $(p_i, P_i)$  entre des caractéristiques  $p_i$  (contenues dans le modèle instantané et exprimées dans le repère robot) et  $P_i$  (contenues dans le modèle local et exprimées dans le repère d'environnement), l'équation de mesure qui lie deux primitives  $p_i$  et  $P_i$  est:

$$\mathbf{f}(\mathbf{P}_i, \mathbf{T}, \mathbf{p}_i) = \mathbf{T}(\mathbf{p}_i) - \mathbf{P}_i = \mathbf{0} \quad (1.1)$$

où  $\mathbf{T}$  = matrice de transformation homogène. Chaque caractéristique, dont la position est connue avec une certaine incertitude, doit être représentée par un vecteur des estimations de ses paramètres et la matrice de variances / covariances associée.

Certaines méthodes présentées dans la partie III de ce mémoire, par contre, nécessitent la mise en oeuvre de méthodes de mise en correspondance et de filtrage adaptées; nous pensons essentiellement là aux techniques de localisation reposant sur l'utilisation de caractéristiques de terrain (amers non structurés), pour lesquelles nous avons développé une méthode faisant coopérer divers types de modèles du terrain.

Dans tous les cas, il convient au fur et à mesure que l'on construit incrémentalement le modèle de localisation, d'assurer la cohérence de celui-ci. Nous devons notamment éliminer les artefacts qui peuvent apparaître, essentiellement lors de la procédure de segmentation des données sensorielles, et surtout lorsque les bruits de mesure sont importants, ce qui est souvent le cas en milieu extérieur. Nous éliminons donc au cours de la construction incrémentale des modèles de localisation les primitives qui ne correspondent pas à des caractéristiques fiables. Connaissant les estimations du déplacement du robot et leurs incertitudes, ainsi que les modèles des capteurs utilisés et les paramètres d'acquisition (résolution et champ de vue de la caméra laser 3D), nous pouvons prédire si les amers connus vont être perçus ou non

lors d'une nouvelle acquisition sensorielle. Pour chaque amer, nous calculons alors le rapport entre le nombre de détections effectives, et le nombre de détections prédites. Nous éliminons ceux pour lesquels ce rapport est inférieur à un certain seuil.

## B/ Localisation et modélisation incrémentale

Le recalage de l'estimée trajectométrique de la position du robot au cours du mouvement permet de connaître celle-ci avec une plus grande précision. Ainsi, on peut réaliser une construction incrémentale plus fiable des modèles de l'environnement, grâce à une meilleure connaissance des positions auxquelles les acquisitions sensorielles sont effectuées.

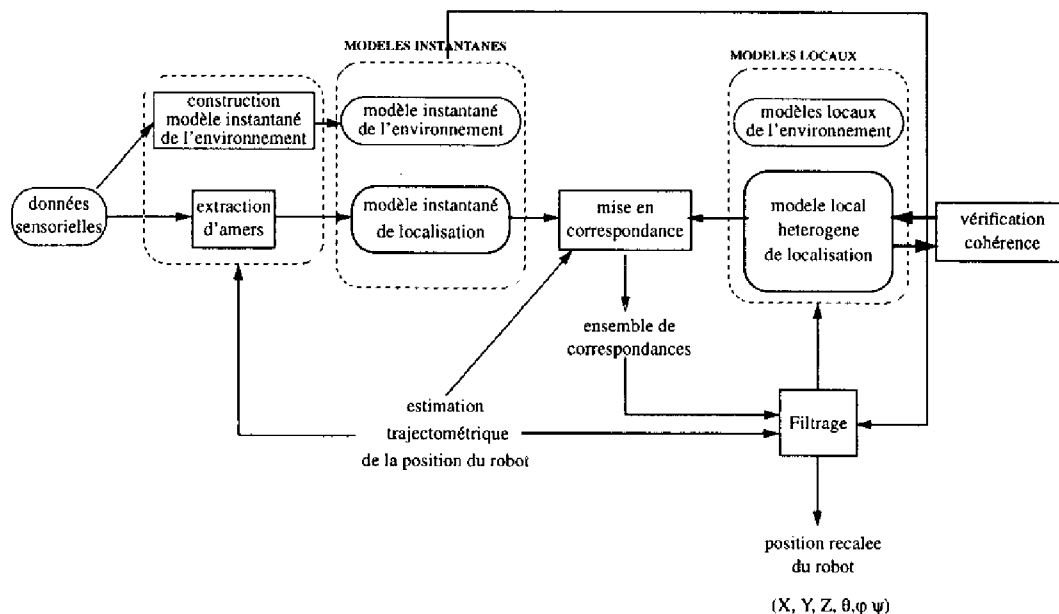


Figure 1.1: Localisation et modélisation incrémentale

La figure 1.1 résume les interactions entre localisation et modélisation.

L'utilisation de la position recalée permet notamment d'exprimer dans un même repère lié à l'environnement (repère *local*) les modèles instantanés et locaux de l'environnement. Ceci rend alors possible la *construction incrémentale* des modèles locaux de l'environnement. En particulier, si on s'intéresse à la construction d'un modèle de localisation contenant une représentation des amers rencontrés dans l'environnement, on peut réaliser la *fusion* des amers de nouveau perçus et l'*agrégation* des amers perçus pour la première fois.

## 1.4 Le contexte applicatif

Notre travail s'est déroulé dans le cadre stimulant de plusieurs grands projets. Nous présentons ici les grandes lignes de ces différents projets, et soulignons leurs principaux objectifs et les applications visées.

### 1.4.1 Le projet AMR

Le projet européen Eureka EU-18 AMR<sup>1</sup> a débuté en 1987. Ce projet très ambitieux est mené par des entreprises industrielles et des laboratoires de recherche en France (**Framatome, Matra Marconi Space, Technicatome, LAAS-CNRS, ...**), en Italie (**Italrobot, ...**) et en Espagne (**Casa**). L'objectif est de définir, concevoir et développer des robots mobiles d'intervention pour des applications de sécurité civile. Parmi les principales applications prévues, citons la lutte contre les incendies, l'intervention après un accident nucléaire ou chimique. Le robot mobile doit être capable de se diriger vers des bâtiments endommagés, puis d'y pénétrer et de se diriger vers le lieu de l'incident. Le robot opère donc dans des environnements que nous appellerons de type semi-structuré (on peut compter sur la présence d'éléments de type structuré comme des murs, couloirs, portes, mais certaines zones peuvent être détruites, et le robot peut avoir à se déplacer sur un terrain accidenté, non structuré – comme un tas de gravats – si les amers de type structuré ont localement disparu).

Dans ce projet, le groupe RIA du LAAS a notamment fortement contribué à la conception, l'implémentation puis l'intégration du sous-système de Perception (*PER* selon la terminologie AMR; la figure 1.2 résume les interactions entre les différents types de modèles). Une démonstration clôturant la phase 2 du projet, et qui a permis de valider bon nombre de concepts concernant l'autonomie du véhicule, la technique de téléopération adoptée et les performances de l'ensemble du système, a eu lieu en Juillet 1992 sur le site expérimental de Matra Marconi Space, sur le démonstrateur ADAM<sup>2</sup> [Lacombe 91], dont nous présentons les caractéristiques plus loin.

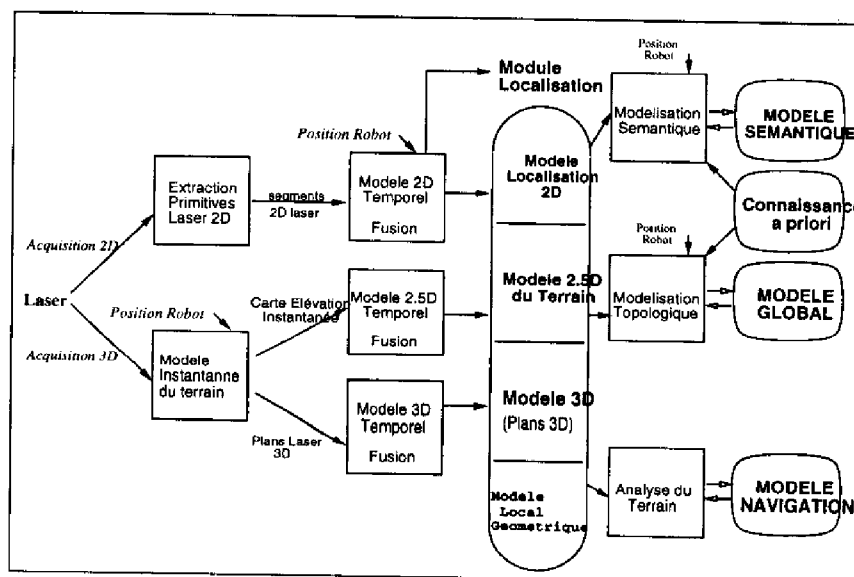


Figure 1.2: Système de Perception (PER): Services de Modélisation.

<sup>1</sup> Advanced Mobile Robots

<sup>2</sup> Advanced Demonstrator for Autonomy and Mobility

Le sous-système PER dialogue et échange des données avec les autres sous-systèmes (il transmet par exemple la carte de navigation au sous-système *PAG*<sup>3</sup>, chargé de la génération de trajectoires) et répond aux requêtes envoyées par le sous-système *SUP*<sup>4</sup> notamment. Le robot peut évoluer sous les trois modes opérationnels suivants:

- **Mode Téléopéré:** Le robot est commandé directement par un opérateur, et une connaissance a priori de la zone opératoire est utilisée; des fonctions de surveillance perceptuelle locale restent activées.
- **Mode Supervisé:** Des primitives de haut niveau, comme "Goto (X,Y) avec évitement d'obstacle", "Suivi de mur", ... sont utilisées pour commander le robot. La perception est ici utilisée pour l'exécution des mouvements asservis.
- **Mode Autonome:** Ce mode est nécessaire lorsque les communications sont coupées ou impossibles, ou si le robot doit naviguer dans des zones inconnues ou partiellement connues, ou pour lesquelles la connaissance a priori n'est pas suffisamment précise ou n'est plus fidèle (zones endommagées par exemple). Le robot doit alors pouvoir, de manière autonome, construire incrémentalement le nouveau modèle de l'environnement, pour permettre la génération et l'exécution de trajectoires.

#### A/ Modèles de l'environnement

Plusieurs niveaux de représentation, correspondant aux contraintes imposées par la réalisation des différentes tâches énumérées ci-dessus, ont été définis:

- **Un modèle 2D**, ou encore *modèle de localisation*, utilisé pour le recalage de la position du robot [Moutarlier 91, Fillatreau 93a].
- **Un modèle "2.5D"**, basé sur une discrétisation du terrain selon une grille régulière. En chaque point de la grille, la *carte d'élévation* [Nashashibi 92], fournit l'altitude du terrain (information numérique), et la *carte de navigation* fournit une étiquette symbolique donnant une information sur la navigabilité (navigable, obstacle, inconnu, ...), en fonction des capacités de locomotion du robot. Ce modèle est une des représentations centrales de l'environnement dans le projet AMR.
- **Un modèle 3D**, composé de l'ensemble des primitives planes contenues dans la scène [Nashashibi 93c], qui peut être utilisé pour tester la cohérence et la consistance des modèles de l'environnement, pour affiner certaines de ces représentations, ou pour des processus d'interprétation sémantique de la scène.

#### B/ Localisation du robot

Le choix des amers de localisation (et par là-même des primitives utilisées pour les représenter), par rapport auxquels s'effectuera le recalage de la position du robot, est lié au type

<sup>3</sup>Path Generation

<sup>4</sup>Supervision

d'environnement dans lequel le robot évolue. Pour le type de missions visées par le projet AMR, des amers de type structuré seront souvent présents dans l'environnement du robot, et notamment:

- Des arbres (à l'extérieur des bâtiments), des lignes verticales ou des colonnes (à l'intérieur) [Fillatreau 93a]. Les primitives utilisées pour représenter ce type d'amers seront des points 2D, donnant la position  $(x, y)$  des amers verticaux.
- Des murs (ou des obstacles verticaux analogues) aux abords ou à l'intérieur de bâtiments. Les primitives utilisées sont alors des droites ou des segments de droite (représentant les projections au sol de murs verticaux) [Moutarlier 89b, Moutarlier 89a].

Le recalage effectué est 2D, position  $(X, Y)$  et lacet  $\theta$ . Il faut ensuite appliquer une technique spécifique pour faire le recalage de la troisième coordonnée de position  $Z$  [Nashashibi 92, Fillatreau 93c]. La verticale est supposée connue avec une bonne précision (à l'aide d'une centrale inertielle ou d'inclinomètres, cf. chapitre 2).

Si des amers de type structuré ne sont disponibles qu'en petit nombre, ou ont disparu localement, il faut se tourner vers des techniques de localisation plus spécifiques d'environnements *non-structurés*. Dans le cadre du projet AMR, deux approches ont été envisagées:

- D'une part, l'utilisation d'amers de type *pics d'élévation* (ou sommets), qui sont des maxima locaux d'altitude [Kweon 90]. Les primitives utilisées pour représenter ces amers peuvent être des points 3D. L'utilisation de tels amers peut permettre de recalculer la position  $(X, Y, Z)$  et l'angle de lacet du robot.
- D'autre part, l'utilisation de techniques de *corrélation*, qui peuvent mettre en correspondance des portions de terrain accidenté en utilisant des critères de parallélisme entre les surfaces représentant le terrain (ou sol accidenté). Ces techniques sont souvent très lourdes et nécessitent une bonne estimation préalable de la position du robot. Leur mise en oeuvre suppose par ailleurs *parfaitement connue* l'attitude du robot; seule la position  $(X, Y, Z)$  du robot est recalée.

#### 1.4.2 Le projet VAP

Le projet *VAP*<sup>5</sup> [Giralt 92], projet d'études entrepris par le CNES en 1989, a pour but de développer les concepts de base pour un robot mobile d'exploration planétaire. La première phase du projet a été menée par le CNES en association avec le groupement scientifique RISP, consortium de laboratoires appartenant au CEA, au CNRS, à l'INRIA et à l'ONERA.<sup>6</sup>

Les sous-systèmes qui nous intéressent plus particulièrement (système de localisation et système de perception) doivent satisfaire des contraintes spécifiques à une mission planétaire,

<sup>5</sup>Véhicule Automatique Planétaire

<sup>6</sup>CEA: Commissariat à l'Energie Atomique - CNRS: Centre National de la Recherche Scientifique - INRIA: Institut National de Recherche en Informatique et Automatique - ONERA: Office National d'Etudes et de Recherches Aérospatiales.



et en particulier martienne. Les capteurs de perception embarqués sont un ensemble de 2 ou 3 caméras CCD, et un télémètre laser. La localisation du robot doit être effectuée par fusion de données acquises extérieurement au robot (balises, position donnée par l'orbiteur, ...) et de données acquises par celui-ci (repérage d'étoiles, moyens inertiels, caméras, ...). Différents niveaux de précision pour la localisation sont requis en fonction de la tâche effectuée (construction d'une carte globale grossière, navigation et identification d'amers, génération de trajectoires et évitement d'obstacles).

### 1.4.3 L'expérimentation EDEN

L'expérimentation *EDEN*<sup>7</sup>, menée actuellement au LAAS avec le robot mobile *ADAM*<sup>8</sup>, a pour but de démontrer un système complet de navigation autonome pour un robot mobile évoluant dans un environnement naturel [Lacroix 93]. La tâche canonique est "ALLER À (un amer)", dans un environnement naturel, inconnu a priori, et découvert incrémentalement par le robot. L'amer est un objet dont on connaît un modèle et la position approximative dans l'environnement.

La navigation autonome en environnement naturel pose des problèmes particulièrement complexes. L'environnement est varié, et peut localement être fortement ou complètement non structuré. Les représentations ou informations utilisées sont largement construites à partir des données sensorielles, qui sont souvent incomplètes, (très) imprécises ou erronées.

Le comportement du robot doit être adapté à la nature des terrains traversés. Ainsi, trois modes de navigation ont été définis:

- Le mode **planifié 2D** est activé lorsque le terrain est essentiellement plat (ou de pente faible par rapport aux capacités de locomotion du robot). Les cartes de navigation manipulées donnent une information de navigabilité (navigable / non navigable / inconnu) sur l'environnement. Des méthodes de génération de trajectoires de type "bitmap" s'appliquent bien ici.
- Le mode **planifié 3D** est activé lorsque le robot doit traverser une zone de terrain accidenté. Un modèle précis du terrain est alors nécessaire et un planificateur 3D génère les trajectoires [Siméon 93, Dacre Wright 93].
- Le mode **réflexe** ou **réactif** est activé lorsque le robot traverse une zone de terrain essentiellement "vide d'obstacles". Le mouvement du robot est généré dans ce cas à partir de la position approximative du but courant à atteindre et les capteurs embarqués assurent une surveillance pour la détection d'obstacles.

Les représentations de l'environnement manipulées (en particulier les modèles du terrain ou des amers situés dans l'environnement) tiennent compte de la diversité des zones traversées,

<sup>7</sup>Expérimentation de Déplacement en Environnement Naturel

<sup>8</sup>ADAM, démonstrateur AMR pour la phase II, est la propriété de Framatome et de Matra Marconi Space, et est actuellement en prêt au LAAS.

ainsi que des différentes fonctionnalités à réaliser. Un modèle multi-niveaux organise la représentation de l'environnement et met en oeuvre des modèles hétérogènes (cf. figure 1.3).

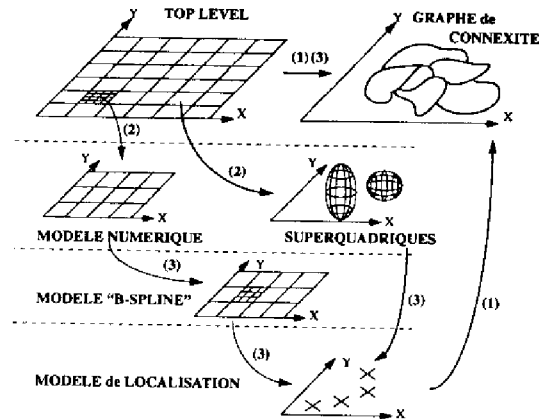


Figure 1.3: Les représentations de l'environnement dans EDEN

Le premier niveau de la représentation est une représentation de type "bitmap", basée sur une procédure de classification rapide du terrain développée par S. Lacroix [Lacroix 93], et qui permet notamment de déterminer le mode de navigation dans lequel le robot va évoluer, et de définir les sous-buts successifs utilisés par les processus de génération de trajectoires. Chaque pixel du bitmap contient entre autres une étiquette donnant le type de terrain (inconnu, plat, pentu, accidenté, obstacle). On établit un graphe de connexité entre les régions homogènes du terrain. Les zones accidentées doivent être modélisées de manière plus fine à l'aide de MNT<sup>9</sup> [Nashashibi 92] pour être utilisées dans le mode de navigation planifié 3D. Pour la localisation du robot, les représentations manipulées sont des *modèles de localisation hétérogènes*. Comme c'est le cas dans le cadre du projet AMR, ils peuvent constituer des cartes globales d'amers de localisation, de forme structurée ou caractéristique le plus souvent [Devy 92b, Fillatreau 93a] (cf. chapitre 5), rencontrés dans l'environnement du robot. Dans ce cas, ces différents amers sont modélisés à l'aide de primitives appropriées au type d'amer à représenter (les rochers peuvent être modélisés à l'aide de superquadriques [Pouzyreff 91, Betge-Brezetz 94] par exemple). Dans le cadre de VAP et d'EDEN, l'environnement est fortement non-structuré et nous avons été amenés à développer une approche plus spécifique du problème du recalage de la position du robot, où différentes représentations du terrain coopèrent. On utilise notamment une hiérarchie de surfaces B-splines, qui permet d'extraire des zones caractéristiques du terrain [Fillatreau 93b] (cf. chapitre 6). Entre les différentes représentations de l'environnement manipulées (cf. figure 1.3) apparaissent des liens de type fonctionnel (notés (1)) si l'utilisation de ces modèles dépend de l'activation de certaines fonctionnalités, spatial ((2) si on affine certaines zones d'une représentation), ou constructif ((3) si un modèle est déduit d'un autre).

Dans l'expérimentation EDEN, notre contribution se situe essentiellement au niveau du

<sup>9</sup>Modèles Numériques de Terrain

module de navigation 3D [Nashashibi 94], dont la figure 1.4 donne un schéma général.

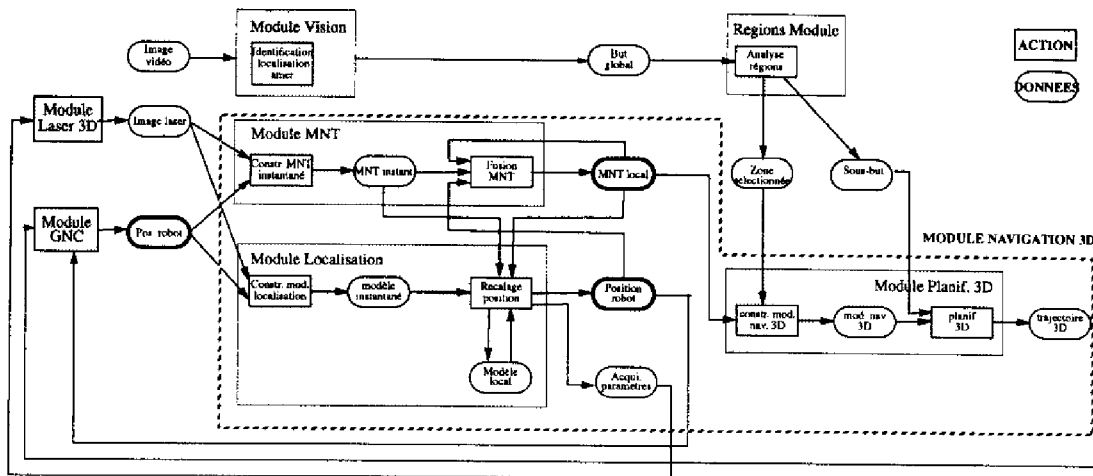


Figure 1.4: *EDEN: le module de navigation 3D*

#### 1.4.4 Les moyens expérimentaux

Nous avons pu disposer durant notre séjour au LAAS de moyens expérimentaux exceptionnels.

Le projet "HILARE" [Chatila 84], projet développé en interne au groupe RIA depuis Septembre 1977, a conduit à la construction de trois robots mobiles. La figure 1.5 présente la famille "HILARE" (de gauche à droite, et de "l'ancêtre" au "petit dernier": HILARE 1, HILARE 1.5/JUNIOR et HILARE 2), dont les architectures mécaniques, les capacités informatiques et les capteurs embarqués sont de plus en plus performants. Ces robots ont permis et permettent de démontrer des capacités d'autonomie en milieu d'intérieur, et constituent des moyens expérimentaux génériques, ouverts et évolutifs, pour valider les concepts avancés étudiés dans les projets de recherche auxquels le groupe RIA participe. La conception mécanique de HILARE 2, que nous avons plus particulièrement utilisé dans le cadre de notre travail, lui permet maintenant de sortir de notre "grande salle" d'expérimentations, et d'affronter l'environnement extérieur. Son système de capteurs embarqués comprend une ceinture de capteurs ultrasons, un télémètre laser, des caméras, et une nappe laser utilisée en proximité pour détecter des obstacles dans le sol (protubérances ou "trous") dans une zone située immédiatement devant le robot.

Nous avons également pu expérimenter durant le projet AMR sur le démonstrateur ADAM, imposante plate-forme expérimentale (dimensions: 3m x 2m x 2m !) dont voici les principales caractéristiques:

- Le châssis provient de VNII TRANSMASH, à St. Pétersbourg en Russie. L'architecture de locomotion comprend 6 roues motrices indépendantes. Le contrôle du robot a été

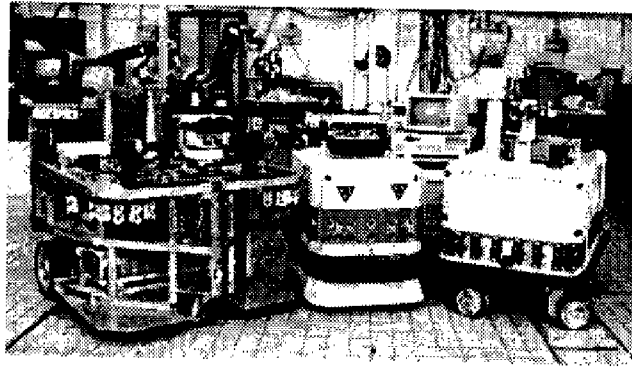


Figure 1.5: La famille "HILARE"

implémenté par Framatome à Lyon. Les capteurs de trajectométrie sont six odomètres (un par roue) et une centrale inertielle.

- Le capteur laser utilisé dans le cadre du projet AMR est le capteur ALIS <sup>10</sup>, développé par le CEA-LETI; ce télémètre laser à temps de vol permet des acquisitions d'images 128x128 à une cadence de l'ordre de 1Hz.
- Les capacités informatiques embarquées sont très grandes avec 3 cartes SPARC et six cartes 680x0, sur lesquelles les systèmes de supervision et de perception ont été implémentés par Matra Marconi Space et par le LAAS.

ADAM est aujourd'hui prêté au LAAS depuis Août 1992. Il sert de démonstrateur dans le cadre de l'expérimentation EDEN actuellement menée au LAAS dans un environnement naturel (voir figure 1.6). Le robot est actuellement équipé d'un télémètre de marque Riegl, du même type que celui d'HILARE 2, et de caméras couleur (voir figure 1.7).

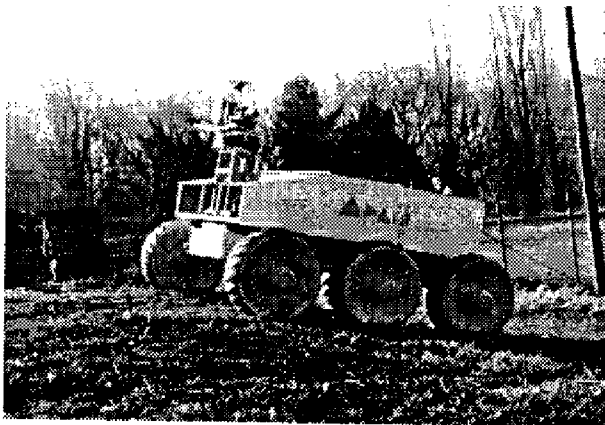


Figure 1.6: Le robot ADAM dans son environnement au LAAS

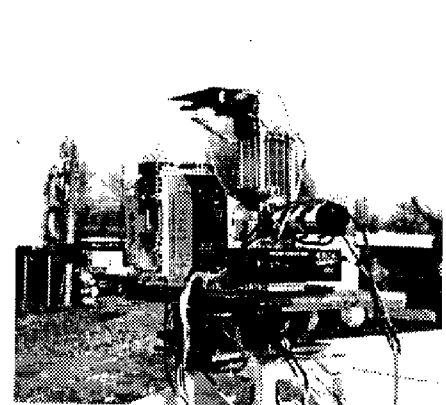


Figure 1.7: ADAM: les capteurs

<sup>10</sup> Advanced Laser Imaging System



## Chapitre 2

# Les Capteurs pour la Localisation/Modélisation

Dans ce chapitre, nous allons faire une revue des principaux capteurs utilisables en robotique mobile pour acquérir les données nécessaires à la localisation du robot et à la modélisation de son environnement. Comme nous l'avons dit au chapitre précédent, deux grandes familles apparaissent:

- Des capteurs qui permettent de réaliser la localisation “à l'estime” du robot. Il s'agit ici de déterminer, au cours du mouvement du robot, la position approximative de celui-ci en évaluant le déplacement qu'il effectue (on parle de fonction de “*trajectométrie*”). Les capteurs utilisés ici sont de type *proprioceptif* (cf. paragraphe 2.1.1).
- Des capteurs qui permettent d'acquérir des informations sur l'environnement extérieur du robot, si on veut réaliser un recalage de sa position ou si on veut construire les modèles de cet environnement. Les capteurs utilisés ici sont de type *extéroceptif*, et nous en ferons une revue rapide dans le paragraphe 2.1.2.

## 2.1 Les différents capteurs utilisables

### 2.1.1 Les capteurs pour la trajectométrie

La localisation “à l'estime” d'un véhicule terrestre peut être réalisée à l'aide de deux types de capteurs [Frappier 90]:

- des capteurs de déplacement,
- des capteurs d'attitude: cap (lacet), tangage, roulis.

#### A/ Capteurs de déplacement

Ce sont principalement les *accéléromètres*, le *radar Doppler*, et les *odomètres*. La précision de ces capteurs, qui mesurent un parcours effectué, s'exprime généralement en pourcentage

de la distance ou de l'abscisse curviligne parcourue, exception faite de l'accéléromètre, dont l'erreur sur la distance dépend aussi du temps écoulé (puisque la position découle de la double intégration de l'accélération). Typiquement, les erreurs commises sont de l'ordre du %.

Parmi ces capteurs, certains ont été très souvent utilisés en robotique mobile; nous les présentons donc très rapidement ici. Par contre, nous reviendrons plus en détail sur les capteurs inertiels au paragraphe 2.2. En effet, cette famille de capteurs regroupe des types différents, une grande variété de techniques et de technologies sont mises en oeuvre [Radix 72], et enfin ces capteurs sont d'une utilisation moins habituelle en robotique.

### **Odométrie**

Les odomètres peuvent être montés directement sur les roues, ou en "roue additionnelle". Pendant le mouvement du robot, la position du robot est mise à jour par intégration des rotations élémentaires de ces roues. La mesure du mouvement obtenue est faussée s'il y a glissement ou patinage (dans le cas où l'odométrie est effectuée sur les roues motrices notamment). Par ailleurs, elle dépend aussi de la connaissance de la structure mécanique et cinématique du véhicule, du principe de locomotion, et des imprécisions sur des paramètres comme l'entraxe ou le rayon des roues. Cette technique est couramment utilisée en robotique mobile 2D; pour ce type d'applications, elle fournit des mesures assez précises de la position et du cap. Par contre, elle est moins bien adaptée dans le cadre de la robotique mobile sur sol accidenté, car c'est l'abscisse curviligne que l'on mesure. De plus, les glissements peuvent être importants (cas d'un véhicule à chenilles par exemple). Si le problème des glissements est maîtrisé (bon odomètre indépendant par exemple), une mesure odométrique est utilisable si l'attitude du robot est connue à tout instant.

L'odométrie peut également être utilisée pour des levées de doute, pour donner des informations de vitesse nulle, etc.

### **Radar Doppler**

On dirige un faisceau électromagnétique de fréquence  $F$  vers le sol. Le faisceau reçu après rediffusion sur le sol présente un décalage de fréquence  $\Delta F$  proportionnel à la vitesse  $V$  du déplacement relatif véhicule/sol.

L'utilisation d'un radar Doppler pour la mesure du déplacement d'un véhicule se déplaçant en extérieur est envisageable, puisqu'il est insensible au patinage des roues ou des chenilles du mobile. Mais la précision se dégrade rapidement avec la diminution de la vitesse.

### **Accéléromètres**

Ils permettent de mesurer l'accélération subie suivant leur axe sensible. Nous présentons ces capteurs plus en détail au paragraphe 2.2.

### **B/ Capteurs d'attitude**

Les principaux capteurs d'attitude sont essentiellement de type inertiel: *gyroscopes et gyromètres, capteurs inertiels composites, inclinomètres, magnétomètres*. Ils sont mis en oeuvre

dans des configurations différentes selon que le but est de mesurer le cap ou la verticale (roulis et tangage). Nous présentons également ces capteurs et les systèmes de navigation à l'estime envisageables au paragraphe 2.2.

### C/ Capteurs particuliers

Notons qu'un certain nombre de capteurs *proprioceptifs* plus *spécifiques de l'application visée* peuvent également être utilisés. Nous pensons principalement aux robots mobiles "FRAS-TAR" et "FRASTAR II", démonstrateurs de locomotion avancée développés dans le cadre du projet "RAMI" <sup>1</sup>. Ces véhicules expérimentaux sont mûs par des chenilles orientables (ou *palonniers*). Chaque palonnier comporte des capteurs d'effort, qui permettent de mesurer ou calculer: l'effort vertical sur le palonnier, le couple nécessaire au "mouvement d'avance", et le moment de torsion exercé par la caisse sur le palonnier, le point de contact avec le sol, etc. Ces informations sont utiles au niveau du système de locomotion, pour garantir la sécurité du véhicule au cours de ses déplacements [Argouarc'h 89]. Elles peuvent également venir compléter avantageusement les données fournies par les capteurs extéroceptifs. Connaissant l'attitude des palonniers et les efforts auxquels ils sont soumis, on peut envisager d'en déduire des informations intéressantes, notamment sur le profil, la pente du terrain, ... Cette possibilité mériterait d'être étudiée de plus près à l'avenir.

#### 2.1.2 Les capteurs extéroceptifs

Nous présentons ici rapidement les principaux capteurs extéroceptifs qui sont utilisés en robotique mobile.

##### Ultrasons

Un capteur ultrasonore fournit des informations de distance dans une certaine région de l'espace, ou lobe, sur l'axe duquel on ramène en général la mesure. Plus le lobe est étroit, plus la résolution angulaire est bonne. Par contre, plus le lobe est large, plus le capteur couvre d'espace. On utilise donc en général plusieurs capteurs ultrasonores à lobe relativement étroit ("ceinture" d'ultrasons), la question étant de trouver un compromis entre le nombre de capteurs et leur résolution angulaire.

Ce type de capteur a été utilisé pour la modélisation d'environnement (par des "grilles d'occupation" contenant des informations probabilistes de présence d'un obstacle) [Elfes 86, Matthies 88, Elfes 89]. Crowley [Crowley 85] ou Leonard, Durrant-Whyte et Cox [Leonard 90] ont fusionné incrémentalement des données ultrasoniques incertaines.

Si un inconvénient important des capteurs ultrasonores est leur faible résolution, une "ceinture" d'ultrasons, dont les lobes sont en recouvrement, permet d'acquérir très rapidement (les données sensorielles peuvent être traitées en parallèle) des informations sur l'environnement tout autour du robot. Dans notre équipe, nous avons ainsi préféré réserver l'utilisation

<sup>1</sup>RAMI: Robotique Autonome Mobile d'Intervention, projet lancé en 1988 par Alcatel Alsthom Recherche et Framatome



de ces capteurs à la détection dynamique d'obstacles imprévus (surveillance), ou pour effectuer certains mouvements asservis (suivi de mur à faible distance, accostage, contournement d'obstacle par exemple) [Khoumsi 87, Khoumsi 88, Noreils 89].

### Caméra 2D

Ce capteur fournit des informations de luminance et permet de détecter des lignes de contraste par des techniques classiques de segmentation d'image. L'image ne fournit pas directement d'information de profondeur, et la qualité des données obtenues est très sensible aux conditions d'éclairage. Par contre, la résolution est en général satisfaisante.

Pour obtenir des données 3D, on utilise, comme il est bien connu, des techniques de stéréovision à deux caméras ou plus. Les traitements nécessaires sont notoirement assez coûteux en temps de calcul (la diminution de la résolution des données, par exemple, permet cependant de réduire de manière significative ces temps de calcul).

### Capteur laser 3D

On peut en distinguer deux types principaux, utilisant des technologies de base très différentes (alors que leurs finalités sont assez proches):

- Un premier type de capteur quelquefois appelé "nappe laser" ou "profilomètre laser". Il a plutôt pour utilité de recueillir des informations 3D sur l'environnement immédiat du robot (de l'ordre de 1 mètre au maximum). La technique utilisée est une technique de *lumière structurée* dans laquelle on projette un plan de lumière sur la scène (le sol et les objets qui s'y trouvent, en l'occurrence) et où une caméra observe la ligne lumineuse matérialisant l'intersection du plan et de la scène; ainsi, les informations tridimensionnelles sont obtenues par triangulation. Pour les profilomètres laser que nous connaissons, l'émetteur du plan de lumière est pourvu d'un degré de liberté qui permet de balayer par plusieurs lignes successives une certaine portion de scène, sans mouvement du robot. Ce type de capteur a donc un champ de vue relativement peu étendu, mais en revanche une bonne résolution. En robotique mobile, il sera plutôt utile pour recueillir des informations précises sur le sol "devant" le robot. On peut en citer quelques exemples: le profilomètre équipant le robot RAMI permet d'obtenir des coupes de terrain qui sont approximées polygonalement; les profils successifs sont ensuite triangulés pour la modélisation de terrain à proximité du robot [Samson 94]. Il existe également des capteurs de ce type à CMU<sup>2</sup> (reposant par exemple sur l'observation par une caméra d'un plan laser en rotation [Kemmonsu 93]).
- Un type de capteurs à champ de vue plus étendu (en général de l'ordre de quelques 10 mètres), et utilisant une technique de *téléométrie laser à temps de vol* ou à *déphasage* (nous présenterons leur principe plus en détail au paragraphe 2.3). La résolution de ces capteurs est en général inférieure à celle des "nappes laser". Ce sont eux que l'on baptise en principe du nom de "caméra laser 3D". Plusieurs d'entre eux ont été ou sont

<sup>2</sup>Carnegie Mellon University

utilisés dans notre équipe [Colly 91a, Grandjean 91, Moutarlier 91, Nashashibi 93a]. D'autres exemples sont le TILT (au CERT), le Perceptron ou l'ERIM (CMU) d'une portée de  $\sim 40$  m [Kweon 91a], ou ALIS (du LETI).

Nous nous sommes exclusivement intéressés à l'utilisation du deuxième type, la caméra laser 3D, dont les propriétés correspondent bien au problème posé et dont nous disposons au LAAS pour expérimenter.

Parmi les nombreux avantages de ce capteur, soulignons que c'est un capteur actif à base de rayons infrarouge, beaucoup moins sensible aux conditions d'éclairage qu'une caméra 2D. Par ailleurs, il délivre directement une information de distance. Associé à un dispositif de balayage mécanique équipé de miroirs pour la déflexion du rayon laser émis, il fournit des images de profondeur. On obtient donc directement des informations tridimensionnelles, sans traitement préalable comme c'est le cas pour la stéréovision par exemple.

Parmi les problèmes liés à l'utilisation de ce capteur, citons les problèmes de specularité (une surface vue de manière trop rasante n'est pas détectée), et l'imprécision des mesures. Un problème spécifique de l'utilisation d'un capteur laser est le problème des points de moyennage. Lorsque le rayon laser atteint le bord d'un objet (limite de discontinuité de profondeur), l'empreinte (ou *footprint* en anglais) du faisceau (qui est divergent) est partagée entre deux surfaces qui ne correspondent pas à la même profondeur, et la mesure retournée est une sorte de moyenne de ces valeurs.

### Autres types de capteurs

Il est possible, selon les applications concernées, d'utiliser d'autres types de capteurs extéroceptifs, en particulier pour assurer la localisation d'un robot mobile.

Une technique répandue est la *radionavigation*. La mise en place de balises radioélectriques dans l'environnement permet de créer une structure connue permettant de localiser un mobile. Aujourd'hui, deux grandes classes de systèmes de radionavigation peuvent être distinguées selon que les émetteurs sont au sol, ou embarqués sur des satellites en orbite (citons notamment dans cette deuxième classe le système GPS). L'avantage de la radionavigation par satellite est qu'il est plus facile d'être en visibilité directe. On peut obtenir des précisions de localisation allant jusqu'à quelques dizaines de mètres environ (certains systèmes offrent une meilleure précision encore). L'utilisation de plusieurs capteurs permet d'effectuer des mesures d'inclinométrie.

Les techniques de radiométrie conduisent à des précisions sur le recalage le plus souvent insuffisantes pour garantir une navigation sûre du robot dans un environnement très contraint; elles nous paraissent plutôt intéressantes pour un recalage global plus grossier, au niveau mission.

Les *capteurs stellaires* peuvent être également utilisés dans certaines applications. La visée d'étoiles (mesure de l'élévation et l'azimut d'étoiles par rapport à l'horizon) permet d'estimer la position et le cap d'un mobile (par exemple pour évaluer, de nuit, le déplacement d'un

véhicule d'exploration, après chaque traversée journalière, avec une précision de l'ordre de quelques dizaines de mètres). On peut également envisager d'utiliser les capteurs stellaires pour des mesures d'inclinométrie.

## 2.2 Les capteurs inertiels

Selon la classification des capteurs de localisation à l'estime adoptée ci-dessus, nous allons distinguer dans ce paragraphe les capteurs de déplacement et d'attitude. Nous allons d'abord présenter le principe et les caractéristiques des différents capteurs disponibles, puis nous allons donner quelques exemples de systèmes envisageables pour une localisation à l'estime d'un robot mobile à l'aide de capteurs inertiels.

### 2.2.1 Les capteurs "élémentaires"

#### A/ Les capteurs de déplacement: accéléromètres

Un accéléromètre est un capteur permettant de mesurer la composante suivant son axe de l'accélération du point auquel il se trouve. En pratique, cette mesure est effectuée à l'aide d'une masse d'épreuve assujettie à se déplacer suivant un axe, et sur laquelle on mesure la force nécessaire pour annuler le mouvement dû à l'accélération subie.

Parmi les nombreuses technologies rencontrées, citons l'accéléromètre pendulaire asservi et l'accéléromètre VBA (*Vibrating Beam Accelerometer*), et le micro-accéléromètre silicium. L'accéléromètre pendulaire asservi (un des plus utilisés aujourd'hui) et l'accéléromètre VBA ont des performances en précision voisines, et se distinguent surtout selon des critères de durée de vie, d'encombrement, de consommation et de sensibilité aux conditions de l'environnement. Ils mesurent des accélérations de quelques 100g avec un biais de quelques mg (pour le bas de gamme), et des accélérations de quelques 10g avec un biais de 10  $\mu$ g (pour le haut de gamme). Les micro-accéléromètres silicium mesurent des accélérations de l'ordre de 10g avec un biais de quelques 100  $\mu$ g (pour le bas de gamme), et des accélérations de l'ordre de 0,1g avec un biais de quelques  $\mu$ g (pour le haut de gamme). La stabilité du facteur d'échelle des accéléromètres haut de gamme pour ces diverses technologies est 10 à 100 fois plus grande que celle des accéléromètres bas de gamme.

#### B/ Les capteurs d'attitude

Comme nous le verrons au cours de cette revue, la mesure du cap est plus difficile et plus sensible que celle du roulis et du tangage.

#### **Les capteurs de base**

Parmi les capteurs permettant de mesurer des orientations, on distingue tout d'abord les "gyros". Suivant la nature de l'information fournie, on en distingue deux types: le *gyroscope* est un capteur d'angle et le *gyromètre* est un capteur de vitesse angulaire [Radix 78]. Les premiers gyroscopes utilisés furent des gyroscopes mécaniques, aujourd'hui les gyros laser et

les gyros à base de fibres optiques entrent sur le marché des capteurs inertiels. Les performances varient de 10 à 100°/h de dérive statique pour les gyros mécaniques ou fibre optique bas de gamme, à 0,01°/h pour un gyro mécanique haut de gamme et 0,1°/h pour un gyro fibre optique haut de gamme. Les gyros laser sont plus précis (1°/h pour le bas de gamme, à 0,005°/h pour le haut de gamme !). Les gyros laser (très performants) et à fibre optique (technique nouvelle) sont plus onéreux que le classique gyro mécanique.

On distingue également des *capteurs inertiels composites*, qui réalisent la double fonction de gyromètre et d'accéléromètre. Le principe le plus répandu est celui du "gyro sec balourdé". Un système de trois capteurs composites de ce type placés sur des axes orthogonaux peut permettre de déterminer les 3 vitesses angulaires et les 3 accélérations linéaires.

Les *magnétomètres* sont utilisés comme mesure du cap magnétique, simple et précise à quelques degrés. Ce capteur présente un temps de réponse élevé. Les mesures sont très sensibles aux perturbations magnétiques. D'une part, il faut corriger les erreurs de type fers durs et fers doux dues au véhicule lui-même (ceci est très pénalisant pour un robot équipé d'un bras manipulateur, dont la configuration varie). D'autre part, la précision dépend fortement de l'environnement magnétique du mobile (celui-ci peut varier rapidement pour un véhicule terrestre). Des procédures d'initialisation permettent de compenser une partie de ces perturbations. L'erreur est comprise en général entre 1° et 10° (sauf importante perturbation magnétique). Sur de longs trajets, l'erreur moyenne est cependant proche de zéro; la précision d'un compas magnétique est très bonne sur de longues distances.

L'*inclinaison* d'un véhicule à l'arrêt peut être mesurée avec une bonne précision, soit à l'aide de deux *accéléromètres* mesurant le vecteur gravité connu, soit à l'aide de *niveaux électrolytiques* ("potentiomètres liquides").

### La mesure du cap

Les systèmes de mesure du cap mettent en oeuvre un ou plusieurs des capteurs cités ci-dessus. On peut distinguer [Frappier 90]:

- Le magnétomètre, ou compas magnétique, ou vanne de flux, mesure la direction du champ magnétique terrestre.
- Le gyrocompas (gyroscope + compas magnétique), qui trouve de façon autonome le nord géographique et le conserve pendant le déplacement du véhicule. La précision du cap initial est comprise entre 0,1 et 20mrd, et celui-ci est gardé avec une dérive de 0,3mrd/h à 20mrd/h si le gyroscope utilisé est de qualité inertielle (i.e. avec une dérive typique inférieure à 0,1°/h) [Frappier 90]. Le gyroscope donne une mesure précise à court terme, alors que le compas magnétique fournit une précision à long terme. Ferrand [Ferrand 91] souligne le très bon rapport qualité/prix obtenu avec l'utilisation d'un gyroscope bas de gamme (pneumatique par exemple).
- Le gyromètre de lacet. Il mesure une vitesse angulaire, et conserve le cap, qui doit être

initialisé par un autre procédé (magnétomètre, balises, ...). Le cap est alors gardé avec une dérive comprise entre  $1^\circ/\text{h}$  et  $100^\circ/\text{h}$ .

Notons tout de même que pour obtenir les performances annoncées, *une bonne connaissance du tangage et du roulis est requise.*

### **La mesure de verticale**

Pour mesurer le tangage et le roulis, on cherche dans un premier temps la verticale locale du lieu, le véhicule étant à l'arrêt, à l'aide d'inclinomètres (deux accéléromètres ou éventuellement des niveaux électrolytiques), mesurant le vecteur gravité connu. L'attitude ainsi déterminée est ensuite conservée en utilisant deux axes gyroscopiques. Ceci constitue la seule manière d'obtenir des précisions correctes sur la verticale pendant des durées assez longues [Frappier 90]. Les accéléromètres ou inclinomètres, aux temps de réponse élevés, donnent des mesures précises en statique seulement. Les solutions utilisant ces seuls composants nécessitent des recalages fréquents de l'estimée de position à l'aide d'informations extéroceptives.

### **2.2.2 Les systèmes envisageables en robotique mobile**

Toutes les combinaisons faisant intervenir un ou plusieurs capteurs de déplacement et un ou plusieurs capteurs d'attitude sont envisageables. Mais les prix des composants des systèmes inertiels ont longtemps rendu ceux-ci inabordable pour la robotique mobile. Les applications plus anciennes (robotique d'intérieur, déplacements 2D sur un sol horizontal) ne justifiaient pas un investissement dans des capteurs inertiels performants, alors que l'odométrie suffisait à donner des performances de trajectométrie suffisantes. La baisse de ces prix, ainsi que les exigences posées par les nouvelles applications en robotique d'intervention (notamment sur sol accidenté) ont justifié le fait d'équiper certains robots mobiles de centrales inertielles performantes. Ainsi, on peut envisager comme capteurs d'attitude le gyromètre, le gyroscope, et la centrale d'attitude, dont les prix sont de 5kF à 200kF pour un gyroscope et de 700kF pour une centrale. Le rapport qualité-prix des équipements inertiels doit être adapté aux applications visées.

#### **A/ Robotique mobile d'intérieur ou 2D**

Dans ce type d'applications, des capteurs peu coûteux peuvent suffire. L'environnement est le plus souvent suffisamment structuré pour permettre des recalages extéroceptifs fréquents [Ferrand 91], et on peut se contenter de performances de trajectométrie plus faibles. Par ailleurs, le déplacement s'effectue le plus souvent sur un sol horizontal, ou incliné mais plan et de faible pente.

Les composants les plus intéressants pour leur prix et pour leur facilité de mise en oeuvre dans cette première catégorie d'applications sont [Ferrand 91] le directionnel ou le gyromètre un axe pour la mesure du cap, et l'horizon artificiel (ou gyroscope de verticale deux axes)

pour la mesure de verticale. Ils ont l'avantage d'être peu coûteux (en tout cas pour le matériel pneumatique) et ne nécessitent pas de logiciel de calcul, contrairement aux centrales inertielles, que nous allons présenter plus loin. Pour les déplacements, l'odométrie sera dans ces cas le plus souvent suffisante.

De nombreuses applications de robotique ont ainsi fait appel à ce type d'équipements inertiels: Ferrand [Ferrand 91] a mesuré les trois angles d'attitude et la position  $(x, y)$  du robot HILARE à l'aide d'un système composé d'un odomètre, d'un gyromètre et d'accéléromètres, obtenant de bonnes performances sur des sols où l'odométrie est fiable. Achour [Achour 87] a utilisé un gyromètre et un radar Doppler pour le positionnement 2D du robot à 3 roues VESA II. On pourrait encore citer les robots japonais VARK (équipé d'un gyrocompas) et MTV (équipé d'un gyromètre), ou le français ASPARAGUS, équipé d'un compas magnétique. Bahnu *et al* [Bahnu 90] ont utilisé 5 capteurs inertiels de faible coût (2 accéléromètres et trois gyros) pour estimer l'attitude d'un robot mobile.

### B/ Robot mobile d'intervention sur terrain accidenté

Ici, on a donc besoin d'une trajectométrie plus performante. L'odométrie seule ne peut suffire si on ne connaît pas le roulis et le tangage du robot, et si les glissements sont fréquents. Les capteurs inertiels de base n'ont souvent pas une précision donnant au robot une autonomie suffisante pour ce type d'applications. Une centrale inertielle peut alors s'avérer très utile, voire indispensable, pour la trajectométrie.

Ces centrales regroupent en général trois axes gyroscopiques, associés à deux ou trois accéléromètres pour initialiser les calculs d'orientations et mesurer les déplacements (en coopération éventuellement avec l'odométrie). Il existe deux types de configuration pour le coeur inertielle:

- Plate-forme à cardans: les composants gyro-accélérométriques sont montés sur une suspension "à cardans" possédant une liberté angulaire totale par rapport au véhicule. Les gyromètres ou gyroscopes détectent toute rotation de leur support et leurs signaux permettent d'asservir le coeur de la plate forme à une direction donnée. Les premières centrales apparues étaient de ce type.
- Plate-forme "strap-down" ou à composants liés [Radix 80]: le coeur est fixé rigidement au véhicule. On mesure les accélérations sur des axes liés au véhicule, et l'orientation de la plate-forme est déterminée à l'aide des gyroscopes.

Les centrales à cardans possèdent les avantages: d'avoir une grande précision grâce aux compensations des erreurs composants par alignement, de nécessiter des composants de moins bonne précision car on utilise une "méthode de zéro", et d'induire une faible charge de calculs. Leurs inconvénients sont: la présence de pièces mobiles qui diminuent la fiabilité, un poids, un volume et une consommation importants, le fait que seuls des gyros mécaniques peuvent être utilisés, et que les erreurs composants sont sensibles aux mouvements

coeur/cardans. Les avantages des centrales "strap-down" sont: leur grande fiabilité (moins de pièces en mouvement), leur volume et leur consommation, plus faibles que pour une centrale à cardans. Les inconvénients sont: une précision moyenne, une calibration possible mais moins aisée, et la forte charge de calculs nécessaire.

Les performances d'une centrale inertielle varient en fonction de ses capteurs de base et de leur qualité de réalisation (classe), mais la façon dont sont fusionnées les informations de nature différentes qui sont disponibles joue aussi un rôle très important. Une centrale inertielle constitue en soi un système sensoriel hybride, regroupant des capteurs complémentaires. Ainsi, le tangage et le roulis peuvent être déterminés en mesurant les composantes du vecteur gravité connu selon les axes de deux accéléromètres. Cette solution n'est pas bien adaptée à la localisation dynamique, car lorsque le robot est en mouvement, une estimation de sa propre accélération doit être connue, et les mesures sont sujettes à de grandes fluctuations à cause des nombreuses accélérations parasites subies lors d'un déplacement sur sol accidenté, et de la double intégration nécessaire. Par contre les accéléromètres permettent un bon contrôle à long terme de la verticale car les valeurs moyennes sont "bonnes". Les gyromètres, pour leur part, fournissent une très bonne mesure instantanée alors que l'estimée à long terme dérive (processus d'intégration, dérive intrinsèque du gyromètre). Vaganay [Vaganay 93] a proposé une estimation de la verticale du robot CHARLY par fusion numérique des informations accélérométriques et gyrométriques; cette technique a donné de bons résultats pour une application de robotique d'intérieur.

Précisons que les robots équipés d'un système inertiel et d'un système odométrique peuvent tirer parti de deux sources d'information complémentaires. Chaque type de capteur utilisé séparément commet des erreurs cumulatives. On sait également que la mesure de l'altitude  $z$  du robot ne peut être mesurée à l'aide de moyens purement inertiels [Faurre 71]. Plus généralement, les performances de trajectométrie peuvent être améliorées par l'utilisation d'un système hybride. Par exemple, la mesure du déplacement peut reposer sur la projection dans un repère d'environnement (grâce à la connaissance de l'attitude du robot) des mesures accélérométriques (donnant une distance après une double intégration par rapport au temps) et de l'abscisse curviligne parcourue, fournie par l'odométrie. Les mesures fournies par l'odométrie seront en général assez bonnes. Mais l'odométrie sera prise en défaut lors du patinage des roues notamment, qui se produit souvent en robotique tout-terrain. A ce moment-là, les accéléromètres, même si leurs mesures instantanées sont éventuellement entachées d'erreurs, peuvent prendre le relais jusqu'à la fin du patinage des roues. Le problème reste de détecter le début et la fin du patinage des roues (ce qui peut être fait par comparaison des mesures accélérométriques et des mesures odométriques). C'est une coopération de ce type qui est mise en oeuvre sur ADAM. Ferrand [Ferrand 92] a détecté les glissements de l'odomètre du robot HILARE par comparaison des mesures de cap fournies par le gyroscope et l'odométrie.

Pour terminer, soulignons que la mise en oeuvre d'un système tel qu'une centrale inertielle a nécessité, et nécessite encore, la résolution de nombreuses questions. La plupart du

temps, une bonne connaissance du modèle de pesanteur est nécessaire, notamment pour pouvoir fournir une estimée correcte de la verticale. Les techniques de fusion numériques d'informations multisensorielles sont à étudier avec soin. Le filtre de Kalman est ici le plus souvent utilisé [Vaganay 93, Barral 91, Bahnu 90]. Le choix des méthodes d'intégration (gyromètres, accéléromètres) révèle une grande importance vis à vis des erreurs commises [Vaganay 93, Barral 91]. L'étude de la précision de tels systèmes reste par ailleurs à notre sens un problème ouvert à cause de la multiplicité des causes d'erreurs, des natures variées de celles-ci et de leurs imbrications complexes. Les applications en robotique tout terrain induisent des difficultés supplémentaires. Ainsi, lorsque le robot se déplace sur un sol très accidenté, de nombreuses accélérations parasites (notamment verticales) sont subies par le robot. Ces conditions ne sont pas favorables, et nous devons tester les centrales inertielles dans de telles situations pour juger leurs performances. Notons cependant que des procédures de recalage dites "à vitesse nulle" permettent de corriger les dérives angulaires et de position.

### 2.2.3 La centrale équipant le robot ADAM

La centrale inertielle équipant le robot ADAM est une centrale "strap-down" constituée de deux blocs capteur. Le premier comporte deux gyromètres mécaniques deux axes, à suspension élastique à rotor accordé. Les performances annoncées par le constructeur donnent une dérive de 1 à 3°/h. Le second comporte trois accéléromètres un axe de type pendulaire asservi.

## 2.3 Les capteurs laser

### 2.3.1 Principe de fonctionnement des télémètres laser

Comme nous l'avons dit au paragraphe 2.1.2, nous nous sommes intéressés aux télémètres laser du type *temps de vol* ou à *déphasage*. Les deux techniques peuvent en effet être rencontrées:

- En mode *continu* ou *modulé*, la profondeur retournée découle de l'évaluation du déphasage entre l'onde modulée émise et l'onde modulée réfléchie. Ce type de capteur retourne à la fois la phase et l'amplitude du signal reçu après réflexion. L'information délivrée pour chaque point de mesure englobe la distance (donnée par le déphasage) et une information de réflectance (rapport des amplitudes onde émise / onde réfléchie), ce qui fait la particularité de ce type de capteur. On peut citer en particulier le Perceptron [Kweon 91a] ou le TILT.
- En mode *pulsé*, la profondeur est déterminée par évaluation du temps de vol de l'impulsion, entre l'instant d'émission et l'instant de réception, après réflexion. Parmi les capteurs de ce type, citons ALIS et les capteurs laser de nos trois robots HILARE.



Pour acquérir des mesures sur l'ensemble d'une scène réelle, il est nécessaire d'équiper le système sensoriel d'un système permettant d'effectuer un balayage suivant deux degrés de liberté angulaires (site et azimut dans notre cas). Cette déflexion est communément réalisée à l'aide d'un système de miroirs plans (HILARE) ou prismatiques (ALIS). Les performances de la "caméra laser" délivrant de véritables "images de profondeur" dépendent très fortement des performances du système de déflexion (précision de la visée angulaire  $(\theta, \phi)$  en chaque point), et de la précision de la mesure de profondeur  $\rho$  en chaque point.

Le système de déflexion doit donc être conçu et élaboré avec le plus grand soin, de manière à assurer à la fois la précision de la visée angulaire et la rapidité d'acquisition. Les systèmes mécaniques et les lois de commandes à mettre en jeu sont complexes. Il faut notamment minimiser les jeux mécaniques, maîtriser les problèmes d'asservissement en position (temps de réponse, amortissement) et éviter les vibrations à grande vitesse de balayage. Un système performant dans ces domaines serait sans doute d'un coût 5 à 10 fois supérieur à celui des très bons télémètres qui équipent nos robots HILARE. L'architecture électronique et mécanique pour les commandes d'axes de nos caméras laser a été entièrement conçue par les ingénieurs de recherche du LAAS [Bauzil 91, Bauzil 92]. Si d'autres "télémètres lasers" sont aujourd'hui capables de fournir des images de profondeur à des cadences proches de 1Hz (ALIS, TILT, Perceptron ...), le compromis adopté dans notre équipe favorise la qualité des mesures par rapport au temps d'acquisition. Ainsi, une mesure de profondeur donnée résulte de la moyenne de dix tirs laser !

### 2.3.2 Les problèmes de calibrage

La première grande étape de tout processus de perception est de définir un modèle du (des) capteurs utilisés, c'est à dire de déterminer la fonction de transfert entre les données brutes issues de la perception d'une scène réelle, et exprimées dans un repère "image"<sup>3</sup>, et les données 3D exploitables par l'utilisateur, exprimées dans un repère réel (cartésien par exemple) lié au capteur dans un premier temps (relation entre un point dans le repère image et le point réel 3D correspondant). Le détermination de ces paramètres constitue l'étape de *calibrage intrinsèque* du capteur. Dans le cas du télémètre laser, le repère "image" est en fait le repère sphérique dans lequel tout point de mesure brut est exprimé par une direction de visée donnée par le couple d'angles d'azimut et de site  $(\theta, \phi)$ , et la profondeur correspondante retournée  $\rho$ , et le problème est d'exprimer ce point dans un repère cartésien donné. Il s'agit d'établir un modèle de cette transformation, puis d'identifier ses paramètres, encore appelés *paramètres intrinsèques* du capteur.

Une deuxième phase de calibrage, dite *calibrage extrinsèque*, est nécessaire, dès que l'on équipe un système d'un ou plusieurs capteurs. En robotique, elle consiste à déterminer les transformations entre les repères cartésiens liés aux différents capteurs et le repère cartésien lié au robot, et/ou les transformations entre les repères liés aux capteurs. Dans le cas où le

<sup>3</sup>i.e. le repère "image de profondeur"; le terme "image" est ici utilisé par abus de langage, tout comme on parle "d'image de profondeur".

robot est équipé d'un système multisensoriel, en effet, il faut pouvoir exprimer l'ensemble des données sensorielles dans un même repère. De nombreuses applications de ce type ont été réalisées dans notre équipe en particulier (reconnaissance d'objet par coopération laser/caméra [Pampagnin 90, Colly 91b, Boumaza 94], fusion laser/stéréo pour la modélisation de scènes d'intérieur [Grandjean 89] [Moutarlier 89c], fusion ultrasons/caméra [Herrb 89] ...

Nous allons maintenant présenter les deux procédures de calibrage (intrinsèque et extrinsèque) choisies et mises en oeuvre, en prenant plus particulièrement l'exemple du laser d'HILARE II (un laser du même type équipe le robot ADAM pour l'expérimentation EDEN). Ce travail a été réalisé en collaboration avec F. Nashashibi. Nous en rappelons l'essentiel dans ce paragraphe 2.3.2, ainsi que dans les annexes A et B. Pour plus de détails, on se référera à [Nashashibi 93a].

### 2.3.3 Le calibrage intrinsèque: télémètre HILARE II / ADAM

Il s'agit d'établir un modèle géométrique du capteur laser, puis d'identifier ses paramètres. Un modèle propre à chaque capteur doit être établi, en fonction de son architecture mécanique et optique propre. La figure 2.1 présente le télémètre laser d'HILARE II ou d'ADAM.

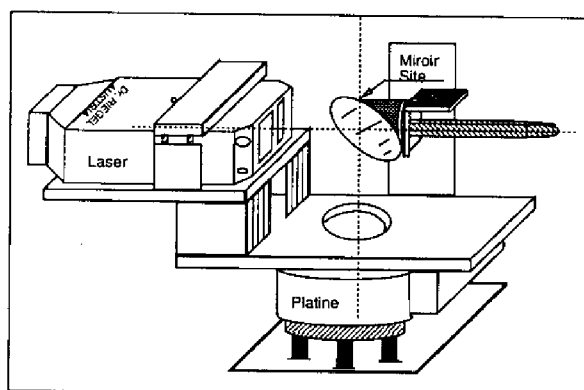


Figure 2.1: Télémètre laser du robot HILARE II

Les éléments que nous avons décidé de faire intervenir pour la définition du modèle géométrique, sont le mode de déflexion, les offsets mécaniques dus au système mécanique de déflexion, et les imprécisions de la mesure laser (en particulier, l'écart type sur la profondeur mesurée est de  $9\text{cm}^2 (\pm 3\text{cm})$ ). Nous donnons dans l'Annexe A le modèle géométrique élaboré pour le télémètre d'HILARE II, et les valeurs des paramètres obtenues après la procédure de calibrage. Le lecteur intéressé trouvera dans [Nashashibi 93a] une description plus détaillée de la procédure et des expérimentations que nous avons mises en oeuvre pour le calibrage intrinsèque, ainsi que des différentes sources d'incertitude et des résultats obtenus.

### 2.3.4 Le calibrage extrinsèque Robot-Capteur

#### A/ Position du problème

Notre but ici est de déterminer la transformation homogène 3D  $T_{rc}$  entre un repère cartésien lié au capteur et un repère cartésien lié au robot (nous avons choisi par commodité le repère odométrique, dont l'origine se trouve entre les deux roues motrices centrales dans le cas d'HILARE II (cf. figure 2.2)).

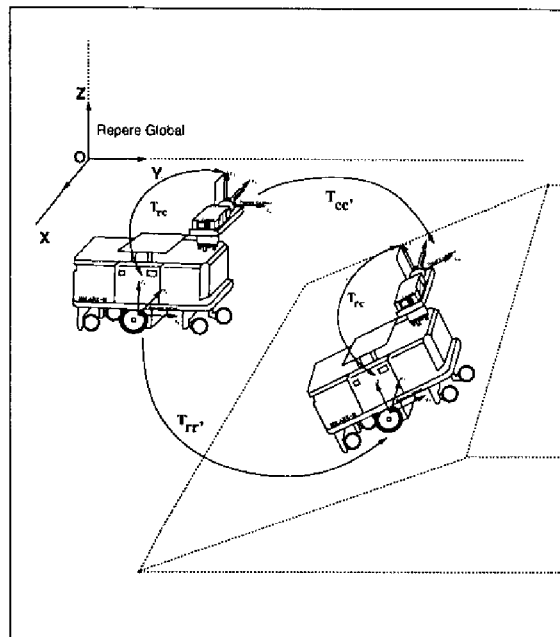


Figure 2.2: Calibrage extrinsèque robot-capteur

Ce problème a déjà été traité en pratique dans notre équipe dans le cas 2D [Grandjean 88, Colly 91a] où on cherche à déterminer la transformation robot-capteur  $(t_x, t_y, \theta)$ . L'équation B.1 conduit alors à un système de deux équations à 4 inconnues  $(t_x, t_y, \cos \theta$  et  $\sin \theta)$  avec une contrainte  $(\cos \theta)^2 + (\sin \theta)^2 = 1$ , le système étant résolu par une technique de type moindres carrés.

Pour résoudre le cas 3D, Puskorius [Puskorius 87] et Izaguirre [Izaguirre 87] ont adopté une approche de type optimisation non linéaire globale. Ces méthodes se sont avérées sensibles à la précision de l'estimée initiale et des données, et coûteuses en temps de calcul. Shiu et Ahmad [Shiu 87] ont proposé une méthode moins coûteuse mais mettant en jeu un grand nombre d'inconnues, puisque les sinus et cosinus sont traités indépendamment (comme ci-dessus). Finalement, nous leur avons préféré la méthode proposée par Tsai et Lenz [Tsai 87], qui limite le nombre d'inconnues au nombre de degrés de liberté du système, a fait l'objet d'une analyse des erreurs commises très détaillée, et est surtout plus rapide et plus simple que les autres méthodes.

### B/ L'approche de Tsai et Lenz: résultats

L'approche de Tsai et Lenz découple les phases de détermination de la rotation et de la translation. Le principe de base est de faire effectuer au robot des déplacements 3D connus (dans notre cas, grâce aux capteurs de trajectométrie: odomètre et inclinomètres pour HILARE II, odomètres et centrale inertielle pour ADAM). Si on peut reconnaître à chaque position du robot un objet ou des éléments de l'environnement permettant de déterminer le déplacement du capteur, alors la connaissance des déplacements du capteur et du déplacement du robot permettent de calculer  $T_{rc}$ . Au moins deux paires de positions du robot sont nécessaires. Dans l'annexe B, nous rappelons rapidement la procédure proposée par Tsai et Lenz, décrivons le dispositif expérimental, et présentons les résultats obtenus.

Soulignons cependant qu'il est difficile en robotique mobile d'obtenir une bonne précision sur le calibrage robot-capteur. Malgré ses avantages, la méthode de Tsai n'échappe pas à la règle. En effet, une telle procédure de calibrage nécessite une très bonne estimation des mouvements du robot et du capteur.

L'estimation du déplacement du robot est limitée par les performances du système de trajectométrie (odométrie + moyens inertiels). Ceci est d'autant plus sensible que l'attitude du robot d'une position à l'autre doit changer de manière significative si on veut avoir une meilleure précision sur les résultats de la procédure de calibrage. De même, la précision de l'estimation du déplacement du capteur est limitée par les bruits de mesure. En ce qui concerne ce point particulier cependant, la précision que nous obtenons est assez bonne car nous utilisons des plans, dont les supports sont connus avec une bonne précision car ils sont définis en général par un grand nombre de points de mesure.

En conclusion, les résultats que nous avons obtenus sont d'une précision moyenne (particulièrement en translation). Les résultats obtenus se sont avérés meilleurs pour HILARE II, pour lequel l'odométrie est performante (déplacements en milieu intérieur). Dans le cas d'ADAM, si l'orientation – et surtout la verticale – est connue avec une bonne précision, les mesures trajectométriques de déplacement sont de moins bonne qualité (on se déplace alors en milieu extérieur, il y a glissement et patinage des roues, en particulier lors des rotations).



## Chapitre 3

# Les Modèles

Nous avons vu au chapitre 1 que le problème de la localisation d'un robot mobile est indissociable de celui de la modélisation. Si nous pouvons utiliser des amers de forme simple ou structurée dans l'environnement, comment allons-nous les modéliser ? Si nous voulons utiliser des zones caractéristiques d'un terrain accidenté, comment modéliser celui-ci ?

La question que nous nous posons dans ce chapitre est celle du *type de primitives à utiliser* pour la modélisation. Nous avons séparé les objets que le robot peut rencontrer dans son environnement (et donc les méthodes de modélisation associées) en deux catégories:

- des objets de forme géométrique "simple" ou "structurée", par rapport auxquels il est naturel de penser à effectuer le recalage de la position du robot,
- des objets de forme peu ou pas structurée, tels que gravats, éboulements, terrains accidentés, pour lesquels les méthodes de modélisation sont bien sûr différentes.

Après avoir présenté une revue de la littérature concernant le problème de la modélisation de formes tridimensionnelles, nous motiverons les choix que nous avons effectués pour la suite de l'étude.

### 3.1 Revue de méthodes de modélisation 3D

#### 3.1.1 La modélisation de formes structurées

Les méthodes présentées dans ce paragraphe permettent une modélisation basée sur l'emploi de primitives relativement simples. Le point important est surtout que *ces primitives correspondent à des formes pouvant être rencontrées dans un environnement structuré*.

Il est possible de distinguer, parmi les méthodes de modélisation de formes structurées, deux grandes catégories. Dans la première, la forme est représentée par l'intermédiaire d'une surface, et dans l'autre on modélise directement un volume.

### 3.1.1.1 Les représentations “surfaci-ques”

#### A/ Modélisation par éléments de surface

Nous nous référons ici à la catégorie de représentation d’un objet ou d’une forme par ses limites (Boundary REPresentations, B-rep). Dans le cas d’objets de forme structurée, les limites de l’objet sont le plus souvent des éléments plans (autrement dit, on obtient une représentation par une surface polyédrique). Colly et Devy [Colly 91b] ont modélisé de la sorte un objet de type “satellite”.

Dans notre équipe, H. Bulata [Bulata 92] a utilisé des *quadriques* pour modéliser (après segmentation des données 3D acquises dans une scène structurée à l’aide d’un télémètre laser) des portions de surfaces appartenant à des objets non plans de forme structurée comme des cylindres, des sphères, ...

Le principe des B-rep peut être étendu à des faces non-planes et des arêtes quelconques, comme nous le verrons au paragraphe 3.1.2.1.

#### B/ Image gaussienne étendue (E.G.I.)

Ce modèle a été proposé par Ikeuchi [Ikeuchi 81]. Il est partiel et simple, mais présente un intérêt pour la reconnaissance d’objets. L’image gaussienne étendue représente la distribution des normales aux surfaces de l’objet. C’est la fonction de la sphère unité, dite sphère de Gauss, vers  $\mathcal{R}$ , qui à chaque point de cette sphère associe le réel mesurant l’aire des surfaces ayant cette direction de normale. Il n’y a correspondance biunivoque entre E.G.I. et objet que dans le cas d’un objet convexe. Ce type de modèle est adapté à la reconnaissance d’objets de formes complexes.

### 3.1.1.2 Les représentations volumiques

#### Polyèdres

Une représentation à l’aide de primitives de type polyèdre est la plus naturelle lorsqu’on est en présence d’un environnement artificiel, contenant en principe beaucoup d’objets délimités par des faces planes.

#### Superquadriques [Barr 81, Pouzyreff 91, Betge-Brezetz 94]

Une superquadrique est une surface fermée et bornée, définissant un volume. Elle est définie par six paramètres de localisation, trois de taille (pour chacun des axes) et deux de “carré-tude” (squareness), un suivant l’axe vertical et l’autre dans le plan horizontal: si ces paramètres sont proches de zéro, la forme est presque carrée, s’ils sont égaux à 1 la courbure est constante. Quelques paramètres supplémentaires peuvent encore déformer une superquadrique et on peut ainsi obtenir des formes très variées comme celles présentées figure 3.1.

Dans notre équipe, nous avons étudié la possibilité d’utiliser les superquadriques pour modéliser des “objets” (correspondant à des entités que l’on peut distinguer du sol, comme

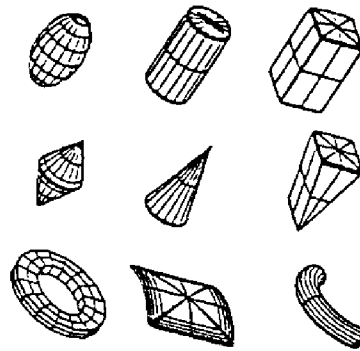


Figure 3.1: *Quelques exemples de superquadriques*

des rochers “émergeant” du terrain par exemple) présents dans l’environnement du robot [Betge-Brezetz 94]. Le but de cette étude est ensuite de construire un modèle topologique de l’environnement, comprenant des objets et un ensemble de relations topologiques entre ceux-ci. Ces travaux ont mis en évidence des problèmes d’instabilité pour certains paramètres de la superquadrique; par ailleurs, il faut identifier un grand nombre de paramètres, ce qui peut conduire à des traitements assez lourds. Pour la suite de ces travaux, il a été décidé de modéliser les objets à l’aide de représentations s’appuyant sur un nombre réduits de paramètres, dont la stabilité est plus grande: position, taille de l’objet suivant trois axes, axes d’inertie.

### Cônes généralisés

Ce modèle a été présenté par Binford [Binford 71]. Un cône généralisé est caractérisé par:

- un axe (qui peut être une courbe 3D quelconque);
- une section plane de forme quelconque;
- une règle de glissement de cette section suivant l’axe.

C’est le volume engendré par la section plane se déplaçant le long de l’axe en maintenant un angle constant avec celui-ci, selon la règle de glissement qui contrôle les modifications de la section plane. La figure 3.2 en présente quelques exemples.

Le système le plus connu mettant en oeuvre le concept de cylindre ou cône généralisé est ACRONYM, proposé par Brooks, Binford et al. [Brooks 79, Brooks 81].

### Géométrie Solide Constructive

Il s’agit ici de représenter un objet complexe à l’aide d’opérations ensemblistes (union, intersection, ...) entre des solides élémentaires (cubes, cylindres, ...). La description obtenue d’un objet (arbre dont les feuilles sont des solides de base et les noeuds des opérations ensemblistes) est bien adaptée à la C.A.O. car elle fait apparaître le processus de construction de l’objet.



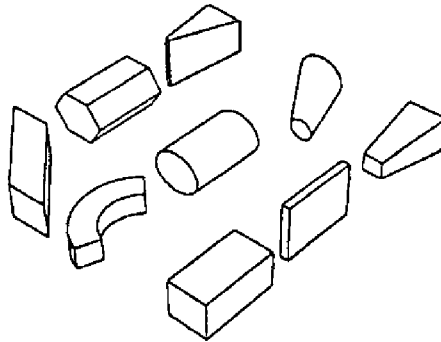


Figure 3.2: Quelques exemples de cônes généralisés

### 3.1.2 La modélisation de formes non structurées ou de terrain

Nous allons présenter ici des méthodes qui nous paraissent plus adaptées à la représentation de formes non structurées. Parmi celles-ci, de nombreuses sont issues de la CAO (telles les splines en général - B-splines, surfaces de Bézier, carreaux de Coons, ...). D'autres sont rencontrées plus traditionnellement dans la recherche en perception (les méthodes basées sur l'utilisation d'une grille, les courbes de niveau, les triangulations, les voxels, les octrees, ...).

#### 3.1.2.1 Modèles surfaciques analytiques

Les méthodes de modélisation d'objets tridimensionnels que nous allons présenter dans ce paragraphe sont principalement issues de la CAO. Elles visent à approcher avec une précision donnée un objet de forme quelconque, par une surface ou un ensemble de "patches" ou "carreaux", dont on donne une expression analytique. Toutes les méthodes abordées font intervenir une modélisation à partir d'éléments de surface. Les surfaces polynômiales notamment sont d'un usage très répandu, un polynôme par morceaux (ou une surface polynômiale par morceaux) prenant le nom général de *spline*. On trouve également souvent en CAO le terme de *surface composite*.

#### Une seule fonction analytique explicite

La représentation de la surface d'un terrain accidenté ou d'un objet de forme non structurée pourrait en théorie être effectuée par une seule fonction sur tout le domaine, mais cette solution n'est pas acceptable en pratique car les calculs induits sont très lourds et des oscillations parasites peuvent apparaître [Pavlidis 81].

C'est pourquoi on préfère en pratique représenter des formes étendues ou très accidentées par des morceaux de surfaces, raccordés entre eux selon des règles définies en fonction du besoin (continuité, continuité de la dérivée, ...).

#### "Lofted surfaces" [Faux 79, Pavlidis 81]

Il s'agit de méthodes inspirées des techniques utilisées avant l'arrivée de l'informatique, pour la conception d'objets de grandes dimensions, comme les coques de navires ou les fuselages

d'avions, qui étaient définis par une succession de sections planes parallèles entre lesquelles on faisait une interpolation à l'aide de "splines mécaniques" (il n'y avait pas alors de surface unique répondant à un ensemble de contraintes, en l'absence de mesure quantitative de la "justesse" d'une telle surface par rapport aux sections données). On fait ici une interpolation non pas entre des points, mais entre des courbes.

D'autres systèmes ont été développés par le passé. Citons par exemple [Faux 79]:

- Le système "Numerical Master Geometry" utilisant des splines pour définir à la fois les sections et les courbes longitudinales définissant la surface.
- Le système CONSURF, développé par Ball dans les années 70, et utilisant deux formes de courbes cubiques rationnelles. La figure 3.3 donne une idée de l'allure des patches (ou carreaux) obtenus.

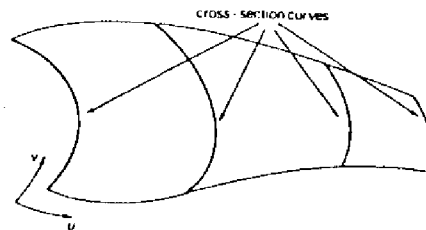


Figure 3.3: Patches obtenus avec le système CONSURF

### Carreaux de Coons

Ce sont des surfaces qui réalisent une interpolation entre les quatre courbes qui délimitent un carreau [Faux 79, Pavlidis 81, Shirai 87]. Si l'on veut assurer la continuité de la dérivée première, il faut connaître les pentes aux bornes du domaine étudié (c'est-à-dire le carreau).

### Surfaces de Bézier

Ici, on effectue une approximation. La surface ne passe pas forcément par tous les points de contrôle, mais la modification de l'un d'entre eux remet en question la totalité de la définition de la surface (contrôle global de la forme). La pente aux limites est donnée par les segments extrêmes du polygone des points de contrôle. Une surface composite, construite à l'aide de surfaces de Bézier, ne peut avoir de pente continue que si un choix adéquat a été fait pour la disposition de ces points de contrôle.

### Surfaces B-splines

Comme les surfaces de Bézier, les B-splines sont des surfaces guidées de produit tensoriel, d'un usage traditionnel en CAO. B-spline signifie "spline de base". Une surface B-splines est une surface polynômiale par morceaux qui s'exprime comme combinaison linéaire d'un certain nombre de fonction de base (cette représentation est très compacte comme nous le

verrons plus tard – cf. paragraphe 6.1 –). On peut ici garantir un certain ordre de continuité de la surface sans manipuler explicitement de conditions aux limites des carreaux qui la composent (ces conditions sont contenues dans la définition des fonctions de base). Enfin, les B-splines permettent un contrôle local de la forme de la surface.

### 3.1.2.2 Modèles surfaciques discrets, basés sur des courbes

Dans cette famille de méthodes, les surfaces sont représentées de manière discrète sous la forme de la donnée d'un certain nombre de courbes situées à leur surface. Nous présentons ici les plus courantes.

#### Courbes de niveau

Cette méthode, bien connue en cartographie, a l'avantage d'être graphiquement parlante. Le terrain est représenté par un ensemble de courbes, chacune d'elles étant le lieu des points situés à une altitude donnée. Une courbe de niveau est souvent représentée par une approximation polygonale des points qui la composent (cf. par exemple [Manaoui 88] où les courbes de niveau sont utilisées pour la navigation d'un robot mobile, et [Kweon 91b] où elles sont utilisées pour l'extraction de caractéristiques topologiques).

#### Lignes de pente

Les lignes de plus grande pente sont perpendiculaires aux lignes de niveau. On les conserve en général dans les zones où elles sont le plus représentatives du terrain. On peut notamment sélectionner les lignes de plus grande pente passant par les extrema de la courbure des courbes de niveau. Ce type de représentation peut servir à avoir un rendu visuel en cartographie; surtout, les lignes de plus grande pente peuvent servir d'amers de recalage [Kweon 91b].

#### Lignes de crête ou de vallée

On peut également représenter le terrain à l'aide d'un ensemble de lignes de crête ou de vallée qui peuvent être utilisés comme entités de recalage ou pour la navigation (réseau de vallées [Morlans 92] utilisés pour l'aide à la génération de trajectoires 3D).

### 3.1.2.3 Méthodes basées sur l'utilisation d'une grille

La modélisation par grille 2D avec attributs est une méthode très courante pour représenter le sol ou le terrain en robotique mobile. Le terrain est représenté par une projection 2D, généralement sur un plan horizontal. Chaque emplacement de la grille peut être pourvu d'un nombre quelconque d'attributs exprimant chacun une propriété de la zone correspondante. Un de ces attributs peut être la hauteur de la zone de terrain, auquel cas on parle de *carte d'élévation*. La figure 3.4 présente quelques exemples de grilles 2D pouvant servir de support à de tels modèles.

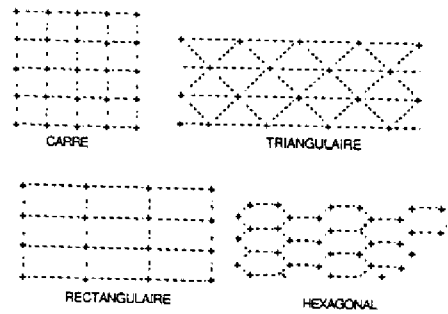


Figure 3.4: Exemples de grilles support

Ces grilles sont régulières, mais on peut bien sûr utiliser des grilles à résolution variable (à partir d'une grille initiale, on augmente la résolution dans les régions où les variations d'attributs sont trop importantes).

Diverses approches dans la même famille de méthodes peuvent être distinguées dans la littérature.

Certains travaux reposent sur l'utilisation d'une grille cartésienne. C'est le cas pour les travaux du Hughes Artificial Intelligence Center (HAIC), qui figurent parmi les premiers en date. Dans le cadre du projet ALV (Autonomous Land Vehicle), Daily *et al.* ont adopté une approche mettant en oeuvre des cartes d'élévation cartésiennes (CEC) pour lesquelles les régions à interpoler étaient modélisées à l'aide de surfaces déformables ("membranes élastiques") [Daily 88]. Les travaux de Asada [Asada 88], poursuivis avec Taniguchi [Taniguchi 91], visent à construire une carte d'élévation pour l'interprétation de scènes.

D'autres approches ont essayé de contourner les problèmes liés aux grilles cartésiennes (disparité et répartition non uniforme des données, difficulté pour détecter les zones inconnues) en représentant l'interprétation géométrique des données sensorielles en coordonnées polaires (travaux de Schalit [Schalit 92] ou, dans notre équipe, S. Lacroix [Lacroix 93] pour élaborer une procédure de classification rapide d'un terrain accidenté). Les avantages de telles approches sont que l'on évite les temps de calculs de la transformation polaire/cartésien, qu'une telle représentation est bien adaptée pour les nombreux capteurs qui suivent un mode de balayage polaire. Mais la grille polaire n'est pas invariante avec les déplacements du robot, les données sont exprimées dans des repères relatifs, ce qui pose des problèmes pour la fusion des modèles.

Rappelons enfin parmi les méthodes de modélisation par grille les "occupancy grids" (*grilles d'occupation*) proposées par Elfes, puis Elfes et Matthies [Elfes 86, Matthies 88, Elfes 89]. Les capteurs de base utilisés sont les capteurs ultrasonores. L'environnement est représenté sous forme d'une grille régulière 2D. Un modèle Bayésien permet de fournir une estimation de la probabilité d'occupation d'une cellule.

### Les MNT développés au LAAS

Nous présentons ici plus en détail la méthode d'interpolation développée par Nashashibi et Devy pour construire la carte d'élévation d'un terrain, suivant une grille régulière, à partir d'images de profondeur brutes.

La procédure d'interpolation s'applique dans notre cas aux données brutes exprimées dans un repère cartésien. Les deux opérations initiales à mener sont donc classiquement:

- l'acquisition, puis la transformation des données brutes en structure sphérique,
- la transformation des données brutes sphériques en points  $(x, y, z)$  exprimés dans un repère cartésien lié au capteur.

Le problème est ensuite de réaliser l'interpolation des données cartésiennes, qui sont éparées, réparties de manière non uniforme sur le terrain, et entachées d'erreurs de plus en plus importantes à mesure que l'on s'éloigne du capteur.

Une approche importante à ce problème est celle qui a été développée à CMU dans le cadre du projet **Mars Rover**. Kweon a proposé dans le cadre de ces travaux une méthode appelée **méthode LOCUS** [Kweon 91b] pour la construction (et l'agrégation) de cartes d'élévation, à partir de données de profondeur fournies par le *Perceptron*. Un avantage important de cette approche réside dans la résolution arbitraire qui peut être choisie pour modéliser le terrain; elle permet en effet de *calculer l'élévation en un point  $(x, y)$  quelconque*, alors que celle développée dans notre laboratoire nécessite de fixer au départ la valeur du pas d'échantillonnage choisi. La méthode LOCUS repose cependant sur le fait que le capteur utilisé a un modèle géométrique simple, et un mode d'acquisition corrélé. Cette hypothèse n'est pas légitime pour les capteurs que nous utilisons au LAAS (voir paragraphe 2.3.3).

C'est pourquoi Nashashibi et Devy ont proposé une autre approche, reposant sur l'interpolation directe des données cartésiennes. La structuration des données fournie par le mode d'acquisition du capteur est utilisée. La méthode mise au point peut être résumée dans les étapes suivantes:

- L'image cartésienne est décomposée en facettes planes élémentaires, appelées *patches*. Un patch correspond à 4 points connexes (au sens de l'acquisition) de l'image laser brute, et occupe une certaine surface sur le terrain.
- Un test est mené sur la dimension de cette zone, pour savoir si une interpolation plane peut être effectuée sur le patch. Le choix d'une approximation plane pour les facettes est justifié par la propriété de voisinage des points définissant un patch. Ce test tient compte de la spécularité de la facette, de ses dimensions, de sa position dans la scène. Il est géométrique et local, et vise en fait à détecter une discontinuité éventuelle dans la surface qu'occupe le patch (auquel cas l'interpolation n'est pas effectuée).

- Si le test autorise l'interpolation, une interpolation plane est réalisée, et l'élévation en tout point de la grille appartenant à la zone d'occupation du patch est calculée. Si le patch correspond à une discontinuité, ces points correspondent à une zone d'ombre et la valeur calculée donne simplement une borne maximale de l'élévation. Un tableau de bits fournit une information binaire sur chaque point de la carte d'élévation (connu/inconnu). Cette information s'avérera précieuse pour les travaux que nous développons au chapitre 6.

L'incertitude sur les données interpolées peut être déterminée grâce à l'utilisation du Filtre de Kalman, qui permet d'estimer la valeur de  $z$  ainsi que celle de sa variance. Les résultats ont montré que l'incertitude maximale sur  $z$  ne dépasse pas en moyenne 3cm; les erreurs les plus fortes sont commises au voisinage des discontinuités de profondeur.

Cette méthode a pour avantages la rapidité et la généralité (notamment vis à vis du capteur utilisé) des procédures (il faut environ 1 à 2 secondes pour calculer une carte d'élévation de la taille AMR – i.e. une grille de 256x256 pixels, obtenue à partir d'une image brute de 121x76 points par exemple –).

Nous présentons simplement ici un exemple de carte d'élévation construite à partir d'une image brute d'une scène d'intérieur (dans notre salle de robotique), et sur laquelle figurent notamment 3 planches de bois (sur la figure 3.5, chaque pixel  $(i, j)$  correspond à un angle de déflexion  $(\theta_i, \phi_j)$  du rayon laser, et la valeur du pixel correspond à un codage en niveaux de gris de la profondeur mesurée). La figure 3.6 présente la carte d'élévation obtenue. On distingue les 3 planches, et les zones d'ombre dans lesquelles l'interpolation n'a pas été effectuée (pour l'affichage, les points situés dans ces zones ont reçu une valeur d'élévation "non-atteignable". Il en est de même des zones situées hors champ de vue du capteur, que l'on peut distinguer en bordure de la figure, de part et d'autre).



Figure 3.5: Image laser brute: "3 planches"

Dans le cadre du projet VAP (cf. chapitre 1), les ingénieurs et chercheurs du CNES ont adopté une approche analogue, et construit des cartes d'élévation à partir de données sensorielles issues de la vision stéréoscopique [Boissier 91, Fua 91, Hotz 91].

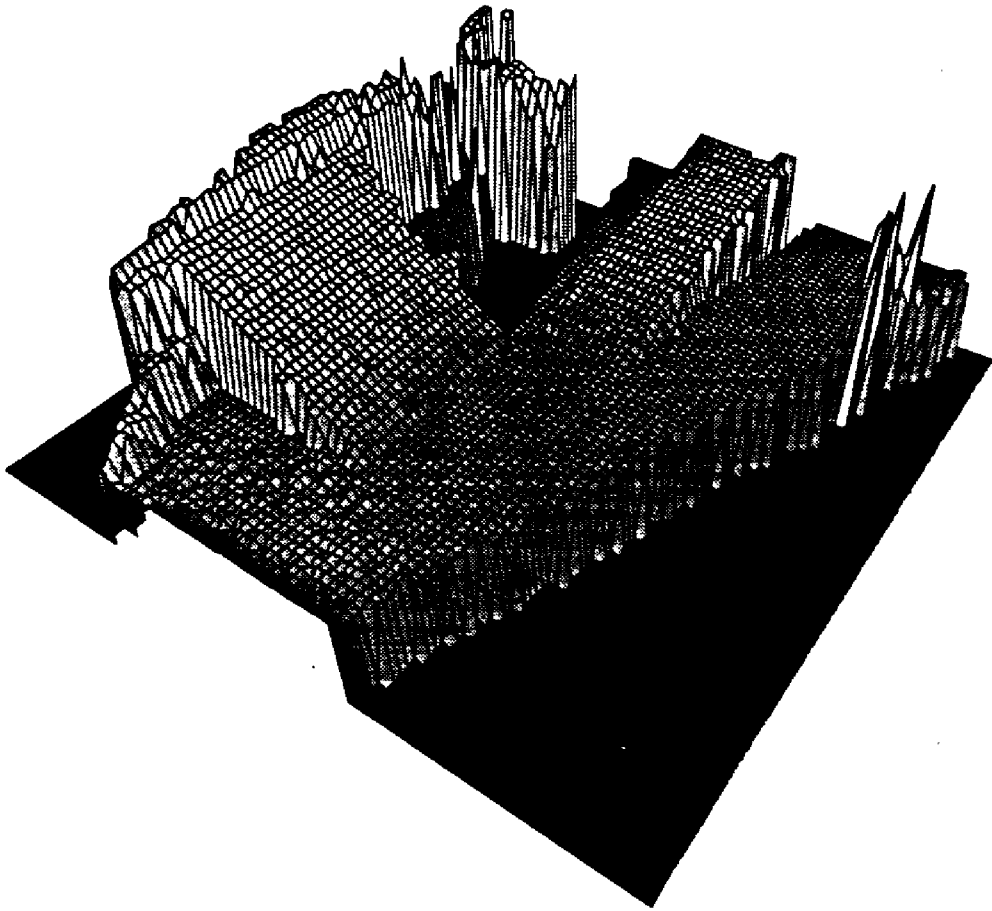


Figure 3.6: *Carte d'Elevation interpolée de l'image "3-planches".*

#### 3.1.2.4 Volumes élémentaires simples

##### Voxels

Les éléments de volume utilisés ici sont généralement des cubes de taille fixe. Le volume à discrétiser est représenté par une matrice d'ordre 3 dont chaque élément est appelé voxel (par analogie avec le terme *pixel* utilisé pour les images 2D). L'information attachée à un voxel est en général une information binaire du type présence/absence de matière dans la zone d'espace correspondant au voxel.

C'est une représentation facile à manipuler, mais la place mémoire consommée peut vite devenir prohibitive si on ne peut pas utiliser un pas de discrétisation relativement grand (et il est évidemment rare qu'on le puisse).

##### Octrees

Cette fois, le domaine à représenter est divisé récursivement en cubes de plus en plus petits. Cette division n'est effectuée que lorsque le cube considéré est partiellement rempli: tout

cube non "homogène" est découpé en huit cubes plus petits, tout cube totalement vide ou totalement plein n'est plus divisé (le modèle en octree est une généralisation au cas tridimensionnel du modèle "quadtree" en 2D). L'objet étudié est alors représenté par un arbre dont les feuilles sont des cubes de tailles non homogènes. Cette représentation est plus difficile à manipuler que la précédente et n'est pas robuste: une simple translation d'un objet dans une scène, par exemple, va en général modifier profondément l'octree correspondant. Par contre, la quantité d'informations à stocker est beaucoup plus faible que dans la représentation en voxels.

### Parallélépipèdes rectangles

Le volume de base utilisé ici est le parallélépipède rectangle. On peut citer par exemple la représentation proposée dans [Aggarwal 86].

### Sphères

Les techniques de modélisation rencontrées ici construisent une représentation volumique où la sphère est le volume de base. Cette représentation présente a priori l'avantage de faire intervenir des volumes simples à modéliser (quatre paramètres suffisent pour représenter une sphère). De plus, une sphère a des propriétés géométriques simples et fournit donc une information facile à manipuler. Il en est de même pour le modèle d'une forme bâti avec des sphères, si ce modèle demande un petit nombre de celles-ci.

Un algorithme de référence est celui proposé par O'Rourke et Badler pour construire la représentation, à l'aide de sphères, d'un ensemble de points situés sur la surface de l'objet à modéliser [Badler 79]. Il consiste à déterminer *un ensemble de sphères situées à l'intérieur de l'objet et tangentes à sa surface*. Si on limite les valeurs maximale et minimale du rayon des sphères, on peut choisir de n'approcher que la surface de l'objet à modéliser et non son volume, comme le montre l'exemple de la figure 3.7: sur la figure de gauche, l'union du volume des sphères représente bien une approximation du volume de l'objet, alors que dans le cas de la figure de droite, le modèle obtenu est "creux".

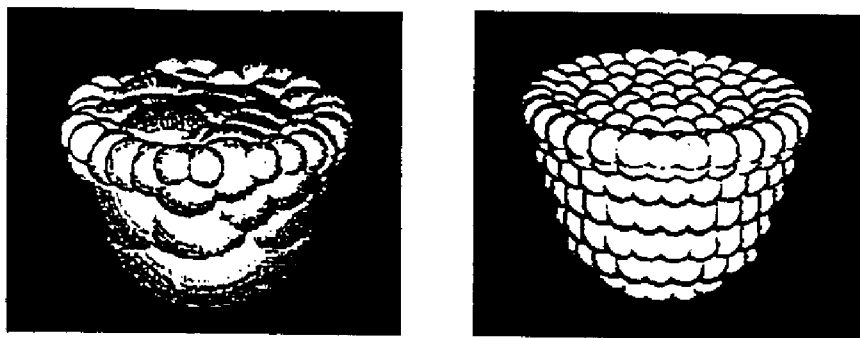


Figure 3.7: Influence du choix des valeurs maximale et minimale pour les rayons des sphères

Ce type de modèle a été adopté dans des applications robotiques, pour le calcul de distance entre objets [Tornero 91, Hamlin 92] ou la représentation des objets pour l'évitement d'obstacles [Del Pobil 92].



### Tétraèdres (triangulations)

Le volume de base utilisé ici pour discrétiser le domaine considéré est le tétraèdre (la primitive de base correspondante dans le cas 2D est le triangle). Ici, tous les points de mesure (ou les points sélectionnés parmi ceux-ci) apparaissent comme sommets de la triangulation. Le but est d'obtenir les triangles (ou les tétraèdres) les plus réguliers possibles. Les algorithmes habituellement utilisés sont la triangulation de Delaunay et l'algorithme de balayage radial.

#### 1/ Triangulation de Delaunay

Pour un ensemble donné de  $n$  points  $M_i$ ,  $i = 1 \dots n$ , le diagramme de Voronoï est un ensemble de  $n$  polyèdres convexes  $V_i$ ,  $V_i$  étant l'ensemble des points qui sont plus proches de  $M_i$  que de n'importe quel autre point  $M_j$ , soit:

$$V_i = \{P : \forall j, 1 \leq j \leq n, d(P, M_i) \leq d(P, M_j)\}$$

$d$  désignant la distance euclidienne.

Le dual géométrique (obtenu en reliant les points  $M_i$  dont les polyèdres sont adjacents, c'est à dire ont une face en commun) du diagramme de Voronoï est une triangulation sur le domaine  $M_i$ , appelée "Triangulation de Delaunay". La figure 3.8 illustre la définition du diagramme de Voronoï et de la triangulation de Delaunay en 2D.

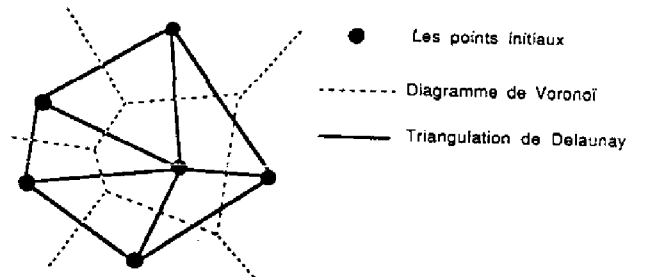


Figure 3.8: Diagramme de Voronoï et triangulation de Delaunay

On obtient une triangulation de l'enveloppe convexe des points de donnée. Si ceux-ci ne sont pas tous situés sur cette enveloppe, des techniques de "sculpture" permettent d'éliminer les tétraèdres indésirables. Parmi les principaux travaux rencontrés dans la littérature, citons [Boissonnat 87], [Watson 81], [Bowyer 81], ou [Hermeline 82] qui proposent divers algorithmes pour trianguler un nuage de points 3D à l'aide d'un ensemble de tétraèdres. Citons également Soucy *et al* [Soucy 92] qui ont utilisé une approche surfacique pour trianguler des données issues d'images de profondeur.

#### 2/ Balayage radial

C'est une méthode rapide mais qui ne satisfait pas la condition d'indépendance par rapport aux conditions initiales. Le point de départ choisi est celui qui se trouve le plus près du barycentre des points de donnée. On range ensuite les autres points de donnée angulairement

par rapport au point initial. On construit alors tous les triangles possibles à l'aide du point initial et de deux points angulairement voisins, comme le montre la figure 3.9. Il peut ensuite être nécessaire de combler les concavités extérieures (voir figure 3.10), puis de passer en revue les triangles obtenus pour obtenir les triangles les plus “réguliers” possible.

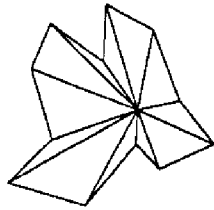


Figure 3.9: Centroïde et triangles associés

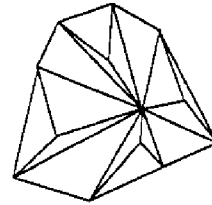


Figure 3.10: Fermeture des concavités

## 3.2 Les représentations choisies

Le choix des primitives de modélisation est lié au type de mission ou de fonctionnalité que l'on veut remplir, mais aussi au type d'objets ou de scènes que l'on veut modéliser.

### 3.2.1 Le modèle de localisation en milieu structuré

Les entités ou amers de recalage que nous souhaitons pouvoir utiliser d'une vue à l'autre sont des obstacles verticaux de type mur, et des discontinuités verticales de profondeur, correspondant aux bords d'un tronc d'arbre, d'une colonne, un coin de bâtiment, le cadre d'une porte, ...

Pour la détection d'obstacles de type “mur”, le choix effectué dans l'équipe avait été d'utiliser comme données sensorielles des coupes laser. Les primitives pour représenter les obstacles structuré de type “mur” dans le plan laser étaient des segments de droite [Moutarlier 91].

Pour modéliser les amers verticaux de type arbre, ou colonne, il nous est de même apparu naturel d'utiliser des primitives géométriques simples, du fait de la nature fortement “structurée”, ou de la forme géométrique assez simple, de tels amers. Nous avons donc décidé de modéliser les lignes de discontinuité verticales par des segments de droite, portés par des droites 3D. La représentation interne d'un tel amer (dans le modèle de localisation) sera un point 2D donnant la position  $(x, y)$  de l'amer correspondant [Fillatreau 93a] (cf. chapitre 5).

### 3.2.2 Le modèle de terrain

Le modèle de terrain que nous allons construire doit nous permettre à la fois de modéliser avec une assez bonne précision l'ensemble du terrain, ou au moins les zones de terrain les plus intéressantes, par exemple du point de vue de la localisation, qui est un des problèmes

centraux de l'ensemble de notre étude. Il doit également permettre une analyse du terrain pour déterminer des zones caractéristiques pour le recalage.

Parmi les méthodes de modélisation de terrain rencontrées, on peut distinguer deux grandes familles, correspondant aux approches *surfactive* d'une part, et aux approches *contours* d'autre part (on pourrait y ajouter les représentations volumiques, plus marginales dans ce contexte précis).

Les types de représentation qui viennent le plus naturellement à l'esprit sont d'abord des représentations du type surfactive. Par ailleurs, les représentations analytiques présentent un intérêt certain. En effet, ces modèles permettent, à partir de la connaissance des paramètres de la fonction (ou des fonctions dans le cas de surfaces composites ou de splines), d'extraire un certain nombre d'informations: dérivées, extréma, continuité, singularité, calcul d'attributs comme la courbure, . . . . Ces calculs peuvent être effectués analytiquement, ce qui est très favorable du point de vue des temps de calcul nécessaires.

Parmi toutes les méthodes de modélisation surfactive que nous avons étudiées, les *surfaces B-splines* ont particulièrement retenu notre attention, car elles possèdent de très bonnes propriétés dans le cadre de notre travail et de nombreux avantages sur d'autres techniques de modélisation. Soulignons notamment que:

- Elles conduisent à des représentations analytiques compactes, si on les compare à des représentations comme les triangulations de Delaunay ou les carreaux de Coons par exemple.
- Les surfaces B-splines permettent de traiter correctement les propriétés de localité de la surface (de manière plus satisfaisante que les surfaces de Bézier par exemple). Ces propriétés peuvent être importantes pour la modélisation incrémentale (cf. paragraphe 6.3).
- On garantit un ordre de continuité donné sans manipuler explicitement de conditions aux limites des carreaux composant la surface, ce qui n'est pas le cas par exemple pour les carreaux de Coons. Les surfaces de Bézier, de leur côté, nécessitent une gestion adéquate des points de guidage pour garantir une simple continuité au premier ordre !
- On obtient des représentations analytiques, qui permettent par exemple de générer à tout moment un MNT de résolution quelconque. Mais surtout, le modèle analytique obtenu permet de raisonner sur l'information stockée. Nous allons notamment voir au chapitre 7 comment il permet d'extraire des caractéristiques de terrain, qui seront utilisées en vue de recalculer la position du robot.

Les représentations à l'aide de B-splines ont fait ces dernières années l'objet d'un intérêt croissant chez les chercheurs du domaine de la robotique ou de la perception. Dans la plupart des cas, les B-splines sont utilisées pour représenter des courbes (on peut se référer

notamment à [Milios 92] pour l'appariement de courbes 2D, à [Guéziec 92] ou à [Kishon 90] pour le lissage et la mise en correspondance de courbes dans l'espace, à [Robert 91] pour des travaux concernant la mise en correspondance stéréo de courbes 3D, à [Parvin 91, Parvin 92] pour des problèmes d'extraction de frontières de surfaces (segmentation), ou encore à [Champleboux 92] pour la calibration). Certains travaux mettent maintenant en oeuvre des surfaces B-splines, par exemple [Wang 92a] pour la modélisation d'objets 3D, ou [Shiller 91] pour la représentation de terrain à des fins de navigation 3D.



## Conclusion de la partie I

Nous avons présenté dans cette 1ère partie les contextes scientifiques et applicatifs de notre étude. A partir d'un état de l'art concernant les principaux capteurs utilisables pour réaliser les fonctions de modélisation ou de localisation d'une part, et les principales techniques de modélisation d'autre part, nous avons justifié le choix des capteurs utilisés et le type de primitives géométriques que nous manipulerons dans nos modèles de l'environnement.

En ce qui concerne les capteurs de trajectométrie, nous avons vu que l'utilisation d'une centrale inertielle (en plus de l'odométrie) se justifie pour les applications de robotique d'intervention, lorsque l'environnement est fortement non-structuré et que le robot doit se déplacer sur un terrain très accidenté, sur lequel les performances de l'odométrie sont mauvaises. L'utilisation d'une caméra laser 3D se justifie par les nombreux avantages de ce capteur, principalement sa faible sensibilité aux conditions d'éclairage, et le fait qu'il fournit directement une information tridimensionnelle. Nul capteur n'est idéal, et celui-ci possède bien sûr également des inconvénients (spéularité, imprécision des mesures, points de moyennage notamment). Ses qualités le rendent complémentaire d'autres capteurs comme la caméra CCD.

En ce qui concerne les primitives géométriques mises en oeuvre dans les modèles de l'environnement que nous allons construire, nous avons expliqué les choix effectués, qui dépendent du type d'entités modélisées (liées au type d'environnement dans lequel le robot évolue).

Lorsque le robot évolue dans un environnement semi-structuré, et pour modéliser des obstacles verticaux de type "mur" ou des amers verticaux de type arbre ou colonne, nous avons opté pour un choix naturel de primitives géométriques simples. Les murs sont détectés à l'aide de coupes laser planes, les obstacles de type "mur" sont donc représentés à l'aide de segments de droite (modélisant l'intersection du plan de la coupe laser avec les murs physiques). Les lignes de discontinuité verticale (arbre, colonne) seront représentées par des droites 3D, la représentation interne d'un tel amer dans le modèle de localisation sera donnée par la position  $(x, y)$  de l'amer correspondant.

Lorsque le robot évolue dans un environnement non structuré, et pour modéliser le terrain (la représentation choisie doit être notamment appropriée à la tâche de *localisation*, qui est un des problèmes centraux de notre étude), nous avons choisi de mettre en oeuvre des surfaces B-splines. Il était naturel de songer à des représentations de type surfacique pour représenter la surface du terrain. Les représentations analytiques permettent d'extraire

rapidement, par un calcul direct, un grand nombre d'informations: dérivées, extrema, continuité, singularité, courbure, ... Les B-splines présentent par ailleurs de nombreux avantages: compacité, localité, continuité aisée à garantir notamment.

Selon les cas, les modèles de l'environnement construits pourront être dédiés à la localisation, ou être adaptés pour des tâches diverses (principalement la modélisation géométrique, la localisation, ou la navigation).

La fonction *localisation* en particulier, occupera une grande place dans la suite de l'étude.

Les plateformes expérimentales dont nous disposons sont équipées de capteurs nous permettant de connaître la verticale du robot avec une bonne précision. HILARE II est équipé d'inclinomètres, qui donnent en statique (le robot étant à l'arrêt) une mesure très précise. ADAM est équipé d'une centrale inertielle qui fournit une estimation assez précise de la verticale tout au long du déplacement, au moins pour les expérimentations menées au LAAS.

Comme nous l'avons vu au chapitre 2, il est beaucoup plus difficile de fournir une estimée trajectométrique précise du cap et des 3 coordonnées de position ( $X, Y, Z$ ). Les modules de localisation que nous présentons dans la suite de ce mémoire s'attachent en conséquence à recalculer ces quatre paramètres. Le recalage des six paramètres de position et d'orientation du robot n'est pas une nécessité absolue dans le cadre de nos applications: les acquisitions sensorielles sont faites à l'arrêt, et les temps de calcul pour recalculer les 6 paramètres seraient beaucoup plus importants que ceux induits par la mise en oeuvre des méthodes que nous allons présenter.

Par ailleurs, la fonction localisation doit posséder un caractère *adaptatif*. D'une part, pour un robot d'intervention, il faudra prévoir différentes méthodes de localisation, en fonction des situations diverses qu'il peut rencontrer. Dans le cadre applicatif du projet AMR par exemple, l'environnement peut être tantôt de type semi-structuré, tantôt de type non structuré; différents types d'amers doivent alors être utilisés pour réaliser le recalage. D'autre part, la méthode à mettre en oeuvre et la fréquence requise seront notamment fonction de la nature du terrain sur lequel le robot évolue, de la précision de l'estimation de trajectométrie, ainsi que de la précision requise pour le recalage, et des temps de calcul nécessaires pour réaliser celui-ci.

## Partie II

# Quand le milieu est semi-structuré





## Introduction à la partie II

Dans cette partie du mémoire, nous allons nous intéresser au module de localisation d'un robot mobile dans le cas où le robot évolue dans un environnement *semi-structuré*, c'est à dire un environnement dans lequel peuvent figurer un certain nombre d'*amers de localisation* de forme géométrique simple<sup>1</sup>. Ceci est par exemple le cas des robots mobiles pour la sécurité civile qui nous intéressent dans le cadre du projet AMR, lesquels peuvent être appelés à se déplacer aux abords, puis à l'intérieur de bâtiments accidentés, et sur un sol non horizontal ou accidenté (extérieur de bâtiments, talus, trottoirs, gravats à l'intérieur d'un bâtiment sinistré, ...).

Percevoir les mêmes entités de recalage lors de plusieurs acquisitions sensorielles différentes permet d'estimer le déplacement du robot. De nombreux auteurs ont étudié le problème du recalage/modélisation dans un environnement structuré (le plus souvent pour des scènes d'intérieur). Les primitives qui sont alors utilisées sont essentiellement des points, des segments ou des plans (cf. [Ayache 87a] pour une formalisation par Ayache et Faugeras du problème de la construction et la mise à jour de représentations basées sur ces différentes primitives, en utilisant le Filtre de Kalman Etendu pour outil de base). Lebègue et Aggarwal [Lebègue 92] utilisent la vision monoculaire; pour une scène d'intérieur, seuls les segments appartenant à des directions privilégiées (la verticale et deux directions orthogonales dans un plan horizontal) sont conservés. L'analyse d'une image donnée permet d'étiqueter les segments 2D en fonction de leur appartenance à une de ces trois directions; l'appariement de segments dans une séquence d'images permet d'estimer le mouvement du robot et les paramètres des segments 3D correspondants, qui constituent le modèle de l'environnement. Ayache et Faugeras [Ayache 87a][Zhang 91] ont construit incrémentalement un modèle de l'environnement résultant de la fusion de segments 3D stéréoscopiques [Ayache 87c][Ayache 87b]. Crowley [Crowley 90], Herrb [Herrb 91] ont fait de même par stéréovision dynamique; un "tracking" des segments dans une séquence d'images monoculaires permet d'appliquer un algorithme de mise en correspondance de complexité linéaire. Dans notre équipe, Moutarlier et Chatila [Moutarlier 91] ont construit incrémentalement un modèle de l'environnement d'un robot mobile (pour des scènes d'intérieur) à l'aide de coupes laser; les primitives de base sont des segments; pour le même type d'environnement, Nashashibi et Devy [Nashashibi 93a] ont construit un modèle polyédrique, à partir d'images de

---

<sup>1</sup>i. e. modélisables à l'aide d'un petit nombre de primitives très simples, comme des segments de droite ou des faces planes

profondeur acquises à l'aide d'un télémètre laser.

Dans cette partie, nous présentons des méthodes de localisation reposant sur la détection d'amers de forme simple (artificiels ou *naturels*) dans un environnement semi-structuré, à partir d'images laser 3D. L'une s'appuie sur la détection de lignes verticales, l'autre sur la détection d'obstacles verticaux comme des murs. Ces choix ont été discutés dans la partie I.

Nous allons construire incrémentalement un modèle de localisation, contenant la représentation de l'ensemble des amers rencontrés au cours d'une mission, en fusionnant les modèles instantanés successifs, qui contiennent les amers détectés au cours d'une acquisition sensorielle donnée. Les données intégrées sont issues de sources sensorielles et/ou de points de vue distincts, et les données brutes sont bruitées ou imprécises; l'estimation de la position du robot au cours du mouvement est elle-même entachée d'imprécisions. Nous devons par ailleurs être capables de réestimer des données entre lesquelles certaines relations ont été détectées; c'est le problème de la *fusion numérique*. Le problème de la *représentation des incertitudes*, de la *propagation* de ces incertitudes à travers les différents traitements effectués, de la *vérification de relations entre des primitives imprécises*, et de la *fusion* d'informations numériques redondantes et imprécises, est crucial pour pouvoir maintenir des représentations *cohérentes* de l'environnement. Dans le chapitre 4, nous présentons rapidement les outils utilisés pour la recherche d'appariements entre primitives, et pour la manipulation de données imprécises.

La plupart des applications classiques en robotique mobile (et la plupart de celles concernant les travaux cités ci-dessus) ont pu par le passé se satisfaire de l'utilisation d'un seul type de primitives pour modéliser les amers. En effet, pour des déplacements à l'intérieur de bâtiments par exemple, on peut toujours compter sur la détection de murs pour la localisation du robot. Dans le cadre des applications qui nous intéressent ici, le robot se déplace dans un environnement varié. Plusieurs types d'amers sont présents dans l'environnement, de manière plus ou moins dense, parfois simultanément, parfois non. Pour faire face à ces situations, et pouvoir assurer le plus souvent possible le succès de la tâche "localisation", il est nécessaire de mettre en oeuvre des méthodes de localisation basées sur plusieurs types d'amers (et donc de primitives). Plusieurs façons de faire coopérer différentes techniques sont envisageables, et nous les discuterons dans cette partie (essentiellement au chapitre 5). Faut-il utiliser toutes les primitives disponibles ? Doit-on utiliser indépendamment les différents types d'amers, et moyenner ou fusionner les résultats obtenus pour le recalage ? Peut-on, au contraire, sélectionner certaines primitives ? Doit-on utiliser les primitives hétérogènes de manière séquentielle, ou en parallèle ? Autant de questions que nous allons aborder dans cette partie.

## Chapitre 4

# Outils pour l'appariement et la manipulation de données imprécises

Le premier paragraphe de ce chapitre présente les différentes méthodes d'appariement susceptibles de nous intéresser, parmi celles rencontrées dans la littérature. Le deuxième paragraphe est consacré au problème de la manipulation de données imprécises. Nous y rappelons le formalisme choisi dans notre équipe pour la représentation des imprécisions, la propagation des incertitudes, la fusion numérique, et la détection de relations entre des primitives imprécises.

### 4.1 L'appariement

Le problème de l'appariement de primitives bruitées est un problème fondamental. C'est par mise en correspondance entre des primitives (percepts) issues des données sensorielles instantanées, avec celles contenues dans le modèle courant de l'environnement, que pourront être réestimés le déplacement du robot (et son incertitude) et les primitives constituant le modèle.

Il existe de nombreuses méthodes d'appariement modèle-perception (cf. [Mohr 88] ou [Grandjean 91] pour une présentation des principales méthodes utilisées). Nous ne présenterons ici que celles qui apparaissent comme les plus intéressantes dans le cadre de notre problématique. Elles peuvent être classées en trois catégories.

#### A/ Méthodes exhaustives

Elles sont caractérisées par le fait qu'elles testent *toutes* les contraintes entre *toutes* les primitives (d'où leur nom). On peut citer notamment ici la *recherche de cliques maximales*, les méthodes de *relaxation*, et la *transformée de Hough*.

La première de ces méthodes consiste à rechercher les *cliques maximales* d'un graphe de compatibilité des appariements (dont les noeuds sont *tous* les appariements possibles et

les arcs expriment la compatibilité entre deux appariements). Une clique maximale est un sous-graphe complet, dont chacun des noeuds est compatible avec tous les autres; la maximalité s'entend au sens de l'inclusion, et une clique correspond à une hypothèse cohérente localement. Cette technique a été utilisée dans notre équipe pour la reconnaissance d'objets polyédriques [Pampagnin 90][Colly 91a].

Les techniques de *relaxation* [Mohr 88] consistent en un processus de filtrage itératif. A chaque percept est associé l'ensemble des primitives modèle (ou étiquettes) qui peuvent lui être appariées. A chaque étiquette est associé un poids <sup>1</sup> reflétant son degré de cohérence. A chaque itération, les contraintes relatives entre les primitives sont vérifiées et les poids des étiquettes modifiés en conséquence. L'algorithme de recherche doit converger vers une situation où les poids n'évoluent plus; les poids les plus forts correspondent alors aux appariements les plus cohérents.

La *transformée de Hough*, enfin, est une technique d'analyse d'histogrammes. L'espace des paramètres est discrétisé et représenté par un tableau multidimensionnel. Un appariement possible définit un sous-ensemble de l'espace des paramètres. On recherche finalement les cases compatibles avec le plus grand nombre d'appariements. Cette méthode a été appliquée avec succès en 2D dans notre équipe [Moutarlier 91], mais en 3D les sous-ensembles deviennent complexes et ce type de méthode difficilement applicable.

Traitant d'un bloc l'ensemble des données sensorielles, ces méthodes exhaustives sont difficilement contrôlables, et nous leur avons préféré, dans notre contexte, les deux catégories de méthodes suivantes.

## B/ Méthodes de recherche arborescente

Ces méthodes construisent incrémentalement des hypothèses cohérentes, en développant un arbre dont un noeud donné correspond à un appariement possible entre deux primitives. Un chemin dans cet arbre correspond à un processus de construction d'une hypothèse regroupant un ensemble d'appariements possibles. Un nouvel appariement est ajouté s'il vérifie des contraintes intrinsèques *et* les contraintes relatives aux autres appariements déjà rassemblés. Si le test est positif, l'arbre se développe avec la recherche d'un autre appariement possible. L'arbre peut être développé *par les données* (*arbre d'interprétation*: on essaie alors d'apparier l'un après l'autre tous les percepts en suivant des relations de voisinage), ou *par le modèle* (*arbre de recherche*: chaque niveau de l'arbre correspond alors à la recherche d'une primitive du modèle dans la scène).

Par rapport aux méthodes exhaustives, les méthodes de recherche arborescentes présentent plusieurs avantages. D'une part elles garantissent à tout moment la *cohérence globale* des appariements découverts, alors que les méthodes exhaustives ne donnent que des solutions cohérentes localement (et une phase finale de vérification de la cohérence est le plus souvent nécessaire). D'autre part, elles permettent de mener une recherche séquentielle, contrôlable

<sup>1</sup>si ce poids est binaire on parle de relaxation *discrète*; s'il est réel - et compris entre 0 et 1 - , ce poids représente une probabilité, et on parle de relaxation *stochastique* ou *continue*

et souple. Elles rendent possible une focalisation de la recherche en abandonnant certaines branches de l'arbre (*tree pruning*), ou en commençant par développer les hypothèses les plus probables.

### C/ Méthodes de génération et vérification d'hypothèses

Ce paradigme est très employé en Intelligence Artificielle et en Perception, notamment pour la reconnaissance [Bolles 82][Faugeras 83][Grandjean 91]. Elles permettent une focalisation heuristique de la recherche en utilisant la localisation des hypothèses. Une première phase de *génération* construit puis localise les hypothèses à partir d'ensembles minimaux d'appariements. Une seconde phase de *vérification* vérifie séquentiellement chaque hypothèse par prédiction (cf. figure 4.1).

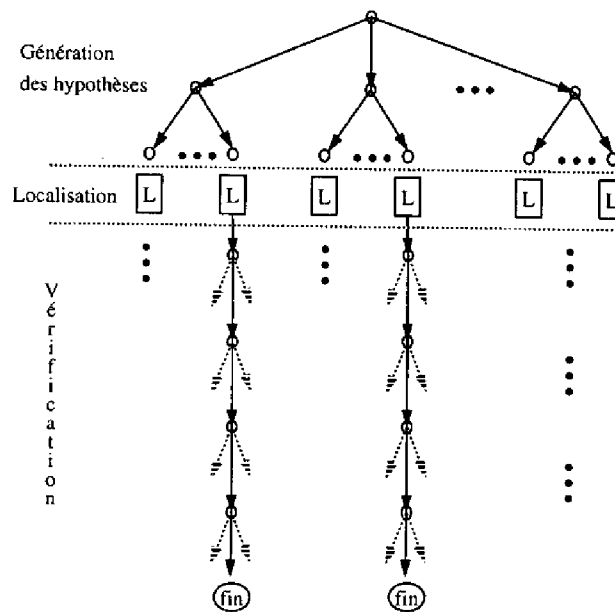


Figure 4.1: *Prédiction-Vérification*

Les méthodes d'appariement que nous avons utilisées ici entrent dans le cadre des paradigmes B/ et C/ ci-dessus.

## 4.2 Imprécisions et fusion numérique

### A/ Représentation de l'imprécision

La modélisation de l'imprécision dépend essentiellement de la nature des données manipulées, et des tâches dans lesquelles elles interviennent.

La **représentation probabiliste**, basée sur l'inférence Bayésienne, a été adoptée dans de nombreux travaux [Durrant-Whyte 87b] [Ayache 87a] [Smith 87] [Moutarlier 89b] [Nashashibi 93a]. La théorie de l'estimation est bien connue, et la représentation probabiliste

permet de s'appuyer sur un outil puissant pour la fusion: le filtre de Kalman. Le théorème de Bayes, outil de base pour l'estimation probabiliste, nécessite de modéliser les densités de probabilité. Les densités de probabilité et les distributions normales ou gaussiennes sont très largement exploitées dans la littérature [Crowley 86, Durrant-Whyte 87a].

Nous avons choisi pour notre part une représentation du premier ordre, dans laquelle une primitive imprécise  $x$  est représentée par son estimée, un vecteur aléatoire  $\hat{x}$ , et les moments statistiques du second ordre donnés par la matrice de covariance  $\sigma_x$ . Cette représentation est bien adaptée à la classe de problèmes qui nous intéressent.

$$x = \hat{x} + \epsilon_x \text{ et } \sigma_x = e[\epsilon_x \cdot \epsilon_x^t]$$

## B/ Propagation de l'incertitude

Au cours des diverses manipulations de données imprécises que nous allons être amenés à faire, nous devons propager la précision des primitives, notamment lors des changements de repère, des projections, des calculs de nouveaux attributs à partir de la détection de relations entre des primitives imprécises. La propagation des imprécisions permet en particulier de connaître la précision d'une prédiction.

Le problème général peut être formulé dans les termes suivants: étant donnée une relation  $z = f(x_1, x_2, \dots, x_n)$ , si on connaît la précision de  $x_1, x_2, \dots, x_n$ , quelle est celle de  $z$  ?

La fonction  $f$  n'est pas linéaire dans le cas général. Si on dispose d'une bonne estimée de  $z$  et des  $x_i$  (ce qui est généralement le cas en pratique), on peut linéariser la fonction  $f$  par son développement de Taylor au premier ordre autour des estimées courantes:

$$z \simeq f(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) + \sum_{i=1}^n J_i(x_i - \hat{x}_i) \simeq f(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) + \sum_{i=1}^n J_i \epsilon_{x_i}$$

les  $J_i$  représentant les matrices jacobiennes de  $f$  par rapport aux  $x_i$  ( $J_i = \frac{\partial f}{\partial x_i}(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ ). On en déduit:

$$\sigma_z = \sum_{i=1}^n J_i \sigma_{x_i} J_i^t + \sum_{i \neq j} J_i \sigma_{x_i, x_j} J_j^t$$

Dans l'expression ci-dessus, le second terme tient compte des corrélations  $\sigma_{x_i, x_j}$  possibles entre les primitives  $x_i$  et  $x_j$ .

Un cas particulier important est celui de la fonction implicite inversible; si  $f(x, z) = 0$  et si  $J_z$ , la jacobienne par rapport à  $z$ , est inversible, alors:

$$\sigma_z = J_z^{-1} J_x \sigma_x J_x^t J_z^{-1t}$$

### C/ Le filtre de Kalman généralisé (FKG)

Mis au point et développé dans notre équipe par P. Moutarlier et R. Chatila [Moutarlier 91], l'outil de filtrage numérique *YAFF*, basé sur le filtre de Kalman, peut être utilisé dans le cas de non linéarités, et en présence de bruits corrélés et colorés. Il prend en compte explicitement les corrélations entre un nombre quelconque de primitives, connues par leurs estimées  $(\hat{x}_i, \sigma_{x_i})$  reliées entre elles par une équation de mesure.

Nous donnons à l'annexe C la formulation et les principales propriétés du filtre généralisé. Il a été utilisé pour de nombreuses applications robotiques dans notre équipe (cf. par exemple [Herrb 91] pour la stéréovision dynamique, [Grandjean 91] pour la calibration inter-capteurs, la fusion multi-sensorielle et la localisation d'objets).

Il présente des avantages très importants pour la modélisation incrémentale de l'environnement (cf. par exemple, [Moutarlier 91] pour la modélisation à l'aide de segments à partir de coupes laser, ou [Nashashibi 93a] pour la construction d'un modèle polyédrique composé de faces planes à partir d'images de profondeur); nous avons utilisé ce formalisme pour la construction incrémentale des modèles de localisation (cf. chapitre 5). Pour pouvoir recalibrer la position du robot et fusionner les primitives composant le(s) modèle(s) de l'environnement de manière cohérente, il faut tenir compte des corrélations entre ces primitives et la position du robot:

- d'une part, le robot porte les capteurs, et donc la position du robot (et son incertitude) intervient dans toutes les mesures;
- d'autre part, la position du robot est recalée à l'aide des primitives contenues dans les modèles de l'environnement (par appariement entre les primitives et estimation du mouvement).

Dans le passé, certaines approches n'ont dans un premier temps pas tenu compte de ces corrélations [Faugeras 86], d'autres ont choisi de considérer un vecteur d'état global, comprenant la position du robot et l'ensemble des primitives contenues dans le modèle de l'environnement [Smith 87]. Cependant, les cas réels s'éloignent beaucoup du cas idéal (mesures linéaires, bruits blancs). La linéarisation des équations de mesure, les bruits dont on ne sait pas bien modéliser la variance, sont à l'origine de biais qui sont propagés à tout le vecteur d'état lors des fusions<sup>2</sup>. Le filtre généralisé permet de mettre en oeuvre une procédure plus robuste de recalage fusion, dans laquelle le robot est d'abord recalé pour réduire au maximum un biais éventuel, puis les primitives du modèle sont fusionnées [Moutarlier 91] [Grandjean 91] [Nashashibi 93a] (cf. chapitre 5).

<sup>2</sup>Un cas important est celui de l'odométrie: les glissements des roues et les dérapages transversaux ne peuvent être identifiés statistiquement et introduisent des biais imprévisibles, susceptibles de compromettre la cohérence globale du modèle [Moutarlier 91]



### D/ La détection de relations entre des primitives imprécises

Ce problème se pose notamment lors des procédures d'appariement entre des primitives bruitées. Soient  $n$  primitives  $(\hat{x}_i, \sigma_i)$  et la relation  $z = f(x_1, x_2, \dots, x_n)$ . Alors,  $\delta = z - f(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$  est de moyenne nulle et de variance:

$$\Gamma_{zz} = \sum_{i=1}^n \sum_{j=1}^n J_i \sigma_{x_i, x_j} J_j^t$$

Si on suppose les  $x_i$  gaussiennes,  $\delta$  est aussi gaussienne. Les surfaces d'iso-probabilité de  $\delta$  sont alors définies par une valeur de la *distance de Mahalanobis*:

$$Dm(\delta) = \delta^t \Gamma_{zz}^{-1} \delta$$

variable aléatoire suivant une loi de probabilité du  $\chi^2$  à  $q$  (rang de la matrice  $\Gamma_{zz}$ ) degrés de liberté. On peut alors déterminer un seuil tel que la probabilité que cette distance soit inférieure à ce seuil soit très élevée (95% par exemple); la probabilité que la relation  $\delta = 0$  soit vérifiée est alors élevée, et on considère que cette relation est possible.

### E/ Plusieurs capteurs: filtre centralisé ou filtre décentralisé ?

Lorsque l'on dispose d'un système multisensoriel sur un robot (laser et vision par exemple), comment convient-il de fusionner les différentes informations disponibles ? De même, si l'on dispose de plusieurs types de primitives (ou de plusieurs *capteurs logiques*<sup>3</sup>) pour effectuer le recalage du robot (ce qui sera notre cas), comment utiliser les différentes informations lors de la phase de filtrage numérique ?

Dans la plupart des travaux traitant du problème du recalage d'un robot mobile et de la construction incrémentale du modèle de son environnement, un seul type de primitives est utilisé, et une formulation classique du Filtre de Kalman Etendu (formulation *centralisée*) est utilisée. Nous nous proposons d'utiliser des primitives hétérogènes pour effectuer le recalage du robot. L'outil de filtrage numérique *YAFF* nous permet d'utiliser successivement des équations de mesure de types différents, mettant en oeuvre des primitives de types distincts. Mais comment utiliser les différentes informations lors de la phase de filtrage numérique ?

Durrant-Whyte [Rao 89] a proposé une architecture complètement décentralisée pour le problème de la fusion de données; l'architecture globale apparaît comme un réseau de "noeuds sensoriels", disposant chacun de ses propres capacités de calcul. Le point de départ repose sur une formulation *décentralisée* des équations du Filtre de Kalman Etendu. Dans l'architecture proposée, chacun des capteurs (physiques ou logiques) calcule séparément une estimée partielle de l'état, en fonction des mesures dont il dispose; ensuite, chaque noeud sensoriel

<sup>3</sup>Par l'expression *capteur logique*, nous désignons l'ensemble de la chaîne: acquisition sensorielle / description de ces données par construction d'un modèle instantané contenant les primitives extraites. A partir de l'information issue d'un même capteur, on peut construire des modèles de nature différente. Dans le chapitre 5, nous construirons à l'aide d'un capteur laser des modèles de localisation contenant des primitives de types distincts.

communiquent avec tous les autres nœuds et remet à jour une estimée globale de l'état à partir des informations reçues. Dans une telle approche, les différents capteurs sont utilisés parallèlement. Chacun dispose de son modèle de localisation, et de la position recalée du robot. Bien sûr, les équations du filtre garantissent qu'ils trouvent tous la même estimation finale de l'état. La figure 4.2 schématise cette méthode.

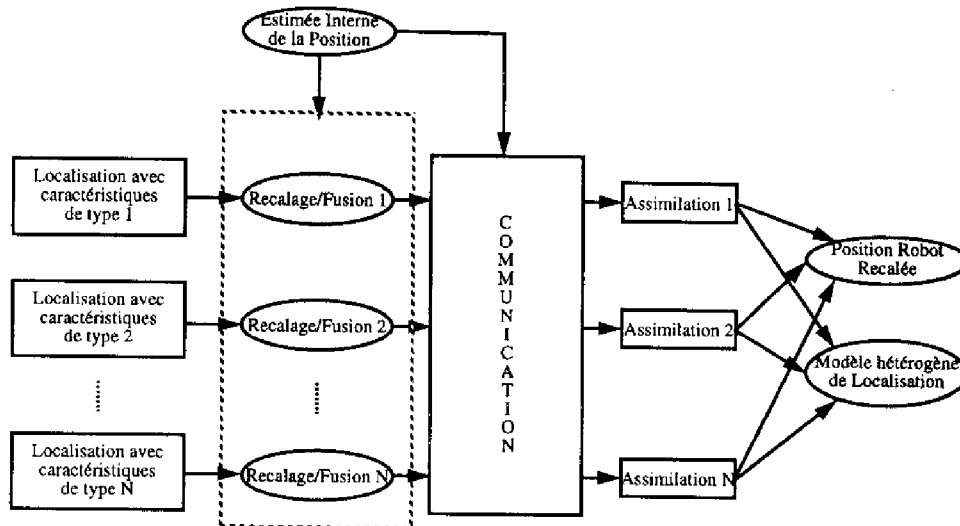


Figure 4.2: Utilisation parallèle des  $N$  modèles de localisation

Une autre approche possible consiste à sélectionner parmi tous les appariements de primitives disponibles, ceux qui sont les plus pertinents pour effectuer un recalage performant de la position du robot, puis d'utiliser les mesures sélectionnées pour recalibrer la position du robot et fusionner les modèles de localisation. Dans cette approche, une stratégie de recalage est mise en œuvre de manière séquentielle pour parvenir à un recalage optimal. La figure 4.3 présente cette deuxième approche.

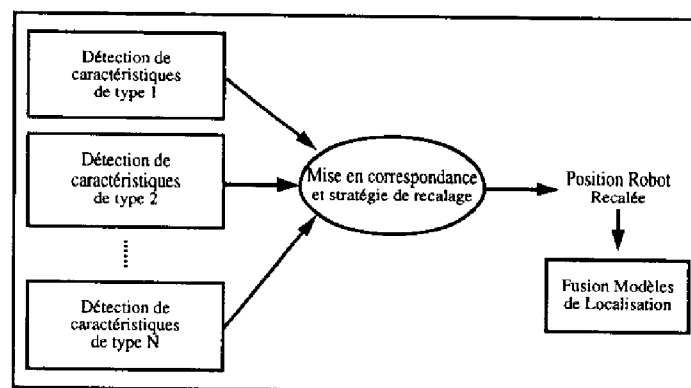


Figure 4.3: Utilisation de primitives hétérogènes

Le recalage de la position du robot et la fusion des modèles de localisation repose sur

l'utilisation du Filtre de Kalman Etendu. Les différentes mesures effectuées permettent, itérativement, de réestimer l'état du système. L'*imprécision* sur les paramètres du système diminue à mesure que de nouvelles observations sont faites.

Une question cruciale est de ne considérer que des “bons” appariements (appariements *certain*), si on veut conserver un état cohérent. Adopter une stratégie de recalage, sélectionner les primitives qui conduisent aux appariements les plus sûrs – ou *certain* – (segments les plus longs par exemple) présente alors un double avantage:

- On diminue, après recalage, le biais sur la position du robot. Ainsi, on diminue les biais dus aux linéarisations d'équations de mesure non-linéaires, car ces linéarisations sont faites au voisinage de l'état courant.
- En diminuant le biais sur la position du robot, on rend les *tests probabilistes* effectués pour la mise en correspondance de primitives *plus sélectifs*.

## Chapitre 5

# Localisation en Milieu Semi-Structuré

Dans ce chapitre, nous présentons des méthodes de localisation reposant sur la détection d'amers de forme simple dans un environnement semi-structuré, à partir d'images laser 3D. Ces méthodes ont été développées dans le cadre du projet AMR. La localisation repose sur la détection d'amers de types différents; les primitives utilisées pour les représenter seront diverses. Nous allons finalement construire incrémentalement un modèle de localisation hétérogène, contenant une description des amers rencontrés au cours du déplacement du robot.

### 5.1 Recalage 2D sur lignes verticales

Un premier type d'amers de localisation que l'on peut détecter dans l'environnement correspond aux lignes de discontinuité de profondeur verticales [Devy 92a, Fillatreau 93a]. Ces lignes peuvent correspondre à des amers présents à l'intérieur ou aux abords de bâtiments, comme le cadre d'une porte, le coin d'une pièce, l'angle d'un bâtiment, ou les bords d'une colonne, d'un poteau ou d'un tronc d'arbre.

Une première méthode de localisation mise en oeuvre pour le robot AMR a donc été basée sur la détection de ces lignes verticales. Les représentations internes (dans les modèles de localisation) de tels amers sont des points 2D, donnant leur position  $(X, Y)$ , c'est à dire la projection au sol de l'arête verticale correspondante. La mise en correspondance de telles caractéristiques permet de recalibrer la position  $(X, Y)$  et le cap du robot. Une première version de cette méthode a constitué le module de recalage du démonstrateur AMR-ADAM, et nous l'avons développée en collaboration avec l'équipe ADAM à Matra Marconi Space <sup>1</sup>.

<sup>1</sup>Nous saluons ici Thierry Blais, Marie-Christine Tatard, Eric Boudin, Christophe Graulle, Arnaud Robert de Saint Vincent, Renaud Laurette et toute l'"équipe ADAM".

### 5.1.1 Extraction de primitives hétérogènes: le modèle instantané de localisation

Le modèle instantané de localisation est construit à partir d'une image de profondeur brute. La première phase du processus de détection de lignes verticales est menée dans le repère sphérique du capteur. Le principe de base est de parcourir l'image ligne par ligne et de détecter les discontinuités de profondeur. En choisissant une orientation pour le parcours de l'image (de gauche à droite par exemple), on peut distinguer des discontinuités positives (si la profondeur augmente) ou négatives. On peut ensuite chaîner dans un contour les points présentant des discontinuités de même type et correspondant à la même arête physique. Dans une deuxième phase, on utilise les paramètres de calibrage intrinsèque du capteur laser, ainsi que la connaissance de la verticale du robot (estimations du roulis et du tangage fournies par la centrale inertielle ou des inclinomètres) pour exprimer ces points dans le repère cartésien redressé du robot. On détermine les droites 3D qui approchent au mieux ces contours, selon un critère des moindres carrés; seuls les bords correspondant à des lignes effectivement verticales, et dont l'erreur d'approximation par rapport aux données est inférieure à un certain seuil, sont conservés ici, et on calcule leur position 2D  $(X, Y)$ .

L'incertitude sur ces primitives est obtenue par propagation à travers les différentes transformations (transformation sphérique-cartésien, approximation d'un ensemble de points par une droite, etc) des incertitudes des points de mesure bruts (selon un mécanisme exposé au paragraphe 4.2). Celles-ci nous sont fournies par le calibrage intrinsèque du télémètre laser, et l'étude statistique associée (cf. chapitre 2).

Notons ici que dans l'implémentation actuelle, nous nous intéressons à la détection de discontinuités de profondeur seulement. Selon l'angle de vue du capteur, l'angle d'un bâtiment, par exemple, qui correspond physiquement à une discontinuité de normale, peut être détectée ou pas (voir figure 5.1). Si le robot se trouve à l'intérieur des zones (a) ou (c), l'angle correspond bien à une discontinuité de profondeur dans l'image laser; par contre, il ne sera pas détecté si le robot se trouve dans la zone (b).

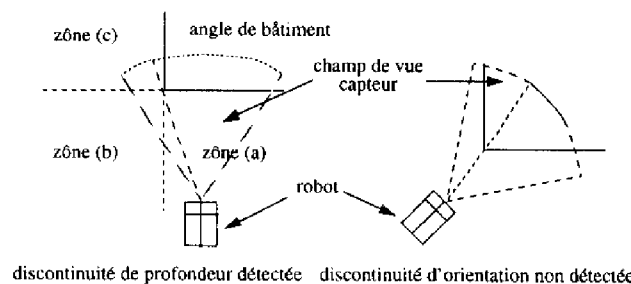


Figure 5.1: *Discontinuités de profondeur ou de normale*

Nous tenons compte de ce point dans la gestion du modèle de lignes verticales: nous verrons qu'une ligne peut être étiquetée comme discontinuité de profondeur ou d'orientation.

Nous pourrions également détecter aisément les discontinuités d'orientation, après une segmentation plane de l'image de profondeur [Devy 91]. Les lignes de discontinuité sont alors extraites avec une très bonne précision, puisqu'elles correspondent à l'intersection de deux faces planes adjacentes, dont les plans porteurs sont connus avec une très bonne précision si de nombreux points de mesure se trouvaient sur ces faces. Cependant, une telle segmentation est pour l'instant très coûteuse, et nous désirons ici éviter les temps de calcul importants.

Si nous recherchons dans l'environnement des poteaux, colonnes ou troncs d'arbre, nous devons rechercher des paires de lignes verticales, pour trouver les bords du même amer physique. Ces mises en correspondance sont réalisées selon des critères (1) de proximité "angulaire" des positions azimutales dans le repère sphérique de l'image de profondeur (2) de distance cartésienne et (3) de recouvrement en site dans l'image sphérique, entre des bords de type de discontinuité opposés. La position d'un amer de type colonne est définie comme la moyenne des positions de chaque bord, pondérées par les incertitudes correspondantes. Si l'amer vertical correspondant a une section circulaire, cette définition est bien adaptée. Sinon, l'estimation de la position de l'amer va varier avec le point de vue, et l'incertitude associée à cette position est plus grande.

Nous obtenons finalement un modèle instantané de localisation "hétérogène", composé de points 2D qui peuvent être étiquetés comme *colonne verticale* si 2 bords présentant des discontinuités de type opposé ont été mis en correspondance, ou comme *bord* sinon, qui peuvent correspondre à des discontinuités de profondeur ou d'orientation.

### 5.1.2 Estimation du mouvement et mise à jour du modèle de localisation

Le problème ici est de mettre en correspondance deux ensembles de points 2D (ceux contenus dans le modèle de localisation instantané et ceux contenus dans le modèle local), correspondant à des amers verticaux de l'environnement. Nous devons tenir compte dans le processus de mise en correspondance du type de primitives que nous manipulons. Si la mise en correspondance de points 2D possède l'avantage de conduire à des temps de calcul réduits, l'information sémantique contenue dans ces primitives est faible, et on risque de se trouver en face d'un grand nombre d'ambiguïtés, notamment si ces primitives sont nombreuses. Lorsque ce sera possible, nous leur associerons donc une information sémantique supplémentaire, qui pourra être utilisée lors de la procédure de mise en correspondance et pourra permettre de limiter le nombre d'appariements ambigus.

Les informations de trajectométrie fournies par l'odométrie ou la centrale inertielle permettent d'exprimer les points contenus dans le modèle instantané dans le repère local d'environnement. Les points locaux pouvant être appariés avec chacun de ces points sont dans un voisinage dont la taille dépend des incertitudes des points et de l'estimée de position trajectométrique. Plusieurs points locaux peuvent en outre se trouver dans un tel voisinage d'un point du modèle instantané (on parle alors de correspondance *ambiguë*). Les mises en correspondances ambiguës sont résolues par une procédure itérative d'appariement, dans

laquelle nous tenons compte des propriétés de rigidité des ensembles de points 2D.

L'algorithme de mise en correspondance des primitives, de recalage de la position du robot et de construction (fusion) incrémentale du modèle de localisation se déroule selon les étapes suivantes:

1. On recherche tous les appariements possibles, et on sélectionne les appariements qui sont non ambigus. S'ils existent, ces appariements "fiabes" sont conservés. Si aucun appariement non-ambigu n'a été détecté, on sélectionne les appariements les moins ambigus. On teste toutes les correspondances possibles, et on associe à chacune d'elles un **score** correspondant au nombre d'appariements cohérents avec elle, détectés au cours d'une procédure de recherche arborescente par propagation du recalage correspondant. Cette étape revient à utiliser des contraintes de rigidité entre les primitives, qui modélisent des amers statiques dans l'environnement. La position relative de plusieurs amers est ainsi notamment implicitement utilisée pour éliminer les correspondances ambiguës. Dans l'exemple donné figure 5.2, l'hypothèse d'appariement (A,1) est compatible avec (B,2), mais pas avec (B,3).

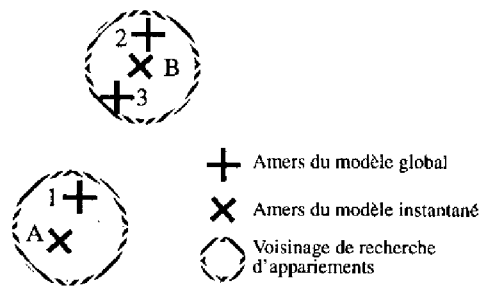


Figure 5.2: Appariements: levée d'une ambiguïté

À l'issue de cette étape 1, basée sur des tests heuristiques (distance entre les points, comparaison des étiquettes sémantiques *bord* ou *colonne*, contraintes de rigidité), on obtient un premier ensemble d'appariements "les plus certains".

2. On utilise l'ensemble d'appariements sélectionnés à l'étape 1 pour recalibrer la position du robot. En utilisant des appariements "sûrs", on diminue les biais sur la position du robot (et on diminue l'effet des futures linéarisations lors de l'introduction de nouvelles mesures, correspondant à de nouveaux appariements générés à l'étape 3); on rend aussi les tests probabilistes utilisés dans la suite plus sélectifs.

Les primitives mises en correspondance sont également réestimées ici.

3. On mène maintenant des tests probabilistes (basés sur une distance de Mahalanobis, cf. paragraphe 4.2) pour générer de nouveaux appariements *possibles*, et opérer un nouveau recalage-fusion avec ces nouveaux appariements. Ici, on fait un rebouclage

pour générer après chaque recalage-fusion de nouveaux appariements. Cette procédure itérative est arrêtée quand on ne peut plus générer de nouveaux appariements non ambigus. En général, 2 ou 3 itérations suffisent pour trouver toutes les “bonnes” correspondances. Les amers instantanés qui n’ont pas été mis en correspondance avec un amer du modèle local correspondent à des caractéristiques de recalage nouvellement perçues.

Au total, l’algorithme de recalage-fusion basé sur des points 2D peut être résumé par le schéma suivant:

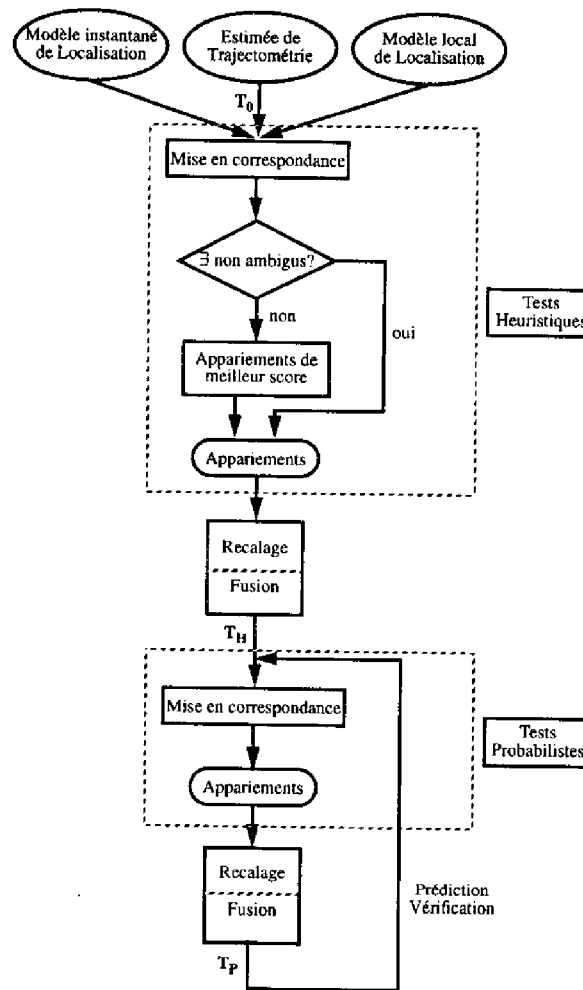


Figure 5.3: Recalage du robot et construction du modèle de localisation

Nous avons présenté au paragraphe 4.2 le formalisme adopté pour la fusion numérique et la manipulation des incertitudes. Ici, l’équation de mesure utilisée pour le filtre de Kalman est la suivante:



$$\begin{cases} X_l = X_s \cos \theta - Y_s \sin \theta + T_x \\ Y_l = X_s \sin \theta + Y_s \cos \theta + T_y \end{cases} \quad (5.1)$$

dans le cas de la mise en correspondance de deux points 2D (un point du modèle local  $(X_l, Y_l)$  et un point du modèle instantané  $(X_s, Y_s)$ );  $(T_x, T_y)$  désigne la translation et  $\theta$  l'angle de rotation (autour de la verticale  $Z$ ) entre les deux repères (ou modèles).

L'étiquette sémantique *bord* ou *colonne*, qui peut être utilisée lors du processus de mise en correspondance comme nous l'avons dit plus haut, doit par ailleurs être remise à jour lors du déplacement du robot, car cette étiquette peut changer:

- Certains amers peuvent d'abord être étiquetés *bord* puis *colonne*, dans le cas où seul un des bords aurait été détecté au départ (à cause de bruits de mesure, ou parce que le robot était loin de l'amer considéré).
- Réciproquement, un amer étiqueté *colonne* le restera (les critères d'extraction et de mise en correspondance de deux lignes verticales appartenant à un même amer de ce type étant sévères).

Dans le cas d'un amer de type *bord*, une information supplémentaire peut être utilisée: le *signe* de la discontinuité de profondeur. Mais, si une simple discontinuité de profondeur est présente dans la scène et peut être détectée, son signe change au fur et à mesure que le robot se déplace autour de ce bord (voir les exemples présentés figures 5.4 et 5.5).

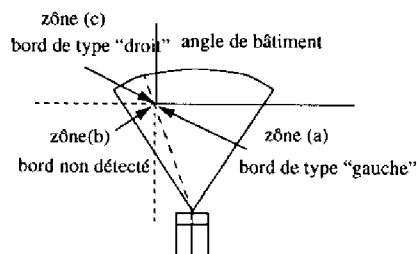


Figure 5.4: Relation entre discontinuité de profondeur et d'orientation

Le signe de la discontinuité de profondeur associée à un tel amer peut donc être exploité, à condition de pouvoir raisonner sur le point de vue. Ainsi par exemple, dans le cas de la figure 5.4, la connaissance de la position du robot et de son champ de vue permet de prédire quel signe doit être associé à la discontinuité qui doit (devrait) être perçue, et cette information supplémentaire peut encore être utilisée lors du processus de mise en correspondance d'amers. Les zones (a) et (b) dans un cas, (a) (b) et (c) dans l'autre cas, ne peuvent cependant être complètement déterminées qu'à partir de la connaissance des positions des murs de part et d'autre de la porte dans le premier cas, et de ceux qui forment

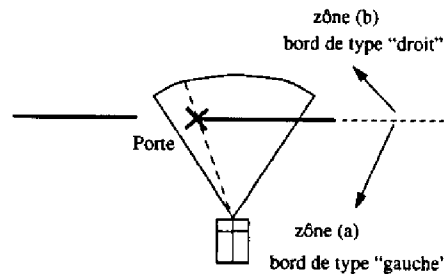


Figure 5.5: *Signe d'une discontinuité de profondeur: cas d'une porte*

l'angle de mur considéré dans l'autre. Ces informations pourront être tirées du modèle "murs" présenté au paragraphe 5.2. Le signe d'une discontinuité de profondeur simple pourra ainsi être utilisé, associé avec un point de vue, à l'aide d'une coopération sémantique entre les deux modèles de localisation présentés aux paragraphes 5.1 et 5.2. Une telle coopération permet de gérer la relation temporelle entre discontinuités de profondeur et discontinuités d'orientation; selon la position du robot par rapport à un amer, on peut, par prédiction des transitions possibles d'une zone (a) (b) ou (c) à l'autre, analyser la cohérence des primitives perçues, des appariements effectués, et du modèle local de localisation.

### 5.1.3 Résultats expérimentaux

Nous présentons dans ce paragraphe les résultats obtenus avec la méthode de localisation basée sur la détection de discontinuités de profondeur verticales. Cette expérience a été menée à l'aide de notre robot HILARE II, dans un environnement dont une image vidéo est présentée figure 5.6. La figure 5.7 donne un plan de l'environnement expérimental, avec le type d'amers verticaux et leurs positions.

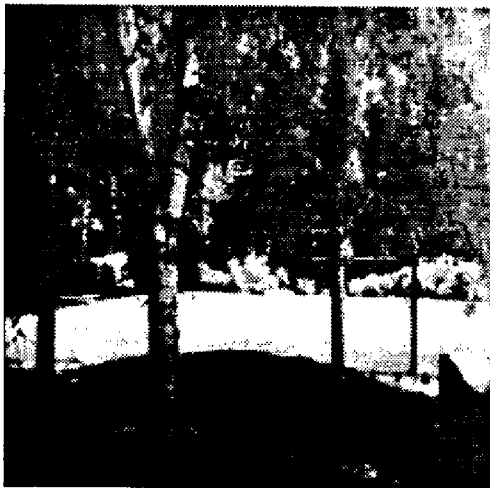


Figure 5.6: *Environnement expérimental: image vidéo*

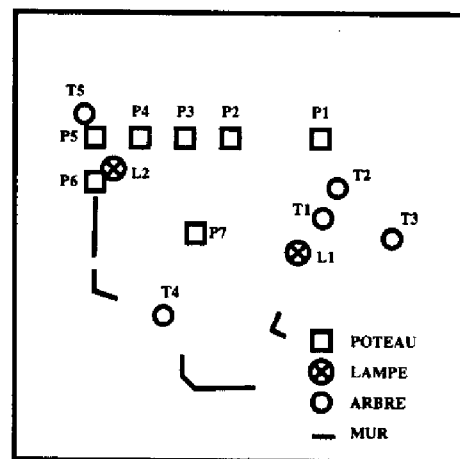


Figure 5.7: *Environnement expérimental: plan (vue de dessus)*

Le robot se déplace dans un environnement dont la taille est de 24m x 24m. Nous avons utilisé une longue séquence de 23 images de profondeur acquises durant le mouvement du robot. Ces acquisitions ont été faites à l'extérieur des bâtiments au LAAS. Ces conditions expérimentales ont conduit à l'obtention de données bruitées, mais cependant très largement exploitables. Nous avons choisi pour le balayage angulaire une résolution plutôt faible (un degré en site et en azimut), qui s'est montrée cependant suffisante pour avoir une bonne localisation du robot. Ainsi, nous avons évité de trop longs temps d'acquisition, la procédure de recalage elle-même étant très peu coûteuse.

Sur les figures suivantes, nous représentons par des (X) les amers des modèles instantanés successifs, et par des (+) les amers contenus dans le modèle local. Les figures 5.8 et 5.9 montrent un exemple d'image de profondeur brute, et l'extraction de lignes verticales correspondantes. Les deux types de discontinuités de profondeur sont codés de manière distincte sur la figure. Quatre paires de lignes verticales de discontinuités de profondeur de types opposés ont été détectées et appariées, et le modèle instantané de localisation contient donc quatre amers verticaux (cf. figure 5.12).

Les figures 5.10 et 5.11 montrent les résultats de la première acquisition. Deux amers ont été détectés dans le modèle instantané, et le modèle local est construit en plaçant ces deux amers dans le modèle local, en utilisant l'information de trajectométrie.

Les figures 5.12 et 5.13 montrent les résultats de la deuxième acquisition sensorielle. Quatre amers verticaux ont été détectés; deux d'entre eux ont été mis en correspondance avec les deux amers constituant le modèle local. Les positions de ces deux amers sont remises à jour et la position du robot recalée (le recalage est rendu apparent sur la figure 5.12). Les deux autres sont rajoutés dans le modèle local en tenant compte de la position recalée du robot (et son incertitude) et l'incertitude sur la position des amers dans le repère instantané.

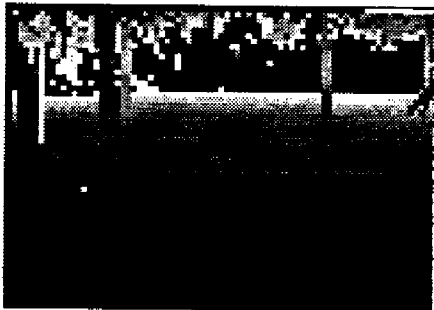
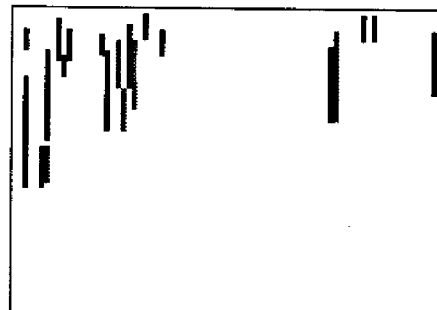
Les figures 5.14 et 5.15 présentent un cas où des glissements se sont produits: l'incertitude sur l'estimée de position fournie par trajectométrie est importante, et le recalage permis par les 3 appariements d'amers est grand. A la position 9, il n'y a pas eu de mise en correspondance et on est dans une situation analogue à celle de la position 1. A la position 22 (figures 5.20 et 5.21), le recalage est important car le robot n'a pas détecté d'amers depuis longtemps et l'incertitude sur l'estimée de position est devenue importante. En position 23, le recalage est de nouveau plus faible. La figure 5.24 présente enfin les 23 positions du robot, avec la superposition dans le repère local de tous les modèles instantanés correspondants.

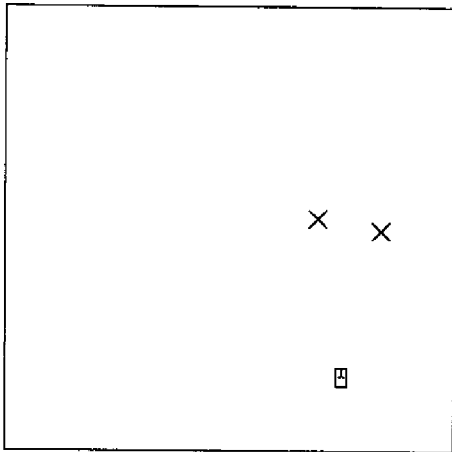
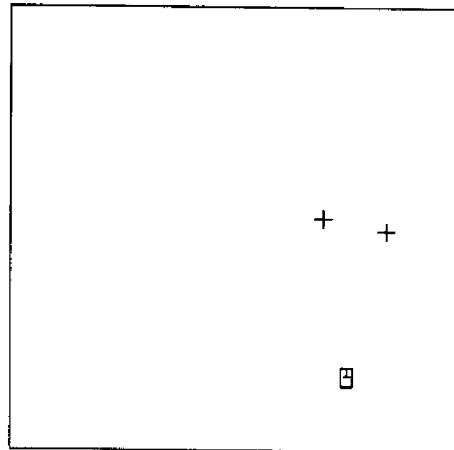
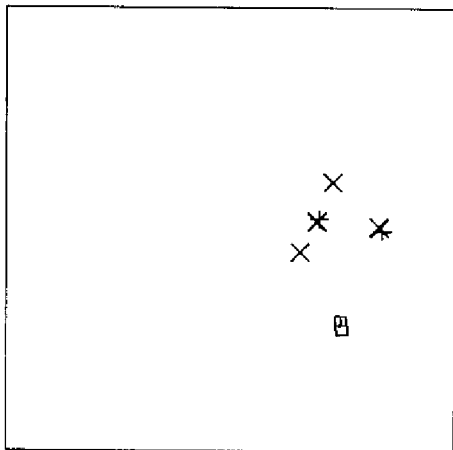
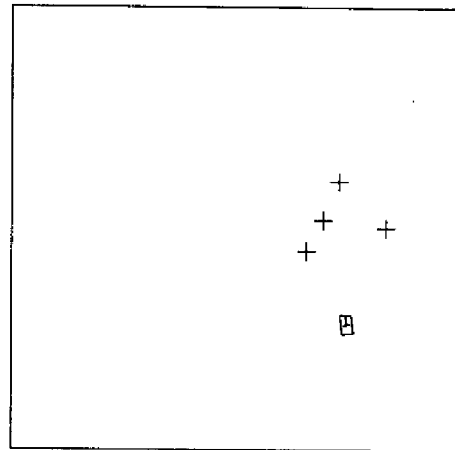
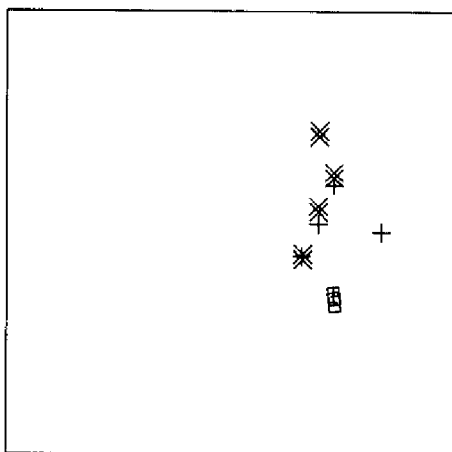
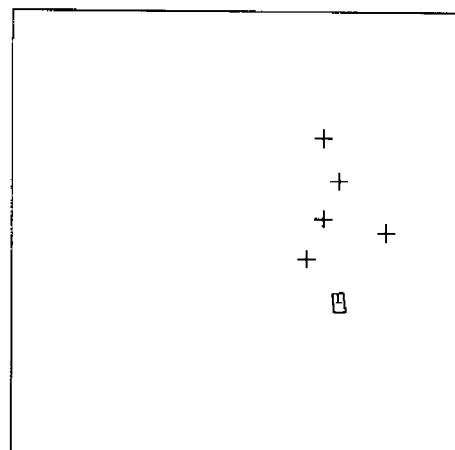
Pour cette mission, HILARE II ne disposait d'aucune connaissance a priori sur son environnement (on aurait pu par ailleurs envisager de pré-charger un modèle de localisation).

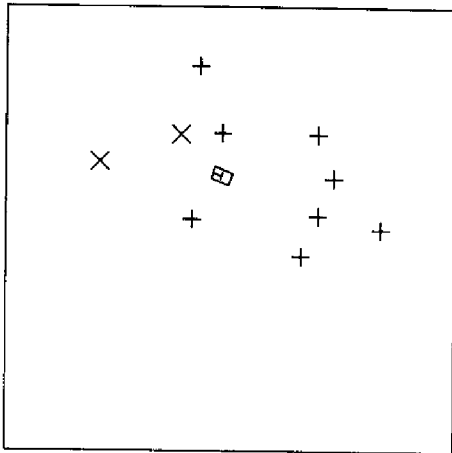
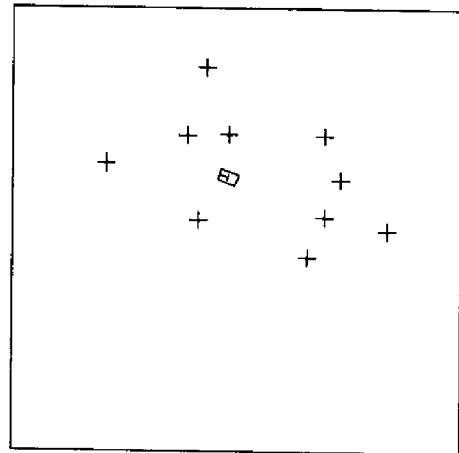
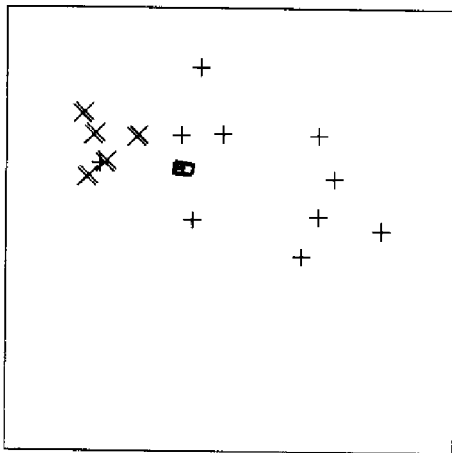
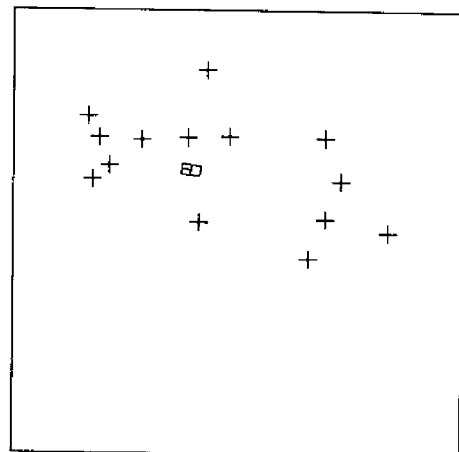
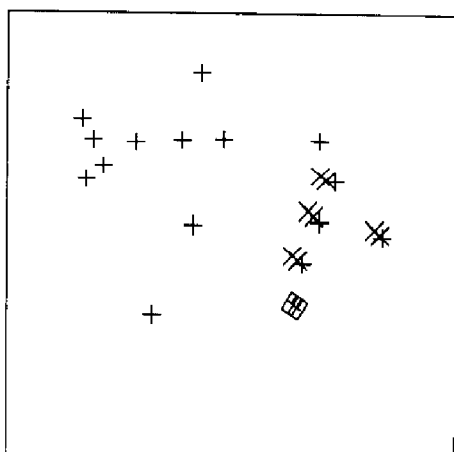
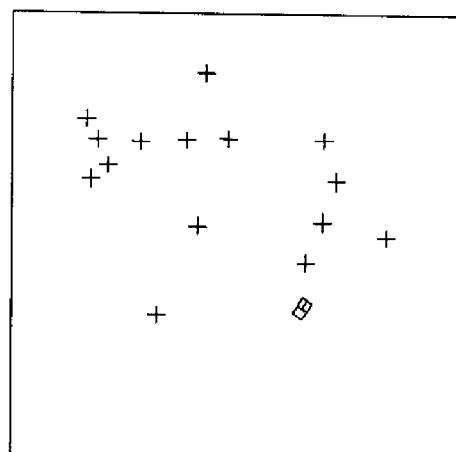
Le tableau suivant présente les résultats numériques que nous avons obtenus pour le modèle local final. Pour chaque amer détecté, nous donnons le nombre de fois où il a été perçu par le robot, sa position 2D finale, et les variances de ses coordonnées  $X$  et  $Y$ .

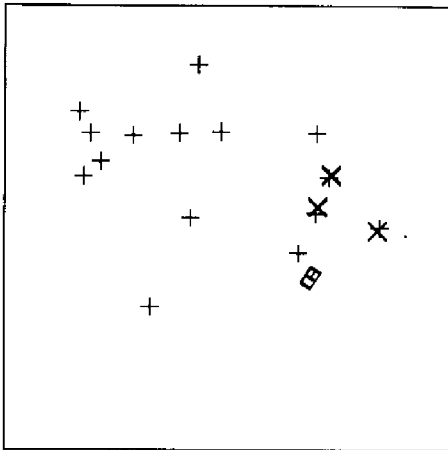
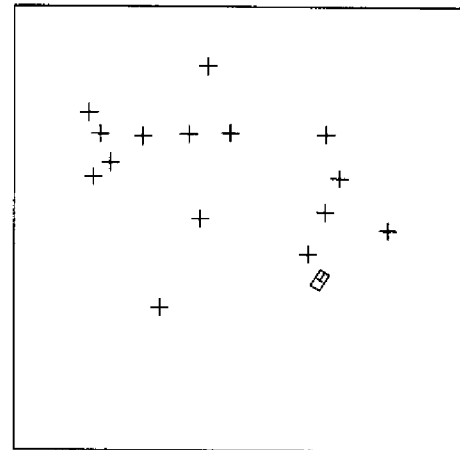
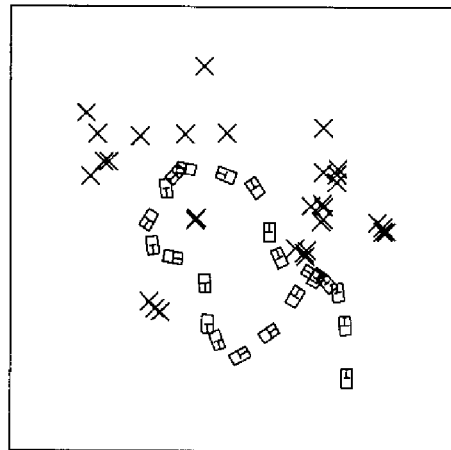
Amer	X (m)	Y (m)	Var X	Var Y	Nb de perceptions
$T_1$	16.49	12.94	0.073	0.007	5
$T_2$	17.23	14.74	0.092	0.007	4
$T_3$	20.16	11.89	0.097	0.007	4
$T_4$	8.03	7.50	0.105	0.009	3
$T_5$	3.92	18.28	0.134	0.125	1
$L_1$	15.22	10.92	0.081	0.007	3
$L_2$	9.93	12.20	0.101	0.105	2
$P_1$	16.70	16.95	0.112	0.025	1
$P_2$	11.93	17.09	0.409	0.250	1
$P_3$	9.26	17.23	0.274	0.250	1
$P_4$	6.71	17.18	0.134	0.125	1
$P_5$	4.52	16.95	0.139	0.126	1
$P_6$	4.07	14.53	0.132	0.120	1
$P_7$	5.12	15.37	0.103	0.021	2

Nous obtenons finalement un modèle local cohérent. Comme nous nous y attendions, les incertitudes sur la position des amers diminuent lorsqu'ils sont de nouveau perçus. Une lecture plus attentive des résultats nous permet d'expliquer les différences entre les incertitudes sur les amers qui n'ont été perçus qu'une fois (par exemple).  $P_2$  et  $P_3$  sont ceux qui ont l'incertitude la plus forte; ils ont été perçus à des positions où aucun recalage du robot n'a pu être effectué (cf. figures 5.16 et 5.17 pour  $P_3$ ).  $T_5$ ,  $P_4$ ,  $P_5$  et  $P_6$  ont été perçus à la position 10 (cf. figures 5.18 et 5.19) où 1 amer a permis un recalage du robot; leur incertitude est plus faible. Enfin,  $P_1$  a été perçu à la position 3, à laquelle 3 amers ont permis un recalage du robot plus robuste;  $P_1$  est, parmi les amers perçus une seule fois, celui qui a l'incertitude la plus faible.

Figure 5.8: *Image laser brute*Figure 5.9: *Lignes verticales détectées*

Figure 5.10: *Modèle instantané 1*Figure 5.11: *Modèle local 1*Figure 5.12: *Modèle instantané 2*Figure 5.13: *Modèle local 2*Figure 5.14: *Modèle instantané 3*Figure 5.15: *Modèle local 3*

Figure 5.16: *Modèle instantané 9*Figure 5.17: *Modèle local 9*Figure 5.18: *Modèle instantané 10*Figure 5.19: *Modèle local 10*Figure 5.20: *Modèle instantané 22*Figure 5.21: *Modèle local 22*

Figure 5.22: *Modèle instantané 23*Figure 5.23: *Modèle local 23*Figure 5.24: *23 Modèles instantanés*

#### 5.1.4 Extension possible: estimation de la verticale

La détection d'amers verticaux peut permettre d'estimer également la verticale du robot (valeurs des angles de roulis et de tangage), si un tel amer, détecté dans l'environnement du robot, peut être mis en correspondance avec un amer dans le modèle local dont nous savons qu'il est vertical dans ce repère.

En utilisant les estimées d'attitude fournies par trajectométrie inertielle, on peut projeter dans le repère local les *droites 3D* instantanées approxinant les bords détectés. On cherche alors les correspondances dans le repère local  $(X, Y)$  comme expliqué ci-dessus. Si au moins un amer instantané peut être mis en correspondance avec un amer vertical du repère local, on utilise les équations des droites 3D dans le repère instantané, approchant les bords détectés, pour estimer le roulis et le tangage. En effet, si  $\phi$  et  $\psi$  désignent respectivement les angles de tangage et de roulis, une droite 3D verticale dans le repère local a pour vecteur directeur

$(-\sin \phi, \sin \psi \cdot \cos \phi, \cos \psi \cdot \cos \phi)$  dans le repère instantané robot non redressé.

## 5.2 Recalage 2D sur obstacles verticaux

Dans cette méthode, la localisation repose sur la détection dans l'environnement du robot, d'obstacles verticaux, du type murs, ou objets polyédriques [Boissier 85, Moutarlier 89a, Moutarlier 91]. De tels obstacles peuvent être détectés en utilisant une coupe laser horizontale de l'environnement (ceci peut correspondre à une ligne d'une image laser, associée à une valeur choisie pour l'angle de site, si le mode de balayage du capteur est *corrélé* –cf. Annexe A–).

Les représentations internes de tels obstacles sont des segments 2D, correspondant à la projection des obstacles verticaux dans le plan Oxy. La mise en correspondance de ces segments d'une vue à l'autre permet de recalibrer la position du robot et de construire incrémentalement un modèle de localisation contenant l'ensemble des segments décrivant les obstacles verticaux rencontrés.

Nashashibi et Devy ont généralisé l'approche "murs" en utilisant non pas les projections au sol de murs verticaux, mais des plans 3D quelconques [Nashashibi 93b], pour recalibrer les 6 coordonnées de position et d'attitude du robot HILARE. Dans le cadre d'AMR, on souhaite surtout disposer d'une méthode de recalage rapide; c'est pourquoi seules des coupes laser ont été utilisées. L'utilisation d'un modèle polyédrique permet un recalage performant du robot et une analyse sémantique très intéressante pour l'interprétation de scènes structurées, ou le maintien de la cohérence des modèles du monde dans de tels environnements. En contrepartie, elle nécessite une segmentation plane des données sensorielles brutes coûteuse en temps de calcul.

### 5.2.1 Modèle instantané de localisation

Nous supposons que le rayon laser, au cours de sa rotation, décrit un plan appelé le *plan laser*, qui contient l'ensemble des points de mesure bruts. Les données obtenues lors d'une telle acquisition sont constituées par un ensemble de points  $(\rho, \theta)$  ordonnés, exprimés dans un repère polaire du plan laser. La précision des mesures a été déterminée expérimentalement (lors d'une procédure de calibrage). Les erreurs sur les coordonnées polaires  $\rho$  et  $\theta$  sont supposées indépendantes.

On réalise tout d'abord une approximation polygonale de l'ensemble des points de mesure, par une procédure de segmentation de type *split and merge*, en appliquant un critère du type moindres carrés. Chaque segment est décrit par une équation de la droite porteuse, et ses extrémités. Un algorithme basé sur le Filtre de Kalman Généralisé fournit les paramètres pour chaque droite, ainsi que leurs variances. Une première estimée de la droite est déterminée comme la droite passant par les deux extrémités du groupe de points alignés. Les extrémités des segments sont déterminées soit comme intersections de droites approximantes



voisines, soit comme intersections des droites approchant les données avec les directions du laser. Leurs incertitudes sont directement déduites de celles des droites mises en jeu.

Enfin, une procédure d'analyse des données permet de caractériser les extrémités des segments [Boissier 85, Khoumsi 88]. La figure 5.25 illustre les différentes étiquettes que l'on peut associer aux extrémités: une extrémité de type **S** correspond à un sommet réel, une extrémité de type **P** correspond à la projection d'un sommet de type **S** (limite d'occultation).

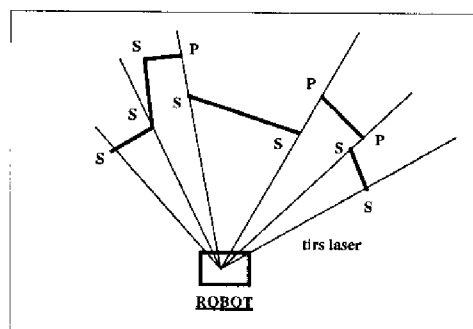


Figure 5.25: *Modèle instantané de localisation sur obstacles verticaux*

### 5.2.2 Estimation du mouvement et mise à jour du modèle local de localisation

Le problème ici est de mettre en correspondance deux ensembles de segments 2D. Deux appariements entre segments du modèle instantané et du modèle local suffisent pour recalculer la position  $(X, Y, \theta)$  du robot. Seules les équations des droites porteuses des segments 2D sont utilisées lors de la procédure de recalage-fusion, qui suit le paradigme général présenté au paragraphe 1.3. Rappelons-en les principales étapes [Moutarlier 89b, Moutarlier 91]:

- on mène tout d'abord une recherche de correspondances entre les segments du modèle instantané et du modèle local, par une méthode du type transformée de Hough; la recherche est basée sur des tests heuristiques sur les longueurs des segments, leurs orientations et les distances entre les centres des segments;
- on calcule une première estimée du recalage de la position du robot, à l'aide du Filtre de Kalman généralisé;
- on mène une nouvelle recherche de correspondances tenant compte de la nouvelle estimée de position du robot, sur la base de tests probabilistes, basés sur la distance de Mahalanobis;
- enfin, on réestime la position du robot et on remet à jour le modèle local de localisation.

Seules les droites porteuses des segments sont utilisées lors de la procédure de filtrage numérique de recalage-fusion. L'équation de mesure utilisée ici pour estimer le mouvement à partir d'un ensemble de mises en correspondance est:

$$\begin{cases} d_l = d_s + X \cos(\phi_s + \theta) - Y \sin(\phi_s + \theta) \\ \phi_l = \phi_s + \theta \end{cases} \quad (5.2)$$

dans le cas où deux segments ont été mis en correspondance.  $(d_l, \phi_l)$  et  $(d_s, \phi_s)$  sont les paramètres polaires définissant les droites support de deux segments appartenant respectivement au modèle local et au modèle instantané de localisation.  $\phi$  désigne la direction de la droite, et  $d$  sa distance à l'origine du repère polaire (instantané -i.e. robot-, ou local).

### 5.2.3 Résultats expérimentaux

De (très !) nombreux résultats peuvent être trouvés dans [Moutarlier 91]. Nous présentons simplement ici un exemple de construction incrémentale du modèle d'obstacles verticaux après 4 itérations de la chaîne perception/modélisation/mouvement, dans l'environnement de notre salle de robotique.

Sur la figure 5.26, nous montrons la superposition, dans le repère local d'environnement, des modèles instantanés successifs. La figure 5.27 présente le modèle local finalement obtenu, et la figure 5.28 montre le recalage effectué à la position 3.

## 5.3 Coopération entre les deux méthodes

Chacune des deux méthodes de localisation présentées dans ce chapitre repose sur la détection d'amers d'un type particulier: lignes verticales ou obstacles verticaux de type "mur". Si, lorsque la mission du robot est définie, on décide d'utiliser l'une ou l'autre de ces méthodes de localisation, il est possible que le recalage ne puisse être fait pour cause d'absence dans l'environnement d'amers du type désiré, ou parce qu'ils ne sont pas présents en assez grand nombre pour recalculer toutes les coordonnées de position que l'on souhaite corriger. La fin de l'expérimentation menée pour le recalage sur des lignes verticales a montré un recalage très important à la position 22, car le robot n'avait pas détecté de tels amers depuis longtemps. Si l'absence d'amers recherchés s'était prolongée encore, l'incertitude sur l'estimée trajectométrique de position serait vite devenue trop importante pour pouvoir construire un modèle local de l'environnement cohérent.

Par ailleurs, dans le cadre de la robotique mobile d'intervention dans un bâtiment endommagé (mission de type AMR), où le robot doit se déplacer en direction d'un bâtiment puis entrer dans celui-ci, les amers présents dans l'environnement seront plutôt des murs, angles de murs, cadres de portes, colonnes, ... à l'intérieur du bâtiment, des murs et des arbres aux abords du bâtiment, et peut-être simplement des arbres à grande distance de celui-ci.

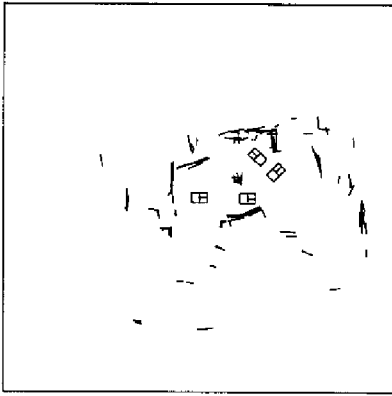


Figure 5.26: *Modèles instantanés successifs*

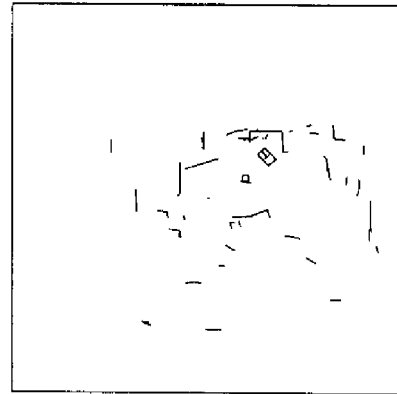


Figure 5.27: *Modèle local final*



Figure 5.28: *Recalage position 3*

Aussi serait-il intéressant de pouvoir utiliser en même temps les deux types d'amers présentés, pour augmenter les chances de pouvoir recalibrer la position du robot dans toute situation durant la mission. Comme nous l'avons souligné au chapitre 4, plusieurs manières d'intégrer les deux méthodes de localisation apparaissent:

1. Utiliser chacun des capteurs (logiques, cf. paragraphe 4.2) séparément et en parallèle tout d'abord; chacun des capteurs calcule une estimée partielle de l'état, puis calcule une estimée globale à l'aide des estimées partielles et des mesures des autres capteurs. Cette approche a été présentée figure 4.2 au chapitre 4.
2. Intégrer les deux types d'amers dans un seul modèle de localisation hétérogène, comprenant des points 2D et des segments 3D. Un seul segment peut être suffisant pour recalibrer le cap du robot, et un seul point supplémentaire peut suffire à recalibrer la position  $(X, Y)$ . Cette approche a été présentée par la figure 4.3 au chapitre 4.

Signalons également qu'il est possible de générer des primitives d'un type donné à partir de primitives contenues dans un autre modèle. L'exemple le plus intéressant est celui de

la génération de points “coins de murs” à partir de deux segments formant un coin dans le modèle “murs”. De tels coins sont connus avec une très bonne précision puisque les droites porteuses des segments sont connues avec une grande précision. La propagation des incertitudes se fait en utilisant le formalisme présenté au paragraphe 4.2.

Dans la suite, nous présentons le choix que nous avons effectué, avant de nous intéresser à la stratégie de sélection des amers pour le recalage.

### 5.3.1 Intégration des deux méthodes de localisation

L'inconvénient de l'approche 1 ci-dessus est que chacun des modèles de localisation peut conduire à un nombre de primitives insuffisantes pour faire le recalage de tous les paramètres de position du robot (recalage de  $X, Y, \theta$  dans notre cas).

Nous lui préférons l'approche 2 qui permet d'utiliser en même temps des amers hétérogènes pour effectuer le recalage. En effet, un segment et un point suffisent à corriger la position  $(X, Y)$  et le cap  $\theta$  du robot, alors que chacune des primitives prise séparément n'aurait pas suffi. Notre outil de filtrage numérique basé sur le Filtre de Kalman Généralisé [Moutarlier 91] permet d'utiliser des primitives hétérogènes pour corriger la position du robot et fusionner ces primitives dans le repère local. Il nous suffit d'utiliser une des deux équations de mesure 5.1 ou 5.2, selon le type de primitives mises en correspondance.

Comme expliqué au paragraphe 4.2 E/, nous utilisons séquentiellement les primitives qui conduisent aux appariements les plus *certaines*, pour *diminuer le biais* sur la position du robot et ceux dûs aux non-linéarités. Les *tests probabilistes* sur les appariements ultérieurs deviennent ainsi *plus sélectifs*.

L'utilisation de primitives hétérogènes pose donc le problème du choix des amers qui seront utilisés pour le recalage, surtout si ceux-ci sont nombreux dans l'environnement. Dans le paragraphe suivant, nous donnons quelques éléments qui pourront être utiles pour choisir, à une certaine étape de la mission du robot, les meilleurs amers pour optimiser les résultats du recalage.

### 5.3.2 Choix des amers et recalage du robot

#### A/ Nombre minimum de primitives appariées nécessaires

Pour effectuer un recalage correct, nous voulons utiliser au moins:

- 2 points dans le modèle lignes verticales, ou deux segments dans le modèle murs, ou un point et un segment, pour recalcr  $X, Y$  et  $\theta$ ;
- un point pour recalcr  $X$  et  $Y$ ;
- un segment pour recalcr  $\theta$ .

## B/ Quelques règles de base

Nous énumérons ici quelques règles de base qui doivent guider le choix des primitives utilisées, pour optimiser le recalage, compte tenu des remarques faites tout au long de ce chapitre.

1/ En ce qui concerne l'*imprécis*, nous connaissons la précision que l'on peut attendre de l'extraction des différents types de primitives.

Dans le modèle "segments", les droites porteuses, qui sont utilisées lors de la procédure de recalage, sont connues avec une très bonne précision si les segments sont de longueur suffisante; les problèmes de specularité et surtout de *footprints* (cf. paragraphe 2.1.2), qui ne se manifestent, éventuellement, que pour les extrémités, sont peu sensibles. Par contre, il peut se trouver des points de faible précision parmi les points laser correspondant à une discontinuité de profondeur (problème de "footprints" plus sensible). Ainsi, dans le modèle "points", un point de type "coin" généré à partir du modèle "segment" sera plus précis qu'un point correspondant à un amer de type *colonne*, puis qu'un point correspondant à un amer de type *bord*.

Au total, on peut s'attendre à un meilleur recalage si on utilise des amers de type "mur" que si on utilise des amers de type "ligne verticale".

2/ En ce qui concerne l'*incertain*, les amers les plus *sûrs* sur lesquels on peut compter sont les segments, s'ils ont une longueur suffisante. Par ailleurs, dans le modèle "points", l'existence d'un amer de type *colonne* ou *arbre* est plus *sûre* que celle d'un amer de type *bord*, puisque l'information sémantique associée est plus forte (deux discontinuités de profondeur de type différent ont été mises en correspondance lors du processus de détection d'amers).

Au cours d'une mission, nous devons choisir les amers utilisés en fonction du type d'amers présents et des critères précédents.

## C/ Missions et stratégies de recalage

Nos travaux dans ce domaine visent à élaborer une stratégie pour le choix des amers à utiliser pour recalculer avec la meilleure précision la position du robot. Nous devons distinguer deux contextes:

### Environnement connu

Si un modèle de localisation *parfait* (carte des amers de l'environnement) connu a priori a été pré-chargé, c'est à dire si la position des amers est connue a priori avec une très grande précision, alors la planification de trajectoire du robot doit tenir compte des positions de ces amers, de telle manière qu'ils puissent être perçus dans de bonnes conditions par le robot. Sur une trajectoire donnée, il faut déterminer à quelles positions et avec quels paramètres d'acquisition les perceptions doivent être effectuées, de manière à optimiser le nombre d'amers détectés d'une part, et le nombre d'amers mis en correspondance avec des amers du modèle local d'autre part. En fonction de la nature des amers "disponibles", le

recalage sera effectué avec la meilleure méthode, c'est à dire en utilisant des murs, des lignes verticales, ou les deux.

#### Environnement inconnu

L'intégration de différents types d'amers dans le même modèle produit des données plus nombreuses. Mais essayer de mettre en correspondance tous les amers possibles à chaque perception peut conduire à une explosion combinatoire. Comment choisir les meilleurs appariements de caractéristiques ?

Si nous sommes en présence d'un des cas minimaux (du point de vue du nombre d'amers détectés et mis en correspondance) présentés ci-dessus, nous utiliserons évidemment tous les amers disponibles pour recalculer les paramètres de position qui peuvent l'être ( $X$  et  $Y$  seulement, ou  $\theta$  seulement, ou les trois). Dans les autres cas, le choix des primitives et l'ordre dans lequel les mesures sont considérées est fait en fonction des critères énoncés ci-dessus. La procédure de sélection peut également faire intervenir la précision des primitives extraites, et des critères heuristiques comme la longueur des segments verticaux et des lignes horizontales détectées puis appariées, la disparité entre les segments appariés, ... Comme nous le verrons dans la conclusion de cette partie II, l'introduction de plus de sémantique dans les modèles de localisation devrait permettre d'améliorer la mesure de certitude des appariements de primitives réalisés.

Le schéma de la figure 5.29 résume l'algorithme de localisation-recalage / fusion basé sur des "points" et "segments" que nous avons développé. Pour diminuer l'*incertain*, on utilise d'abord les segments pour le recalage. Le biais sur la position du robot étant alors fortement réduit, les tests d'appariements entre points, dont nous avons vu qu'ils peuvent facilement conduire à des cas ambigus, sont plus sélectifs.

Enfin, deux possibilités se présentaient pour l'estimation des paramètres  $(X, Y, \theta)$  de position et d'orientation du robot. Une première approche consistait à estimer ces 3 paramètres simultanément, en utilisant, selon le type d'amer mis en correspondance, une des deux équations de mesure 5.1 ou 5.2. Nous avons préféré estimer séparément  $\theta$  et  $(X, Y)$  lorsque cela est possible. En effet, si des segments de longueur suffisante sont détectés et mis en correspondance, ils sont bien adaptés pour estimer l'orientation du robot, car les droites porteuses sont connues avec une bonne précision; ils sont ainsi mieux adaptés que les primitives de type point. L'équation de mesure 5.2 devient alors l'équation 5.3:

$$\phi_{\mathbf{l}} = \phi_{\mathbf{s}} + \theta \quad (5.3)$$

La détermination préalable de  $\theta$  permet ensuite de contraindre fortement (focaliser) la recherche d'appariements pour la détermination de  $(X, Y)$ , en réduisant beaucoup l'effet "bras de levier".

Nous savons que l'estimation séparée des différents paramètres de l'état ne conduit pas aux mêmes résultats qu'une estimation globale [Ayache 88]. Cependant, l'approche que nous

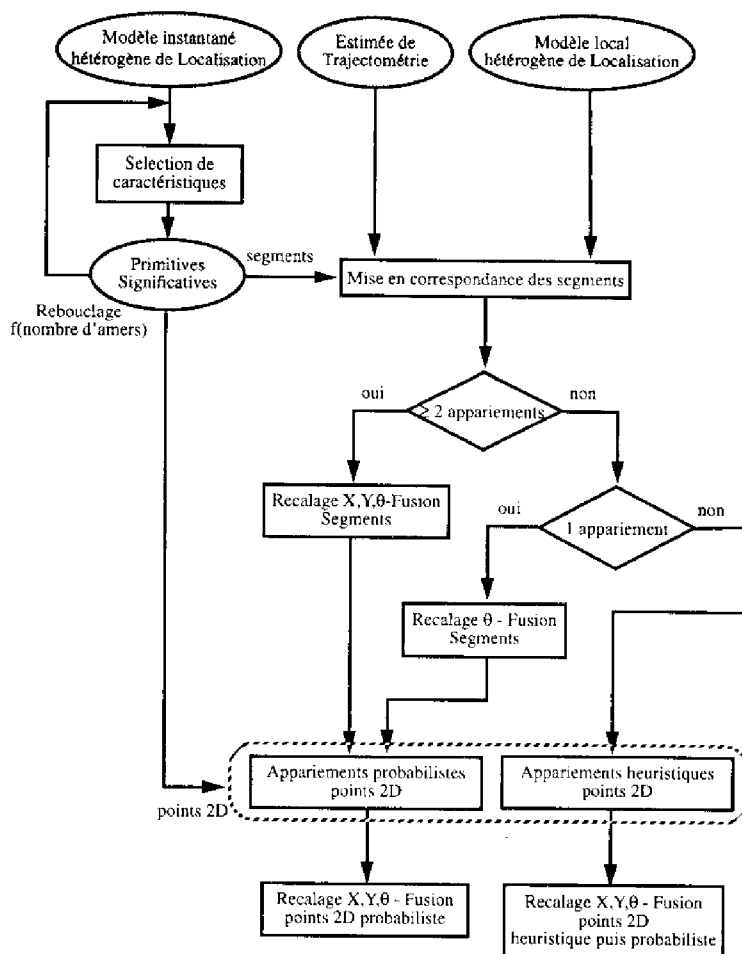
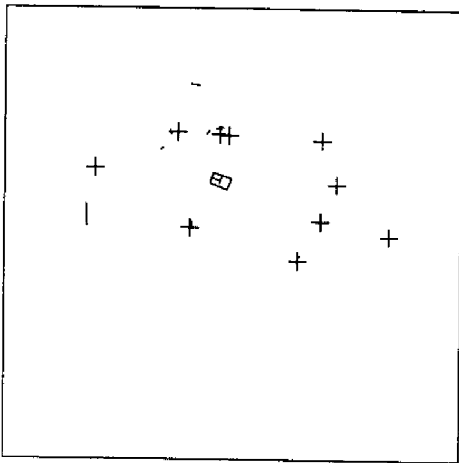
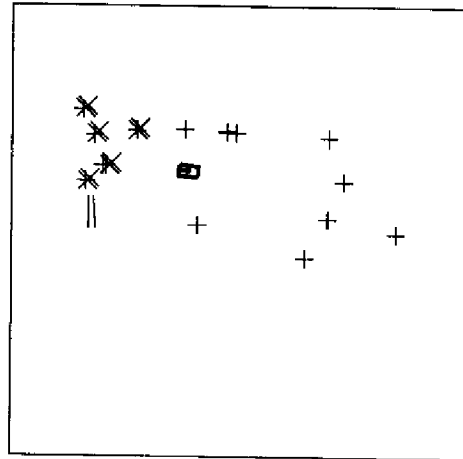
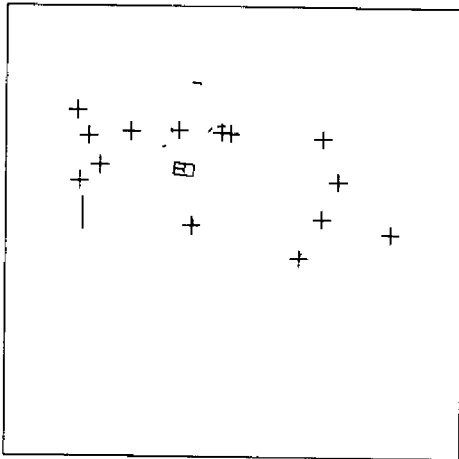
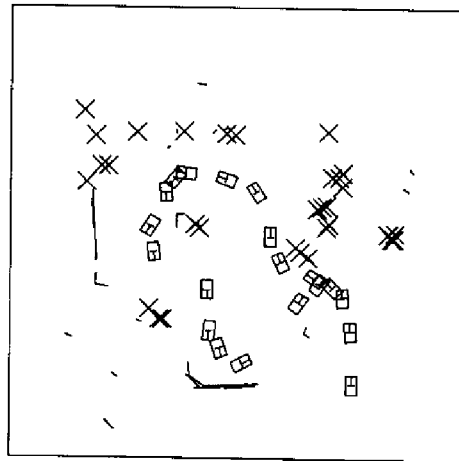
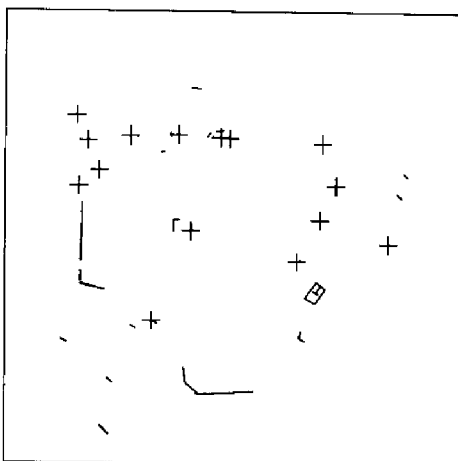


Figure 5.29: *Recalage-fusion à l'aide de primitives hétérogènes*

avons adoptée présente les avantages de permettre une focalisation de la recherche d'appariements (diminution de l'*incertain*), d'utiliser chaque modèle de localisation en fonction de ses qualités, et de rendre les équations de mesure utilisées linéaires, ce qui permet d'éviter les biais normalement induits par la linéarisation des équations non-linéaires lors de l'utilisation du FKG.

Nous donnons figures 5.30 à 5.34 quelques résultats expérimentaux obtenus pour le recalage du robot à partir de lignes verticales ou d'obstacles de type "mur", toujours dans l'environnement expérimental présenté au paragraphe 5.1.3, mais dans lequel des obstacles verticaux de type "mur" sont désormais utilisés. Les figures 5.30, 5.31 et 5.32 montrent le modèle hétérogène local à la position 9, le recalage à la position 10, et le modèle local après fusion. Le recalage de la position  $(X, Y, \theta)$  a été calculé en utilisant une correspondance entre points et une correspondance entre segments. Les figures 5.33 et 5.34 montrent enfin la superposition de tous les modèles hétérogènes de localisation et le modèle local hétérogène final de localisation.

Figure 5.30: *Modèle local hétérogène 9*Figure 5.31: *Recalage hétérogène position 10*Figure 5.32: *Modèle local hétérogène 10*Figure 5.33: *Modèles instantanés hétérogènes*Figure 5.34: *Modèle local 23*





## Conclusion de la partie II

Dans cette partie, nous avons abordé le problème du recalage de la position d'un robot mobile dans un environnement semi-structuré, pouvant être extérieur, et inconnu (ou partiellement connu) a priori. Deux systèmes de modélisation incrémentale, dédiée à la localisation, ont été présentés. Chacun de ces modèles de localisation contient une représentation de l'ensemble des amers (qui peuvent éventuellement être *naturels*) d'un type donné, rencontrés au cours d'une mission. Ces amers sont détectés dans une image de profondeur entière, ou une simple coupe horizontale, acquises à l'aide d'un télémètre laser. Le recalage repose sur la mise en correspondance d'une acquisition sensorielle à l'autre:

- d'une part, d'amers verticaux de type arbre, colonne, ou ligne de discontinuité verticale; la représentation interne d'un tel amer est un point  $(x, y)$  donnant sa position 2D.
- d'autre part, d'amers de type "mur"; la représentation d'un tel amer est un segment 3D horizontal.

Les modèles sont mis à jour à l'aide d'une procédure de recalage-fusion reposant sur l'utilisation du filtre de Kalman. Les paramètres de position du robot recalés sont la position  $(X, Y)$  et le cap  $\theta$ , le roulis et le tangage  $(\phi, \psi)$  pouvant être éventuellement réestimés en s'appuyant sur la reconnaissance d'amers verticaux connus a priori.

Cependant, si des amers d'un type déterminé ne sont pas présents dans l'environnement ou ne peuvent être détectés, chacune des deux méthodes de localisation (utilisée séparément), peut conduire à l'échec de la procédure de recalage, alors que l'utilisation de primitives des deux types aurait permis de réaliser le recalage avec succès. La construction d'un modèle hétérogène de localisation, à partir d'amers de nature différente, contenant donc des primitives de nature différente (ici des points ou des segments), est nécessaire pour que le robot puisse s'adapter à différents types d'environnements. Le robot AMR peut ainsi se diriger vers des bâtiments dans un environnement d'extérieur faiblement structuré, en utilisant des arbres pour recalculer sa position. Aux abords du bâtiment, il peut utiliser des arbres, des murs, des lignes de discontinuité de profondeur verticales. Dans un bâtiment, il peut utiliser des murs et des lignes verticales.

L'utilisation de primitives hétérogènes pose le problème de l'élaboration d'une stratégie de choix des amers. Celle-ci doit tenir compte de critères à la fois sémantiques (type d'amer

extrait, fiabilité de la procédure d'extraction, confiance dans l'existence d'un amer, ...) et numériques. Nous avons proposé un algorithme de recalage sur la base d'amers hétérogènes.

Construire séparément deux modèles *reposant exclusivement sur une description géométrique* conduit à des modèles trop complexes et trop fragiles. Il est notamment difficile de maintenir la cohérence du modèle. On accumule des primitives d'un type donné, qu'il n'est pas toujours très facile de filtrer. Certaines mises en correspondance peuvent révéler des ambiguïtés difficiles à lever. Enfin, les procédures de mise en correspondance deviennent plus complexes au fur et à mesure que les modèles croissent, et peuvent conduire à une explosion combinatoire.

D'autres niveaux de connaissance sont nécessaires. La modélisation par primitives hétérogènes introduite dans ce chapitre amorce les travaux de notre équipe dans ce domaine. Comme nous l'avons dit dans le chapitre 5, la gestion de la cohérence du modèle obtenu (notamment le filtrage des primitives non-significatives) peut se faire de manière plus robuste à travers une coopération sur les plans numérique et surtout sémantique des deux types de modèles (ou de différents types de modèles en général).

Dans l'état actuel du système, nous pouvons utiliser des amers hétérogènes pour assurer le recalage du robot. Nous avons validé les méthodes de localisation proposées par expérimentation sur de longues séquences d'images laser et de longs déplacements du robot HILARE II, dans un environnement extérieur semi-structuré. Afin de mieux mettre en évidence encore les atouts et les limitations de la méthode, nous allons dans nos prochains travaux la tester lors de déplacements du robot HILARE II en environnement intérieur. Les primitives extraites seront alors encore plus nombreuses, et les problèmes d'appariements ambigus et de gestion de la cohérence du modèle se poseront de manière plus critique. Une meilleure connaissance de l'imprécision des mesures proprioceptives de trajectométrie serait également nécessaire (mais ce problème est très difficile; nous avons signalé au paragraphe 4.2 l'impossibilité de modéliser statistiquement les glissements des roues et les dérapages transversaux. Sur un sol peu accidenté, nous avons pu estimer grossièrement les erreurs commises, qui sont de l'ordre de quelques % de la distance parcourue. Sur un terrain accidenté, il est extrêmement difficile d'estimer les performances de trajectométrie). Nous travaillons aussi maintenant sur plusieurs améliorations du système de localisation.

Tout d'abord, nous voulons introduire dans chacun des modèles de localisation plus d'information sémantique. Cette évolution apparaît nécessaire pour:

- *Structurer le modèle géométrique.* Par exemple, le petit obstacle polyédrique (carré en vue de dessus) dont le robot a fait le tour – figures 5.26 à 5.28 – est constitué de quatre segments formant un contour (à peu près) fermé. Il sera beaucoup plus fiable et efficace, pour la localisation du robot, d'utiliser l'information "obstacle polyédrique carré" dans un processus du type "reconnaissance d'objet" que d'utiliser simplement l'ensemble des segments qui se trouvent dans la région de l'environnement qu'il occupe (ceci sera d'autant plus intéressant que le "modèle" de l'amer à reconnaître sera

plus discriminant). On pourrait également introduire la notion de “groupe d’amers”; dans le modèle “points” par exemple, les trois arbres détectés à plusieurs reprises dans l’environnement (cf. figures 5.12, 5.20 ou 5.22 notamment) forment un triangle rigide, dont la forme pourrait être utilisée pour contraindre le processus d’appariement de primitives, et renforcer la prédiction et la certitude des appariements réalisés, dans une exploitation plus *dynamique* des amers détectés. L’introduction d’une information sémantique plus importante dans les modèles de localisation devrait ainsi nous conduire à une *mesure de certitude plus robuste* que celle que nous utilisons actuellement, et permettre d’utiliser *explicitement* la configuration, ou la périodicité des amers sur le terrain). Pour le moment, la position relative de plusieurs amers est utilisée de manière *implicite* dans la procédure d’appariement (cf. paragraphe 5.1.2).

- *Gérer la cohérence globale du modèle.* Nous avons déjà évoqué la nécessité de cette gestion au paragraphe 1.3 notamment. D’autres éléments peuvent être considérés ici. Nous venons de voir que des critères sémantiques permettent une mesure plus robuste de la certitude des amers détectés ou des appariements effectués. Associer une information sémantique à un ensemble de segments, peut également permettre d’abandonner des segments qui n’apportent pas d’information supplémentaire sur un objet, comme de petits segments parasites.

Par ailleurs, nous voulons étendre et améliorer la coopération entre les différentes méthodes de localisation.

- **Sur un plan numérique,** la position d’une caractéristique dans un des modèles peut être affinée si la même caractéristique peut être extraite, avec une plus grande précision, dans l’autre modèle. La position de l’extrémité d’un segment dans le modèle “segments” peut être affinée par la connaissance de la position de l’arête verticale (extrémité du mur) dans le modèle “points”, ou encore par le calcul de l’intersection entre la droite porteuse du tir laser correspondant et celle du mur. La position 2D d’un angle de bâtiment sera connue de manière beaucoup plus précise si elle est définie comme l’intersection de deux segments que si elle est perçue comme une ligne verticale de discontinuité de profondeur (cf. figure 5.35).
- **Sur un plan sémantique,** les amers contenus dans un des modèles peuvent être utiles pour caractériser ou préciser les attributs des amers contenus dans l’autre modèle. Par ailleurs, une coopération sémantique peut être très utile pour l’interprétation de scènes, par exemple pour la reconnaissance d’amers ou d’objets, car certains amers peuvent être beaucoup mieux décrits si on utilise les différentes caractéristiques extraites. Ainsi, figure 5.36, on peut reconnaître avec une plus grande certitude une porte si on est *à la fois* en présence de deux segments admettant la même droite porteuse (à une tolérance près) et délimitant une ouverture, encadrée par deux lignes de discontinuité verticale, correspondant aux extrémités du segment (et présentant des signes de discontinuité opposés suivant un angle de vue donné).

Ainsi, la mise en correspondance d'amers compris dans un des modèles avec les amers compris dans l'autre peut permettre de confirmer l'existence des amers détectés et de renforcer la cohérence du modèle.

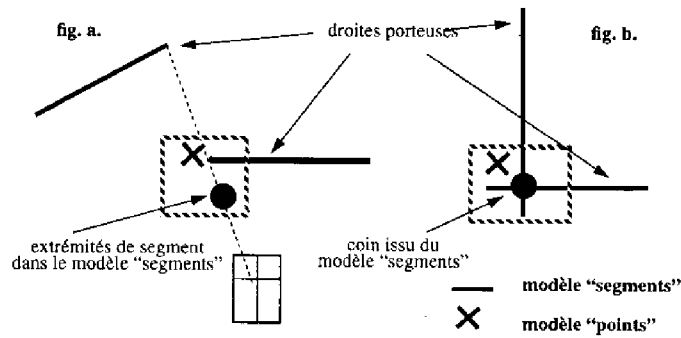


Figure 5.35: *Modèles hétérogènes: fusion*

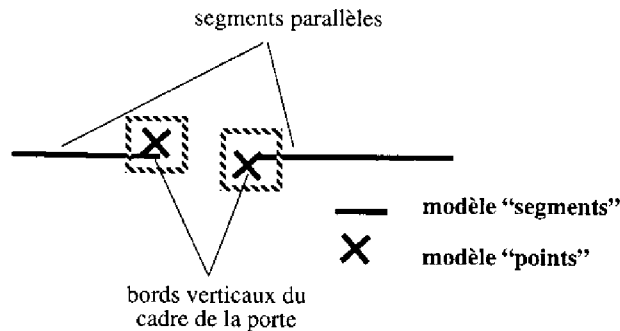


Figure 5.36: *Reconnaissance d'amers: exemple de la porte*

Enfin, nous voulons étendre dans nos travaux futurs la notion de modèle hétérogène. La panoplie de méthodes de modélisation géométrique que nous possédons maintenant dans l'équipe nous incite à généraliser la construction de modèles du monde hétérogènes pour différentes tâches: localisation comme ici, interprétation, ... Par exemple, le modèle polyédrique de l'environnement construit par F. Nashashibi et M. Devy [Nashashibi 93b, Nashashibi 93c] pourrait nous permettre de ne retenir parmi les segments du modèle "murs" que ceux qui ont été effectivement appariés avec des plans verticaux. Une coopération entre tous les modèles disponibles permettrait ainsi de faire un grand pas dans la réponse à une question difficile en modélisation: quelle information conserver ?

## Partie III

# Quand le milieu est non structuré



## Introduction à la partie III

Dans cette partie, nous nous intéressons au cas où l'environnement du robot est non structuré. Du point de vue de la localisation, qui est une des problématiques centrales de notre étude, la différence avec le contexte de la partie II est qu'on ne peut compter sur la présence dans l'environnement d'amers de forme structurée. Pourtant, le terrain peut présenter des zones assez caractéristiques pour permettre le recalage du robot (rocher particulier par exemple), et nous allons utiliser pour le recalage des caractéristiques du terrain.

Une première question qui se pose est celle de la représentation de telles caractéristiques, et donc de la modélisation de terrain. Pour celui-ci, nous faisons l'hypothèse fondamentale que *l'altitude  $z$  est une fonction de la position ( $z = f(x, y)$ )*: autrement dit, la surface du terrain ne se replie pas sur elle-même.

Dans le chapitre 6, nous présentons une méthode de modélisation de terrain à l'aide d'une hiérarchie de surfaces B-splines.

Nous montrons au cours de cette partie que l'on peut adapter l'utilisation de ce modèle en fonction de la tâche à effectuer. Dans le chapitre 6, on cherche à obtenir une représentation de la surface du terrain avec une certaine précision. Dans le chapitre 7, nous dédions l'utilisation du modèle à la fonction *localisation*. Nous montrons comment le modèle permet la focalisation de la recherche de caractéristiques du terrain pour le recalage. La méthode de recalage que nous présentons ensuite fait coopérer ce modèle avec d'autres représentations du terrain.

En conclusion, nous discutons enfin de la possibilité de dédier le modèle de terrain à d'autres tâches, comme la *navigation*.





## Chapitre 6

# Le Modèle du Terrain

Nous présentons dans ce chapitre la méthode de modélisation de terrain à l'aide de surfaces B-splines, que nous avons développée.

### 6.1 B-splines: le formalisme adopté

La théorie des splines <sup>1</sup> est bien établie. Nous avons expliqué dans le chapitre 3 les raisons qui nous ont amené à nous intéresser à la représentation de surfaces à l'aide de splines, et à utiliser pour cela le formalisme particulier des *B-splines*. Ce formalisme, pour lequel les roboticiens montrent un intérêt croissant depuis quelques années, est plutôt utilisé traditionnellement en C.A.O.<sup>2</sup> [Fillatreau 91]. Dans ce paragraphe, il nous semble donc nécessaire pour la bonne compréhension de ce chapitre de présenter un certain nombre de notions fondamentales à propos des B-splines. Le lecteur intéressé peut trouver plus de détails dans [Ahlberg 67, Bartels 87, Bohm 84, De Boor 78, Faux 79, Mortenson 85, Pavlidis 81, Risler 91].

Pour modéliser la surface du terrain accidenté, nous utiliserons des surfaces B-splines dites *de produit tensoriel*, définies comme produit tensoriel de deux courbes B-splines. Nous allons donc tout d'abord présenter les courbes, puis les surfaces B-splines.

#### 6.1.1 B-splines en dimension 1, courbes B-splines

Le terme *B-spline* désigne une spline de *base*. Dans ce qui suit, tout polynôme par morceaux  $P(x)$  ( $x$  variable réelle) de degré  $m$ , pourra s'écrire de manière unique comme une combinaison linéaire sur une base de B-splines de degré  $m$ , qui sont des splines particulières, non nulles sur un horizon fini, et définies sur un découpage de l'intervalle de variation de la variable  $x$  (délimitant les segments de la courbe  $y = P(x)$ ).

<sup>1</sup>Le terme *spline* désigne une fonction polynomiale par morceaux. Nous parlerons de courbe spline ou de surface spline pour désigner une courbe ou une surface polynomiales par morceaux.

<sup>2</sup>Conception Assistée par Ordinateur

Le tableau suivant et la figure 6.1 permettent de définir les notations que nous avons adoptées.

Notations et définitions relatives aux B-splines	
$x$	variable réelle
$\{x_i\}_{i \in \{0, \dots, k+2m\}}$	séquence des <i>noeuds</i> . Plusieurs noeuds successifs peuvent être confondus. Si $i$ noeuds successifs sont confondus, on parle d'un noeud de <i>multiplicité</i> $i$
$\{X_i\}_{i \in \{0, \dots, k\}}$	séquence des valeurs distinctes prises par les <i>noeuds</i>
$\{(X_i, Y(X_i))\}_{i \in \{0, \dots, k\}}$	séquence des <i>points de raccordement</i> , souvent aussi appelés <i>noeuds</i> par abus de langage <sup>3</sup>
$B_{i,m}(x)$	B-spline de degré $m$ non nulle sur $[x_i, x_{i+m+1}]$
$B_{i,j}(x, y) = B_{i,m_x}(x) \times B_{j,m_y}(y)$	B-spline de produit tensoriel, de degrés $m_x$ et $m_y$ suivant $x$ et $y$ respectivement, non nulle sur $[x_i, x_{i+m_x+1}] \times [y_j, y_{j+m_y+1}]$

Etant donnée une séquence (ordonnée) de noeuds  $\{x_i\}$  divisant l'intervalle  $[a, b]$  de variation de la variable  $x$ , on peut définir une base de fonctions  $B_{i,m}$  de l'espace vectoriel des fonctions sur  $[a, b]$ , polynômiales de degré  $m$  par morceaux. L'ordre de continuité aux noeuds (noté  $C^i$  si les dérivées sont continues jusqu'à l'ordre  $i$ ) dépend de la multiplicité des noeuds.

Les fonctions constituant la base de B-splines sont définies pour tout réel  $x$ , par récurrence sur le degré  $m$ , de la façon suivante:

$$\mathbf{B}_{i,0}(\mathbf{x}) = \begin{cases} 1 & \text{si } x_i \leq x \leq x_{i+1} \\ 0 & \text{sinon} \end{cases} \quad (6.1)$$

$$\mathbf{B}_{i,m}(\mathbf{x}) = \frac{x - x_i}{x_{i+m} - x_i} \mathbf{B}_{i,m-1}(\mathbf{x}) + \frac{x_{i+m+1} - x}{x_{i+m+1} - x_{i+1}} \mathbf{B}_{i+1,m-1}(\mathbf{x}) \quad (6.2)$$

Si on choisit pour tous les noeuds "internes" ( $X_1$  à  $X_{k-1}$ ) une multiplicité égale à 1, et une multiplicité égale à  $m+1$  pour les noeuds extrêmes, chaque B-spline de degré  $m$  est  $C^{m-1}$ . Dans nos applications, nous utilisons en permanence des splines  $C^{m-1}$ , et faisons donc ce choix pour les multiplicités des noeuds. La base de B-splines est alors composée de  $k+m$  fonctions. La séquence de noeuds  $\{x_i\}$  ainsi définie, tout polynôme  $y = C(x)$  de degré

<sup>3</sup>Cet abus de langage est fréquent dans la littérature spécialisée. Nous y ferons parfois appel par commodité, s'il n'y a pas d'ambiguïté

$m$  avec des points de raccordement de continuité  $C^{m-1}$  sur  $[a, b]$  peut s'écrire de manière unique comme combinaison linéaire des B-splines de base:

$$y = C(x) = \sum_{i=0}^{k+m-1} a_i B_{i,m}(x) \quad (6.3)$$

La figure 6.1 montre la base de B-splines correspondante dans le cas où  $m = 1$ . Les noeuds aux deux extrémités de  $[a, b]$  sont de multiplicité 2, et les autres de multiplicité 1. Chaque B-spline  $B_{i,1}$  est non nulle sur l'intervalle  $[x_i, x_{i+2}]$ . La figure 6.2 montre des B-splines de degré 3.

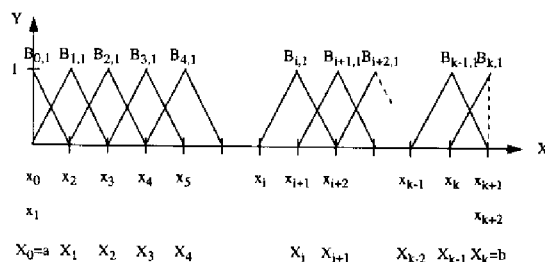


Figure 6.1: Base de B-splines de degré 1

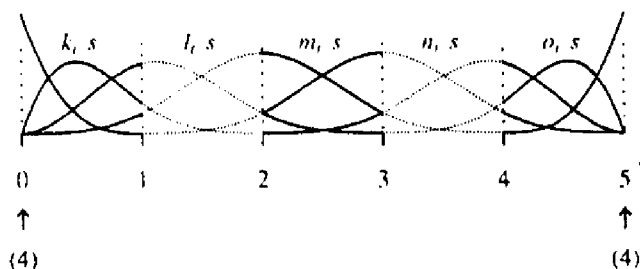


Figure 6.2: Base de B-splines de degré 3

Par ailleurs, dans le cas général, les B-splines possèdent notamment les propriétés suivantes [Risler 91]:

1.  $B_{i,m}(x)$  est polynôme de degré  $m$  par morceaux.
2.  $B_{i,m}(x) = 0$  pour  $x \notin [x_i, x_{i+m+1}]$ .
3.  $\begin{cases} B_{i,m}(x) > 0 \text{ pour } x \in ]x_i, x_{i+m+1}[ \\ B_{i,m}(x_i) = 0 \text{ sauf si } x_i \text{ correspond à un noeud de multiplicité } m+1, \text{ auquel} \\ \text{cas } B_{i,m}(x_i) = 1 \end{cases}$
4.  $\forall x \in [a, b], \sum_i B_{i,m}(x) = 1$

*Remarque: la propriété 4. ci-dessus n'est vraie qu'à l'abus de langage près signalé dans [Risler 91] et consistant à poser:  $B_{k+m-1,m}(b) = 1$ . L'exemple de la figure 6.1 illustre la légitimité de cet abus de langage, qui s'exprime dans ce cas par  $B_{k,1}(b) = 1$ .*

A partir de ces propriétés, signalons que:

- La séquence des  $x_i$  étant donnée,  $k + m$  coefficients suffisent à définir un polynôme par morceaux de raccords  $C^{m-1}$  (avec le choix des multiplicités des noeuds adopté au début de ce paragraphe) sur  $[a, b]$ .
- La modification d'un des coefficients  $a_i$  n'a d'influence que sur  $m+1$  intervalles, chaque B-spline étant non nulle sur  $m + 1$  intervalles. Lorsque l'on représente une courbe ou surface spline à l'aide de B-splines, des modifications locales ne se propagent pas, propriété que d'autres représentations de splines ou de courbes ou surfaces composites (celles de Bézier par exemple) ne possèdent pas (ou pas de manière aussi explicite).
- La multiplicité des noeuds agit sur l'ordre de continuité au raccordement correspondant. L'ajout d'une unité à la multiplicité d'un noeud fait tomber l'ordre de continuité de la spline au raccordement correspondant de 1. La figure 6.3 illustre ce phénomène pour une B-spline cubique.

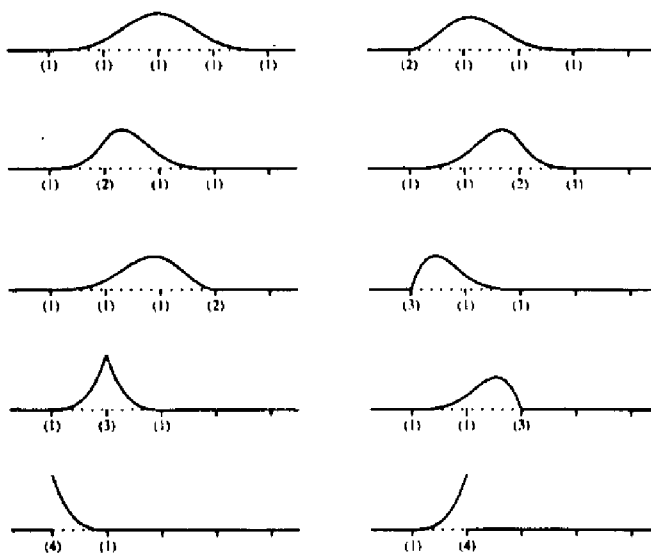


Figure 6.3: Multiplicité des noeuds et ordre de continuité

### 6.1.2 Surfaces B-splines de "produit tensoriel"

#### Surfaces de produit tensoriel

Une présentation théorique de ce type de surface peut être trouvée dans [De Boor 78]. Nous reprenons ici la présentation plus intuitive donnée dans [Bohm 84].

Soit:

$$\mathbf{x}(\mathbf{u}) = \sum_{i=0}^n \mathbf{c}_i \mathbf{F}_i(\mathbf{u}) \quad (6.4)$$

une courbe (2D ou 3D) exprimée comme une combinaison linéaire sur une base de fonctions  $\{\mathbf{F}_i\}$ .

Faisons décrire à cette courbe une surface, en la déplaçant dans l'espace. Le balayage effectué peut être décrit par le fait que chaque coefficient  $\mathbf{c}_i$  suit une courbe  $\mathbf{c}_i(v)$ . Si toutes ces courbes  $\mathbf{c}_i(v)$  s'expriment comme combinaisons linéaires de fonctions de base  $\mathbf{G}_k(v)$ :

$$\mathbf{c}_i(\mathbf{v}) = \sum_{k=0}^m \mathbf{a}_{i,k} \mathbf{G}_k(\mathbf{v}) \quad (6.5)$$

alors la surface résultante  $S(u, v)$  est appelée surface de produit tensoriel:

$$\mathbf{x}(\mathbf{u}, \mathbf{v}) = \sum_i \mathbf{c}_i(\mathbf{v}) \mathbf{F}_i(\mathbf{u}) = \sum_i \sum_k \mathbf{a}_{i,k} \mathbf{F}_i(\mathbf{u}) \mathbf{G}_k(\mathbf{v}) \quad (6.6)$$

Les *produits tensoriels*  $\mathbf{F}_i(u)\mathbf{G}_k(v)$  forment une base de fonctions pour ces surfaces.

### Surfaces B-splines de produit tensoriel

Par analogie, une surface B-splines de produit tensoriel est définie comme le produit tensoriel de deux courbes B-splines. Cette fois, deux vecteurs  $(\{x_i\})$  et  $(\{y_j\})$  définissent un réseau rectangulaire de noeuds  $(x_k, y_l)$  et un réseau de carreaux sur la surface. Comme dans le cas des courbes, on peut définir un ensemble de fonctions formant une base de l'espace vectoriel des surfaces  $S(x, y)$  polynômiales par morceaux sur le réseau de carreaux, de degré  $m_x$  suivant  $x$  et  $m_y$  suivant  $y$  (la surface a une expression polynômiale donnée sur chaque carreau). Les multiplicités des abscisses ou ordonnées des noeuds permettent de contrôler l'ordre de continuité de la surface le long des frontières des carreaux dans les directions  $x$  et  $y$ . Si tous les  $x_i$  et  $y_j$  "intérieurs" sont de multiplicité 1, et ceux situés aux extrémités suivant  $x$  et  $y$  sont de multiplicité  $m_x + 1$  et  $m_y + 1$  respectivement, la surface est  $C^{m_x-1}$  suivant  $x$  et  $C^{m_y-1}$  suivant  $y$ , et peut s'exprimer comme combinaison linéaire de  $(k_x + m_x)(k_y + m_y)$  B-splines de base:

$$\mathbf{z} = \mathbf{S}(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{k_x+m_x-1} \sum_{j=0}^{k_y+m_y-1} \mathbf{a}_{i,j} \mathbf{B}_{i,m_x}(\mathbf{x}) \mathbf{B}_{j,m_y}(\mathbf{y}) \quad (6.7)$$

ou encore:

$$\mathbf{z} = \mathbf{S}(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{k_x+m_x-1} \sum_{j=0}^{k_y+m_y-1} \mathbf{a}_{i,j} \mathbf{B}_{i,j}(\mathbf{x}, \mathbf{y}) \quad (6.8)$$

où  $B_{i,j}(x, y) = B_{i,m_x}(x) \times B_{j,m_y}(y)$ .

Remarques:

- Une B-Spline de degrés  $(m_x, m_y)$  et de continuité  $C^{m_x-1}$  suivant  $x$  et  $C^{m_y-1}$  suivant  $y$  est non nulle sur  $(m_x + 1)(m_y + 1)$  carreaux (définis par  $\{x_i, x_{i+1}, \dots, x_{i+m_x+1}\}$  et  $\{y_i, y_{i+1}, \dots, y_{i+m_y+1}\}$  suivant  $x$  et  $y$  respectivement).
- Seules  $(m_x + 1)(m_y + 1)$  B-Splines de base sont non nulles sur chaque carreau.

La figure 6.4 montre des exemples de B-splines bicubiques.

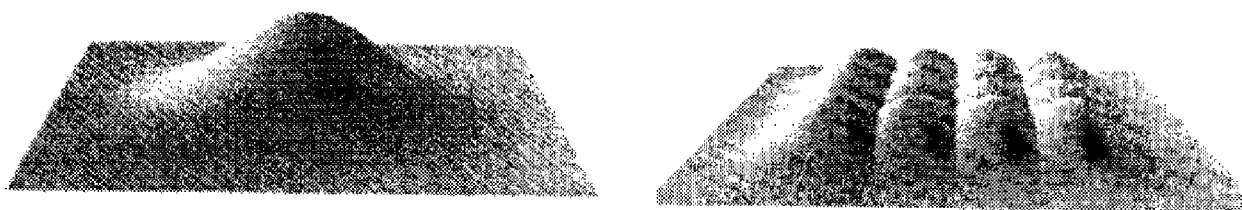


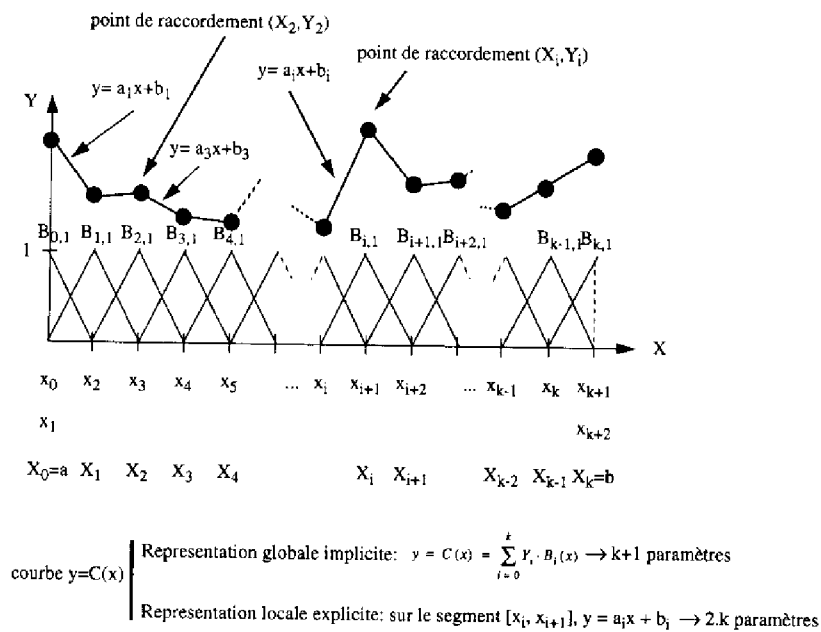
Figure 6.4: *B-splines bicubiques*

Au total, on obtient pour une surface B-splines  $S(x, y)$  une représentation qui peut être manipulée sous deux formes différentes:

- D'une part, nous avons une forme **matricielle globale**, et sous laquelle la surface est décrite par la matrice de ses coefficients sur la base de B-splines. Cette représentation peut être qualifiée d'**implicite**. Elle présente l'avantage d'être très compacte. (De manière analogue, une courbe B-splines peut être décrite par le vecteur de ses coefficients sur la base de B-splines monodimensionnelles).
- D'autre part, on peut recalculer localement l'**expression polynômiale explicite** de la surface sur tout carreau  $(i, j)$  délimité par les noeuds  $(X_i, Y_j)$ ,  $(X_i, Y_{j+1})$ ,  $(X_{i+1}, Y_j)$  et  $(X_{i+1}, Y_{j+1})$ . Les formules 6.1 et 6.2 permettent d'établir les expressions polynômiales explicites des B-splines de base; ces expressions sont par ailleurs bien connues dans les cas d'utilisation courante où le degré est inférieur ou égal à trois. On utilise alors sur un carreau  $(i, j)$  les expressions analytiques des  $(m_x + 1)(m_y + 1)$  B-splines qui sont non nulles sur celui-ci, et les coefficients  $a_{i,m}$  correspondants. (De manière analogue, on peut retrouver sur les segments successifs d'une courbe B-splines son expression analytique explicite).

Cette description peut s'avérer utile pour la planification de trajectoires d'un robot mobile sur terrain accidenté [Shiller 91, Siméon 93, Dacre Wright 93], tâche pour laquelle on peut avoir besoin d'utiliser les équations analytiques des portions de terrain sur lesquelles le robot évolue.

L'exemple de la figure 6.5 illustre le caractère compact de la représentation matricielle ou vectorielle, et la comparaison entre les deux formes (globale implicite ou locale explicite) sous laquelle une courbe ou surface B-splines peut être manipulée. Sans perdre de généralité, il montre ce qui se passe pour une courbe B-splines de degré 1. La spline correspondante est décomposée en  $k$  morceaux linéaires. Chacun de ces morceaux peut être défini par son expression polynômiale explicite, du type  $y = a_i x + b_i$  sur l'intervalle  $i$ . Cela donne  $2 \times k$  paramètres. Mais l'écriture des  $k - 1$  conditions de continuité de la spline aux noeuds  $\{X_1, X_2, \dots, X_{k-1}\}$  réduit le nombre de degrés de liberté de la spline (c'est à dire le nombre de paramètres nécessaires pour la définir entièrement) à  $2 \times k - (k - 1)$  soit  $k + 1$  (à comparer avec le résultat  $k + m$  donné plus haut dans le cas d'une spline  $C^{m-1}$  de degré  $m$ ). On peut observer en effet que la même spline peut être entièrement donnée par la seule donnée des  $k + 1$  valeurs de l'ordonnée  $y$  aux  $k + 1$  points de raccordement  $\{(X_0, y_0), (X_1, y_1), \dots, (X_k, y_k)\}$ , qui ne sont autres dans ce cas que les coefficients de la spline sur la base de B-splines triangulaires de degré 1 ! Ainsi, si on utilise les expressions polynômiales explicites successives, on a  $2 \times k$  paramètres, alors que la forme globale implicite réduit le nombre de paramètres à  $k + 1$ .



La quantité d'information nécessaire pour définir entièrement une surface B-splines est finalement:

- Les degrés  $m_x$  et  $m_y$  dans les directions  $x$  et  $y$ .
- Le réseau de noeuds définis par les deux vecteurs  $(X_0, X_1, \dots, X_{k_x})$  et  $(Y_0, Y_1, \dots, Y_{k_y})$  (et leurs multiplicités, pour lesquelles les choix adoptés ont été présentés plus haut).



- Les coefficients  $a_{i,j}$  de la surface sur la base de B-splines, avec  $i \in [0 \quad k_x + m_x - 1]$  et  $j \in [0 \quad k_y + m_y - 1]$ .

Le type de représentation d'une surface adopté ici conduit donc à des représentations analytiques compactes. Nous l'utiliserons pour modéliser un terrain.

### 6.1.3 Interpolation ou approximation ?

Si nous devons modéliser un ensemble donné de  $N$  points de mesure  $P_k, k \in \{1, \dots, N\}$ , nous pouvons envisager de déterminer une surface B-splines réalisant une *interpolation* de l'ensemble de ces points (c'est à dire passant par tous les points), ou simplement qui en donne une *approximation* (la surface n'est pas assujettie à passer par tous les points de mesure) optimisant un certain critère de ressemblance (comme le classique critère des moindres carrés).

Pour réaliser une **interpolation**, plusieurs approches ont été étudiées [Pavlidis 81].

Sans nuire à la généralité de ce qui suit, mais pour illustrer les différentes approches, supposons que l'on souhaite interpoler dans le plan  $(x, y)$  un ensemble de  $N$  points  $\{(t_1, y_1), (t_2, y_2), \dots, (t_N, y_N)\}$  à l'aide d'une courbe B-splines de degré  $m$ .

Une première approche consiste à identifier chacun des points à interpoler avec un point de raccordement de la spline. Une spline ayant  $k + m$  degrés de liberté (pour  $k + 1$  valeurs distinctes de noeuds), si  $m$  est petit (ce qui est généralement le cas), on peut choisir le nombre de noeuds de telle sorte que  $k = N - 1$  et identifier les noeuds avec  $t_1, t_2, \dots, t_N$ .

Dans le cas d'une courbe spline linéaire par exemple, il faut  $k+1 = N$  paramètres (coefficients sur la base de B-splines) pour définir la courbe. L'écriture des conditions de passage de la courbe par les  $N$  points de donnée (conditions d'interpolation) conduit à un système linéaire de  $N$  équations à  $N$  inconnues. La courbe est en fait complètement spécifiée par les segments de droite joignant les points de donnée.

Dans le cas d'une courbe spline cubique, on a  $k + 3 = N + 2$  degrés de liberté. L'écriture des conditions d'interpolation des  $N$  points de donnée conduit à un système de  $N$  équations à  $N + 2$  inconnues. Il reste donc deux degrés de liberté, utilisés généralement pour spécifier les tangentes de la courbe aux extrémités.

Une autre approche consiste à intercaler les noeuds de la spline et les points de données, en choisissant le nombre de noeuds de telle sorte que  $k + m = N$  dans le cas où on ne veut pas avoir à fixer d'autres conditions que celles exprimant que la spline interpole les  $N$  points de donnée. Pour que l'unicité de la solution au problème d'interpolation soit assurée, la répartition des noeuds doit obéir à une règle rappelée dans [Pavlidis 81]. Si le point de donnée  $(t_j, y_j)$  est tel que  $x_i \leq t_j \leq x_{i+1}$ , la condition de passage de la spline  $C(x) = \sum_{i=0}^{k+m-1} a_i B_{i,m}(x)$  par le point  $(t_j, y_j)$  s'écrit alors:

$$\mathbf{a}_i \mathbf{B}_{i,m}(t_j) + \mathbf{a}_{i-1} \mathbf{B}_{i-1,m}(t_j) + \dots + \mathbf{a}_{i-m} \mathbf{B}_{i-m,m}(t_j) = y_j \quad (6.9)$$

et l'écriture des conditions de passage par l'ensemble des points de donnée conduit pour la détermination des coefficients de la spline à un système linéaire de  $N$  équations à  $k+m = N$  inconnues, se ramenant à l'inversion d'une matrice bande<sup>4</sup> puisque seules  $m+1$  B-splines sont non nulles sur un segment quelconque de la courbe spline (et, dans nos applications, le degré est petit devant le nombre de points de donnée).

En conclusion, réaliser l'interpolation d'un ensemble de données par une courbe (ou une surface) B-splines suppose:

- d'une part de satisfaire des contraintes sévères sur le choix du nombre et de l'emplacement des noeuds;
- d'autre part, que le nombre de noeuds et de coefficients obtenu est grand si le nombre de points de données à interpoler est lui-même grand.

Dans notre cas, le problème est de modéliser à l'aide d'une surface B-splines, un grand nombre de points de mesure  $\{P_k = (x_k, y_k, z_k), k \in \{1, \dots, N\}\}$  acquis sur la surface du terrain, à l'aide d'un télémètre laser. Nous préférons donc réaliser une **approximation** de ces points de mesure. Il faut alors choisir un réseau de points noeuds (un vecteur dans chacune des directions  $x$  et  $y$ ). Les degrés dans chacune de ces directions étant choisis, ceci définit une base de B-splines. Il faut alors déterminer les coefficients de la surface spline sur cette base, qui minimisent un certain critère d'erreur d'approximation. Si le critère choisi est celui des moindres carrés, on veut déterminer les coefficients  $a_{i,j}$  qui minimisent (avec les notations adoptées ci-dessus):

$$\sum_{k=1}^N \left[ \sum_{i=0}^{k_x+m_x-1} \sum_{j=0}^{k_y+m_y-1} a_{i,j} \mathbf{B}_{i,j}(x_k, y_k) - z_k \right]^2 \quad (6.10)$$

C'est le critère classique que nous avons adopté dans la suite pour ajuster une surface B-splines à nos points de mesure sur un terrain.

Certains travaux rajoutent dans le critère à minimiser des termes supplémentaires, et notamment un terme d'énergie de flexion. Nous reviendrons sur ce point et comparerons les différents critères rencontrés dans le chapitre 7.

Résoudre le problème aux moindres carrés se ramène à la résolution d'un système linéaire bande (car chaque B-spline de degré  $m$  est non nulle sur  $m+1$  intervalles seulement au plus).

<sup>4</sup>On définit la *largeur de bande* d'une matrice  $B = (b_{ij})$  par:  $\max\{|i-j| \text{ tel que } b_{ij} \neq 0\}$ . B est dite "matrice bande" si sa largeur de bande est petite par rapport à la taille de B [Demko 77]

Rajoutons que la procédure que nous utilisons résoud le problème aux moindres carrés en déterminant et résolvant les équations normales [De Boor 78, Grosse 80]. Les équations normales peuvent quelquefois causer des problèmes numériques, mais ici, leur utilisation ne doit pas créer de problèmes car la base de B-splines conduit généralement à des matrices bande bien conditionnées. Précisons enfin que la procédure que nous utilisons permet de fixer des poids différents en fonction des données.

## 6.2 Modèle hiérarchique du terrain

Dans ce paragraphe, nous proposons une méthode de modélisation d'un terrain quelconque à l'aide de surfaces B-splines de produit tensoriel [Fillatreau 93b]. Nous voulons obtenir un modèle qui puisse s'adapter à la grande diversité de relief que le terrain d'évolution du type de robots qui nous intéressent peut présenter (zones plates ou peu irrégulières, zones très accidentées, obstacles de forme quelconque...). Comment alors répartir les noeuds, c'est à dire comment définir le réseau de carreaux composant la surface ? Ceci est notamment important du point de vue des indéterminations possibles pour le calcul des coefficients sur la base de B-splines si certains carreaux de la surface B-splines recouvrent des zones de terrain dans lesquelles il n'y a pas de points de mesure. Comment représenter les zones inconnues ? Une autre question importante, enfin, est celle du choix du degré des B-splines.

### 6.2.1 Réseau de carreaux, multi-résolution et approche hiérarchique

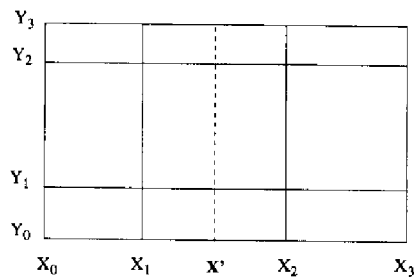
Un premier problème qui se pose lorsqu'on veut modéliser une surface à l'aide de B-splines (ou d'une surface polynômiale par morceaux en général) est celui du choix du réseau de carreaux composant la surface, c'est à dire de la répartition des noeuds, dont nous avons souligné plus haut qu'il s'agit d'un des problèmes essentiels.

Nous savons que dans les zones accidentées ou irrégulières, où des propriétés différentielles de la surface, comme la pente ou la courbure, présentent de fortes variations, une meilleure "résolution" (c'est à dire des carreaux de plus petite taille) est nécessaire [Guéziec 92].

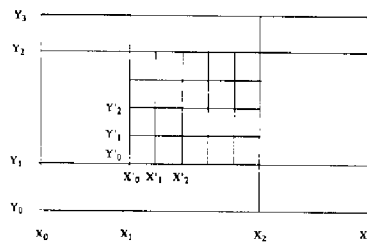
Cependant, utiliser le formalisme des surfaces B-splines entraîne le fait que si on fait une subdivision dans le vecteur de noeuds dans une des directions  $x$  ou  $y$ , tous les carreaux correspondants suivant l'autre direction sont affectés. La figure 6.6 montre l'effet de l'insertion du noeud  $x'$  suivant la direction  $x$ .

Cet effet est indésirable pour la modélisation de terrain (ou d'une surface en général), car nous ne voulons pas "propager" de telles modifications du réseau de noeuds, mais seulement raffiner le modèle localement.

C'est pourquoi nous avons été amenés à adopter une approche *hiérarchique*, dans laquelle chaque niveau a un **réseau uniforme** de carreaux. Nous savons que le choix d'un réseau uniforme n'est pas optimal pour obtenir la meilleure approximation [De Boor 78], mais ceci

Figure 6.6: *Insertion d'un nouveau noeud  $x'$* 

est compensé par le raffinage récursif de la description du terrain, et par ailleurs un réseau uniforme est en général plus facile à gérer ! Chaque carreau à un niveau donné est subdivisé, si nécessaire, en un nouveau réseau de carreaux (cf. figure 6.7). La décision de subdiviser un carreau ou non est prise en fonction d'un critère mesurant la qualité de l'approximation surfacique des points de mesure effectuée sur ce carreau (nous donnons plus loin davantage de détails sur le processus de raffinage).

Figure 6.7: *Subdivision d'un carreau.*

Le choix d'un réseau uniforme se justifie cependant par plusieurs autres raisons:

- Les B-splines de base obtenues à partir d'un réseau uniforme de noeuds prennent une forme beaucoup plus simple que dans le cas d'une répartition non uniforme. Les expressions analytiques des B-splines de degré assez petit sont bien connues. On peut avantageusement les utiliser directement, notamment lorsque se pose le problème de l'évaluation d'une B-spline (ou d'un point d'une surface B-splines) en une position  $(x, y)$  donnée, ou pour tout problème nécessitant de manipuler l'expression analytique de la surface (par exemple lors du calcul d'attributs comme la courbure).
- Nous nous poserons par la suite le problème de construire incrémentalement le modèle du terrain (par exemple pour fusionner l'information correspondant à des acquisitions sensorielles effectuées depuis différentes position du robot). La modélisation incrémentale sera plus aisée dans un réseau uniforme.

### 6.2.2 Choix du degré des B-splines

Le choix du degré de la surface est également un problème important. Il dépend des caractéristiques dont nous voulons doter notre modèle, et de considérations de complexité de calcul.

En ce qui concerne ce dernier point, nous devons observer que si on élève le degré d'une surface B-splines, on augmente le nombre de degrés de liberté de la surface, c'est à dire le nombre de coefficients à déterminer pour un même réseau de noeuds, et donc les temps de calcul induits (cependant, c'est surtout du nombre de noeuds que dépend la complexité – une surface B-splines a  $(k_x + m_x + 1) \times (k_y + m_y + 1)$  degrés de liberté, avec dans notre cas  $k_x \gg m_x$  et  $k_y \gg m_y$  –).

Par ailleurs, si on augmente le degré de la surface, on risque de faire apparaître des oscillations indésirables. Les figures 6.9 et 6.10 montrent deux interpolations d'un même ensemble de données de terrain (présenté figure 6.8). Le réseau de carreaux est à chaque fois défini en intercalant les noeuds de la spline et les points de donnée [Pavlidis 81] (voir présentation au paragraphe 6.1.3 ci-dessus). L'interpolation est réalisée par une surface B-splines bi-cubique dans le premier cas, et "bi-degré 7" dans le deuxième cas. Ce dernier, volontairement caricatural (il est hors de question que nous utilisions un tel degré pour modéliser le terrain), fait clairement apparaître des oscillations importantes.

Le choix du degré est également dicté par le souci d'obtenir un modèle adapté au type de surface que l'on veut représenter, et l'utilisation que l'on souhaite pouvoir faire de ce modèle, notamment le calcul de certains attributs pour caractériser telle ou telle portion de la surface. Il est classique de calculer par exemple des caractéristiques comme la courbure. Dans un tel cas, nous devons utiliser des surfaces de degré trois au moins si nous voulons obtenir une courbure continue, de degré 2 sinon. Nous montrons figures 6.11 et 6.12 deux approximations par surfaces B-splines du même ensemble de données de terrain, avec des carreaux de taille  $4 \times 4$  en termes de ces données (1 point de mesure sur 4 correspond à un noeud), avec des surfaces bi-quadratiques et bi-cubiques respectivement. Les surfaces obtenues sont assez semblables visuellement, mais dans le second cas la courbure est continue.

Nous avons gardé ces considérations présentes à l'esprit lors des différents choix que nous avons dû faire pour les différents niveaux du modèle.

### 6.2.3 Construction du modèle hiérarchique

Dans ce paragraphe, nous présentons la procédure de calcul du modèle de terrain. Nous décrivons les choix faits pour le niveau supérieur, puis le processus de raffinement récursif du modèle.

#### A/ Niveau supérieur du modèle

Le niveau supérieur du modèle hiérarchique de terrain en donne la description la plus grossière.

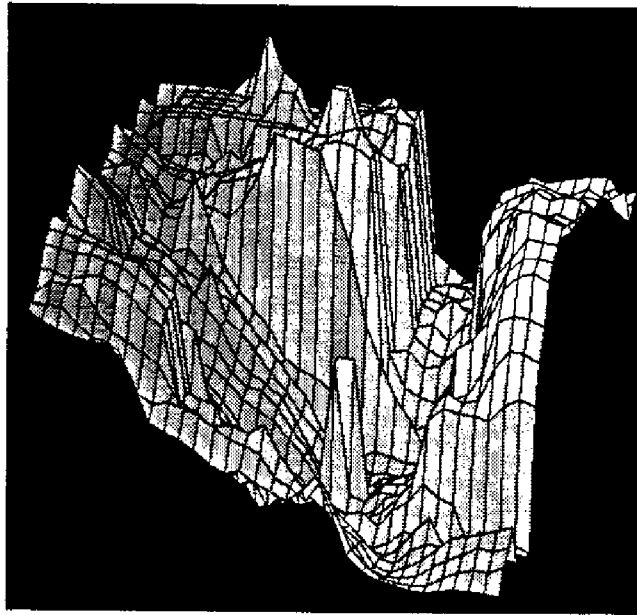


Figure 6.8: *Données MNT à modéliser*

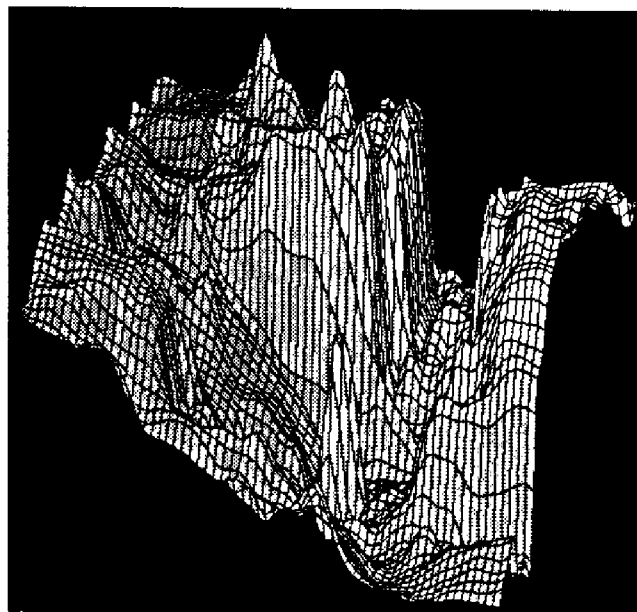


Figure 6.9: *Interpolation bicubique*

Le degré de la surface et la taille des carreaux doivent être adaptés à la modélisation des zones de l'environnement les plus simples à modéliser. Nous avons choisi d'utiliser à ce niveau des splines de degré peu élevé. En pratique, nous avons choisi des B-splines bi-linéaires, qui sont bien adaptées pour représenter les régions planes ou peu accidentées de l'environnement.

La taille des carreaux doit résulter d'un compromis entre le nombre de coefficients à déterminer (pour l'ensemble du modèle hiérarchique, c'est au niveau supérieur que se fait, de ce

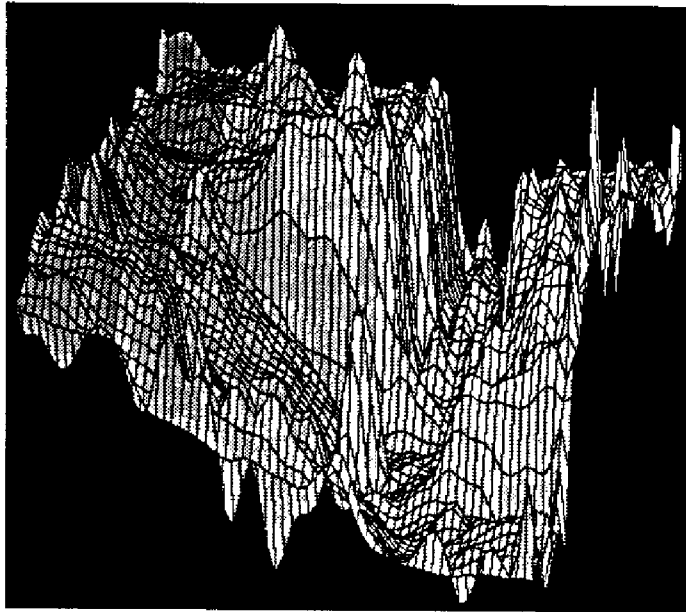


Figure 6.10: *Interpolation bi-degré 7*

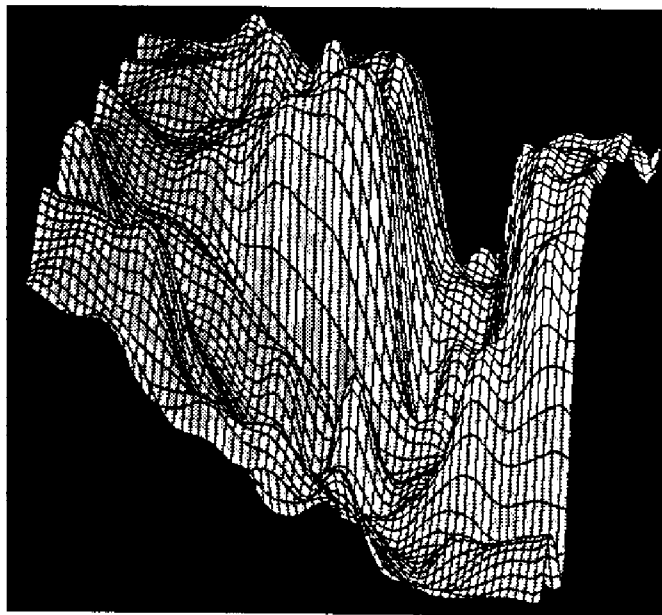
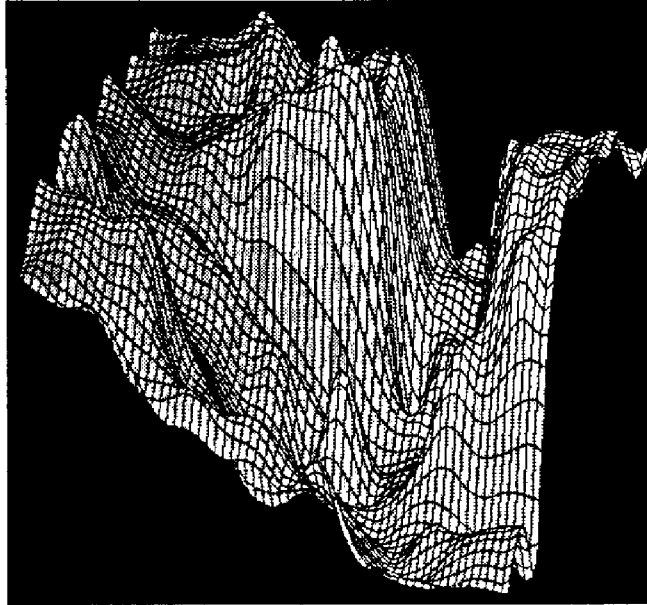


Figure 6.11: *Approximation biquadratique*

point de vue-là, le calcul d'approximation le plus lourd) et la précision du modèle obtenu (plus les carreaux sont grands, et plus la surface obtenue est lisse et ne peut représenter les accidents du terrain; les B-splines constituent des filtres spatiaux passe-bas !). Nous donnerons lors de la présentation des résultats expérimentaux les choix standards que nous avons faits en pratique.

Figure 6.12: *Approximation bicubique*

### B/ Construction du modèle hiérarchique

Pendant le calcul du niveau supérieur du modèle (approximation de l'ensemble des données), l'erreur aux moindres carrés commise sur chaque carreau de la surface est également évaluée. Si, pour un carreau donné, cette erreur est supérieure à un seuil, on définit un nouveau réseau de carreaux (de plus petite taille) sur ce carreau, et une nouvelle représentation locale (surface B-splines ajustée aux moindres carrés aux données situées sur ce carreau) est calculée. On raffine ainsi le modèle de terrain de manière récursive, chaque carreau étant subdivisé par un nouveau réseau de carreaux uniformes si son erreur associée est trop importante. Le processus de calcul récursif du modèle est arrêté si la totalité des données surfaciques ont été modélisées avec une précision suffisante, ou si une taille minimale autorisée pour les carreaux a été atteinte.

Les paramètres d'entrée de l'algorithme de construction du modèle de terrain sont:

$m_x[nb-niveaux], m_y[nb-niveaux]$	degrés de la surface pour les différents niveaux;
taille-carreaux[nb-niveaux]	vecteur des tailles de carreaux niveau par niveau;
taille-min-carreaux	taille minimale admise pour les carreaux;
seuil-raffinage	seuil sur l'erreur d'approximation sur un carreau.



L'algorithme de construction du modèle de terrain repose dans l'appel récursif de la fonction MODELISE suivante (et appelée initialement avec  $NIV = 0$ ,  $CARREAU =$  environnement perçu complet et  $POINTS-DE-MESURE =$  ensemble des mesures):

MODELISE(NIV, CARREAU, POINTS-DE-MESURE)

**Debut**

- Créer une structure CARTE\_B-SPLINE, avec pour paramètres  $m_x[NIV]$ ,  $m_y[NIV]$  et  $taille-carreaux[NIV]$ .
- Calculer la surface B-splines donnant une approximation aux moindres carrés des points de donnée de CARREAU, et l'erreur d'approximation sur chaque carreau de la surface.
- Construire la liste des carreaux Liste-carreaux-à-raffiner pour lesquels l'erreur est  $>$  seuil-raffinage.
- si (Liste-carreaux-à-raffiner est vide) **ou** ( $taille-carreaux[NIV+1] <$   $taille-min-carreaux$ )  
alors Aller à **Fin**
- **sinon**
  - pour** chaque carreau C de Liste-carreaux-à-raffiner
    - Ranger dans POINTS-DE-DONNEE les points de mesure situés sur C
    - Affecter à C le pointeur retourné par MODELISE(NIV+1,C,POINTS-DE-DONNEE)

**Fin:** retourner le pointeur sur la CARTE\_B-SPLINE construite.

Le modèle finalement obtenu est un arbre, dont chaque noeud correspond à une surface B-splines. La figure 6.13 illustre le modèle hiérarchique proposé.

A chaque noeud du modèle, on stocke (à l'aide de la structure CARTE\_B-SPLINE) l'information suivante:

- le réseau de noeuds;
- les coefficients sur la base de B-splines;
- les attributs de chaque carreau (qui peuvent être les erreurs de modélisation ou d'autres attributs (cf chapitre 7).
- pour chaque carreau raffiné, un pointeur vers la nouvelle surface B-splines raffinée correspondante, pour laquelle nous stockerons la même information que celle ci-dessus.

A chaque niveau du modèle (la définition de la notion de *niveau* étant illustrée par la figure 6.13: elle désigne pour un noeud sa profondeur dans l'arbre) nous pouvons choisir le degré de la surface dans chacune des directions  $x$  et  $y$ , et la taille du réseau de carreaux.

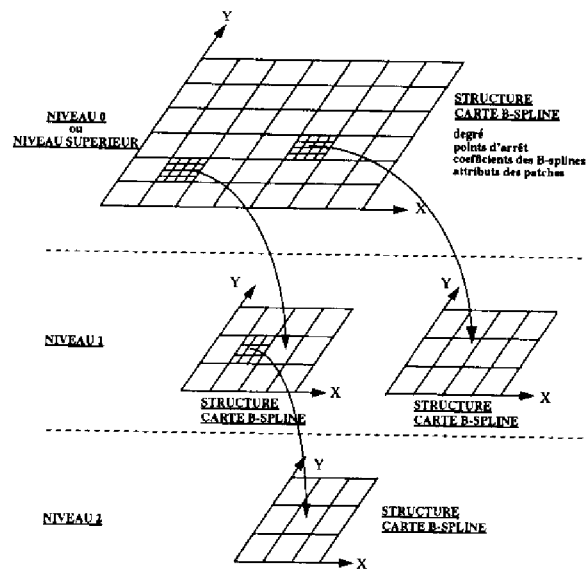


Figure 6.13: *Modèle surfacique B-splines hiérarchique*

En pratique, une profondeur du modèle hiérarchique à 2 ou 3 niveaux est amplement suffisante, dans la plupart des cas, pour obtenir une description assez précise du terrain, si la résolution au niveau le plus profond est suffisamment fine. Le nombre de niveaux et le choix de la taille minimale de carreaux est fonction, dans le cas général, de la tâche à accomplir. Il est dicté dans ce chapitre-ci par la précision souhaitée pour la modélisation de terrain, mais son choix sera autre au chapitre 7 pour des besoins de localisation. Pour la navigation, la taille minimale des carreaux correspond à la taille des surfaces d'étude du contact roue/sol ( $\sim 400\text{cm}^2$  est un choix standard).

De même, des degrés inférieurs ou égaux à 3 dans chaque direction seront le plus souvent suffisants, compte tenu des contraintes évoquées ci-dessus.

#### Raffinage géométrique ou dépendant de la tâche à accomplir ?

Une remarque importante doit être faite à ce stade. La procédure de modélisation proposée dans ce chapitre raffine la représentation du terrain selon un critère d'erreur d'approximation uniquement. Cela signifie que l'on cherche à représenter avec une précision donnée *toutes* les régions du terrain. C'est donc une modélisation purement géométrique qui est faite pour le moment. Au chapitre 7, nous verrons comment le même type de modèle permet d'introduire des aspects *sémantiques* dans l'analyse du terrain. Alors, seules des zones bien déterminées seront raffinées. Ces zones correspondent à des régions d'intérêt, sélectionnées en fonction de la *tâche* à accomplir; dans le chapitre 7, nous nous attacherons essentiellement à une utilisation *dédiée à la localisation* du modèle de terrain.

### Continuité sur un niveau et discontinuité entre les niveaux

Nous avons souligné dans le chapitre 3 que représenter une surface à l'aide de B-splines présente l'avantage considérable de garantir un certain ordre de continuité entre les carreaux constituant la surface, et ceci sans avoir à manipuler explicitement de conditions aux limites de ces carreaux.

Bien évidemment, si nous raffinons la représentation sur un carreau à un niveau donné, nous allons perdre la continuité de la surface entre le carreau raffiné et ses voisins au niveau plus grossier. Ce phénomène, induit par une utilisation non habituelle des B-splines (par rapport aux utilisations les plus couramment faites en CAO) peut dans certains cas être néfaste, et rendre nécessaires des processus d'ajustement des frontières des carreaux pour rétablir la continuité de la surface. . . et de manipuler à nouveau des conditions aux limites des carreaux [Wang 92b], ce que nous cherchions justement à éviter en adoptant le formalisme des B-splines !

Cependant, en fonction de la tâche à accomplir, les différents niveaux peuvent jouer des rôles différents; ce sera le cas au chapitre 7. Nous ne chercherons pas alors à rétablir de continuité entre des niveaux différents, qui ne correspondent pas au même degré de précision de la description (en termes de résolution de la grille de carreau) de la surface; la continuité d'une surface B-splines à un niveau donné est, pour sa part, garantie. Pour utiliser le modèle à des fins de navigation, on préfère demander une modélisation plus précise afin de rendre ces discontinuités négligeables.

### Caractérisation de l'incertitude du modèle

Pendant la construction du modèle hiérarchique du terrain, l'incertitude associée est évaluée. En effet, nous savons représenter l'incertitude des points de mesure à modéliser, et nous pouvons propager cette incertitude à travers la résolution du système linéaire donnant les coefficients de la surface sur la base de B-splines (le formalisme de propagation a été présenté au chapitre 4). L'incertitude de la représentation matricielle (coefficients sur la base de B-splines) est caractérisée par les variances et covariances de ses différents éléments (c'est à dire des coefficients de la surface). Ces coefficients fournissent directement les incertitudes des coefficients de l'expression polynômiale explicite d'un carreau quelconque de la surface, et donc, l'incertitude sur l'altitude d'un point quelconque de la surface.

#### **6.2.4 Résultats expérimentaux**

Nous allons présenter ici quelques résultats expérimentaux obtenus en appliquant la méthode de modélisation que nous venons de présenter. Nous avons utilisé comme données de départ des cartes d'élévation [Nashashibi 92, Nashashibi 93a] (voir chapitres 1 et 3), construites à partir d'images de profondeur réelles d'un terrain non structuré du type de celui présenté figure 6.14.

Rappelons qu'une carte d'élévation donne l'altitude du terrain en tout point d'une grille régulière  $(x, y)$ . Une étiquette binaire indique si un point donné de la carte d'élévation ap-



Figure 6.14: *Environnement expérimental simulant des rochers martiens*

partient à une zone connue (zone perçue, dans laquelle on a interpolé les données sensorielles brutes) ou non. Cette information est stockée dans une carte de *bits*.

Le problème de l'utilisation directe des données sensorielles brutes (qui sont anisotropes) va maintenant être étudié dans notre équipe. Il est essentiellement lié à la gestion des zones inconnues et au problème des indéterminations possibles dans la détermination des coefficients de la surface sur une base de B-splines, si des carreaux sont "vides". On éviterait ainsi l'utilisation d'une carte intermédiaire. L'utilisation d'une carte d'élévation comme support à la modélisation à l'aide de B-splines nous permet de résoudre ce problème puisque la carte de bits donne l'emplacement des zones inconnues. Aux points qui se trouvent dans ces zones sont affectées des valeurs d'élévation correspondant à l'élévation maximale possible, associée à une incertitude très grande. Les cartes d'élévation fournissent donc des données uniformément réparties. Nous nous assurons alors que le choix de la taille minimale des carreaux de splines garantit l'absence d'indéterminations sur le calcul des coefficients sur la base de B-splines (typiquement, on prend des carreaux minimaux de 4x4 données pour les surfaces bicubiques, ou de 3x3 données pour les surfaces bilinéaires, qui garantissent en pratique une bonne approximation des données).

La figure 6.15 montre une vue 3D d'une carte d'élévation obtenue à partir d'une image de profondeur de cet environnement, avec une résolution de 5cm×5cm de la grille ( $x, y$ ). La figure 6.16 en donne une représentation (en "vue de dessus") sous forme d'une image dont chaque pixel code en niveaux de gris la valeur de l'altitude au point de la grille correspondant. La figure 6.18 montre l'ensemble du modèle hiérarchique de surfaces B-splines obtenu pour ces données d'élévation, et les figures 6.19 et 6.20 en montrent deux vues partielles et rapprochées, correspondant à deux points de vue différents, et focalisées sur des zones de terrain caractéristiques (les rochers de la figure 6.14). La figure 6.17 montre sur la carte d'élévation les réseaux de carreaux au niveau supérieur.

Le modèle de terrain présenté ici comporte deux niveaux. Nous avons utilisé des B-splines bi-linéaires pour chacun des 2 niveaux de résolution. Nous avons pris pour définir le réseau uniforme de noeuds du niveau supérieur un point de la carte d'élévation sur 8, c'est à dire que la taille des carreaux est 40cm×40cm. Pour le niveau de résolution plus fine, nous avons choisi une taille de carreaux de 10cm×10cm.

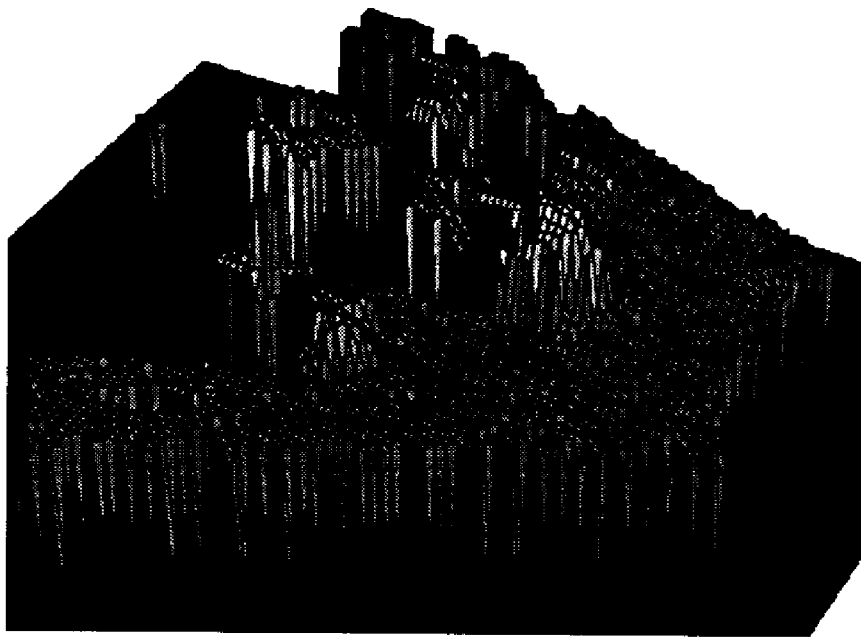


Figure 6.15: *Vue 3D de la carte d'élévation initiale*

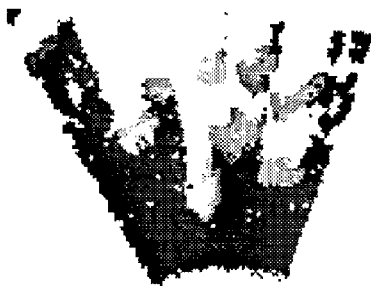


Figure 6.16: *Carte d'élévation initiale*

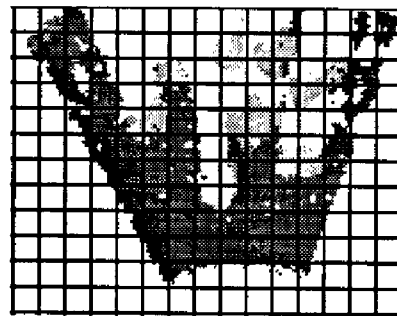


Figure 6.17: *Réseau de carreaux du niveau supérieur*

2 vecteurs d'environ 20 noeuds suivant chaque direction ont permis de définir les carreaux du niveau supérieur pour une zone de taille 8m x 8m environ; la surface B-splines modélisant le niveau supérieur est donc décrite par environ  $20 \times 20$  coefficients.

Sur la vue complète du modèle (figure 6.18), chaque point de la surface est calculé, pour l'affichage, selon le degré de résolution le plus fin disponible dans la représentation hiérarchique. On peut distinguer les carreaux du niveau supérieur qui n'ont pas été raffinés lors de la construction du modèle. En effet, nous avons volontairement omis d'afficher les points qui se trouvent sur les frontières de ces carreaux, ce qui fait apparaître des "coupures" artificielles entre les carreaux (la surface B-splines du niveau supérieur du modèle du terrain est bien évidemment continue !), mais constituait le seul moyen de matérialiser les limites des carreaux. On peut remarquer que, les B-splines utilisées ici étant bilinéaires, les frontières

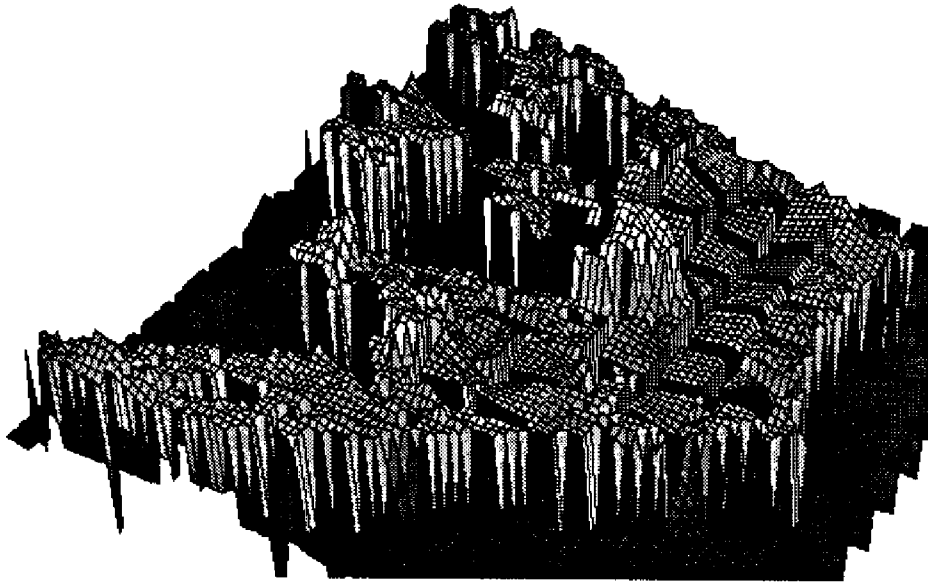


Figure 6.18: *Modèle B-splines hiérarchique d'un terrain accidenté*



Figure 6.19: *Modèle de terrain: vue partielle 1*

des carreaux sont des segments de droite. Nous n'avons pas fait apparaître par contre les limites des carreaux des niveaux inférieurs.

Si nous comparons le modèle finalement obtenu avec les images vidéo de la figure 6.14, nous constatons que le modèle de terrain conserve la structure topologique du terrain. Par

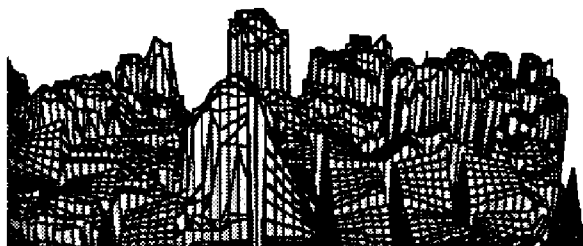


Figure 6.20: *Modèle de terrain: vue partielle 2*

exemple, les rochers de la figure 6.14 sont correctement représentés. Avec le niveau supérieur de la représentation (niveau 0), les plus grosses erreurs commises le sont au voisinage des fortes variations d'élévation, que des B-splines bilinéaires ne sauraient suffire à modéliser avec une bonne précision, avec la taille de carreaux choisie à ce niveau de description. La représentation finale décrit le terrain avec une précision bien meilleure.

Le modèle de terrain obtenu est très compact du point de vue de la place mémoire occupée, si on le compare par exemple aux MNT. Nous avons vu ci-dessus que le niveau supérieur est représenté par un nombre très petit de coefficients (20x20 dans l'exemple ci-dessus, pour un MNT de 160x160 pixels, environ); le modèle n'est raffiné que localement, près des zones très accidentées du terrain.

Les temps de calcul, sur une Sun Sparc II, sont de environ 1 seconde pour le calcul du niveau supérieur du modèle, et de l'ordre de quelques 10 ms pour le raffinage d'un carreau.

### 6.3 Modélisation incrémentale

Pour pouvoir évoluer dans un environnement partiellement connu ou inconnu *a priori*, un robot mobile doit pouvoir au cours de ses déplacements construire incrémentalement, en incorporant les données sensorielles issues de chaque nouvelle acquisition (images de profondeur délivrées par le télémètre laser), le(s) modèle(s) de l'environnement. Pour certaines tâches de modélisation, nous devons être capables de construire un modèle local du terrain, par fusions successives des différents modèles de terrain correspondant aux acquisitions sensorielles instantanées.

Rappelons qu'il faut connaître avec la meilleure précision possible la position du robot à chaque acquisition (ce qui suppose de recalculer les estimations trajectométriques de la position au cours du mouvement du robot). Ce problème a déjà fait l'objet du chapitre 5, où nous nous situons dans le cas où des amers de forme structurée se trouvent et peuvent être détectés dans l'environnement. Nous nous intéresserons également, dans le chapitre 7, au problème du recalage de la position du robot en l'absence de tels amers. Pour le moment, dans ce chapitre, nous supposons que les *six paramètres de position et d'orientation du robot* à chaque nouvelle acquisition sensorielle sont *connus*.

Les représentations manipulées doivent présenter certaines propriétés de localité, car il faut éviter de devoir re-calculer toute la représentation du terrain à chaque nouvelle acquisition. L'incorporation de nouvelles données instantanées, correspondant à des mesures sur une zone délimitée du terrain (champ de vue courant), ne doit nécessiter qu'une remise en cause locale du modèle de l'environnement.

Dans ce paragraphe, nous allons donc tout d'abord évoquer les propriétés de localité que les représentations B-splines possèdent. Nous présenterons ensuite les différentes méthodes que nous avons envisagées, puis mises au point, pour mettre à jour le modèle de terrain après une nouvelle acquisition de données. Nous présenterons enfin des résultats expérimentaux obtenus à partir de cartes d'élévation réelles.

### 6.3.1 Localité

Nous avons déjà dit dans le chapitre 3 que les propriétés de localité présentées par les surfaces B-splines présentent un intérêt pour la construction incrémentale d'un modèle de terrain utilisant ce type de primitives. Cette notion de localité doit être examinée sous deux aspects différents.

#### 1. Modification d'un coefficient de la surface sur la base de B-splines

Si on modifie un des coefficients  $a_{i,j}$  d'une surface B-splines décrite par l'équation 6.8, seuls les carreaux de la surface sur lesquels la B-spline associée  $B_{i,j}$  prend des valeurs non nulles seront affectés. Ces carreaux sont bien connus, ils se trouvent dans la zone de la surface délimitée par  $x_i$  et  $x_{i+m_x+1}$  suivant la direction  $x$ ,  $y_j$  et  $y_{j+m_y+1}$  suivant la direction  $y$  (cf paragraphe 6.1). Si les noeuds concernés sont de multiplicité égale à 1 (rappelons que c'est le cas de tous les noeuds intérieurs avec nos choix de multiplicités), cette zone recouvre  $(m_x + 1) \times (m_y + 1)$  carreaux.

#### 2. Modification d'un point de donnée

Lors du calcul du modèle de terrain, nous réalisons aux différents niveaux de résolution une approximation des données surfaciques aux moindres carrés (l'expression du critère minimisé est donnée par l'équation 6.10).

Nous devons ici tenir compte de notre processus d'ajustement pour savoir quelle est l'influence de la modification de l'altitude d'un point de données sur la représentation induite. L'ajustement de données à l'aide de surfaces B-splines (avec un réseau uniforme de carreaux) aux moindres carrés se ramène à la résolution d'un système linéaire. Le problème est d'inverser une matrice bande (cf. paragraphe 6.1.3).

Les recherches en Mathématiques et Mathématiques appliquées se sont attachées à étudier les propriétés des matrices inverses de telles matrices. Demko a montré [Demko 77] que les éléments  $a_{ij}$  de l'inverse d'une matrice bande décroissent uniformément exponentiellement lorsqu'on s'éloigne de la diagonale. Jaffart [Jaffard 90] a montré récemment que pour deux classes de matrice (à décroissance exponentielle ou polynomiale en s'éloignant de la diagonale), les coefficients de la matrice inverse ont la même propriété de décroissance.

Dans notre cas, ces résultats impliquent que la procédure d'approximation est **essentiellement locale** [De Boor 78], c'est à dire que l'influence de la modification de l'altitude d'un point de mesure décroît (*exponentiellement*) au fur et à mesure que l'on s'éloigne de ce point. Cette propriété de *forte localité* est illustrée par l'exemple suivant, dans le cas d'une approximation de points de donnée dans le plan par une courbe B-splines. Nous avons successivement réalisé des approximations aux moindres carrées par une courbe B-splines de degré 2 puis 3, sur deux ensembles de données  $E_1$  et  $E_2$ . Une partie des données de  $E_2$  est identique à celles de  $E_1$ , et une partie a changé. Les figures 6.21 et 6.22 montrent graphiquement les résultats de l'approximation (pour les données de  $E_1$  à gauche, et de  $E_2$  à droite), et les tableaux montrent les vecteurs de coefficients sur les bases de B-splines



obtenus. Les données sont réparties de manière uniforme, et les noeuds aussi (les noeuds sont placés toutes les 5 données; pour la clarté de l'affichage, toutes les données ne figurent pas sur les graphiques). Dans le cas de l'approximation de degré 3 par exemple, la perturbation des données de  $E_2$  par rapport aux données de  $E_1$  intervient sur des segments de courbe pour lesquels les B-splines  $B_{17,2}$  à  $B_{22,2}$  sont non nulles. Les tableaux illustrent la décroissance exponentielle de l'influence de cette perturbation sur les autres coefficients de B-splines, lorsqu'on s'éloigne de la perturbation.

B-spline	$E_1$ , degré m=2	$E_2$ , degré m=2	$E_1$ , degré m=3	$E_2$ , degré m=3
$B_{0,m}$	0.53	0.53	0.57	0.57
$B_{1,m}$	1.12	1.12	0.61	0.61
$B_{2,m}$	2.66	2.66	2.53	2.53
$B_{3,m}$	1.00	1.00	2.02	2.02
$B_{4,m}$	0.43	0.43	0.35	0.35
$B_{5,m}$	0.53	0.53	0.57	0.57
$B_{6,m}$	0.49	0.49	0.46	0.46
$B_{7,m}$	0.50	0.50	0.52	0.52
$B_{8,m}$	0.50	0.50	0.49	0.49
$B_{9,m}$	0.50	0.50	0.51	0.51
$B_{10,m}$	0.50	0.50	0.50	0.50
$B_{11,m}$	0.50	0.50	0.50	0.50
$B_{12,m}$	0.50	0.50	0.50	0.50
$B_{13,m}$	0.50	0.50	0.50	0.50
$B_{14,m}$	0.50	0.49	0.50	0.50
$B_{15,m}$	0.50	0.51	0.50	0.50
$B_{16,m}$	0.50	0.47	0.50	0.49
$B_{17,m}$	0.50	0.57	0.50	0.52
$B_{18,m}$	0.50	0.50	0.50	0.44
$B_{19,m}$	0.50	3.41	0.50	1.37
$B_{20,m}$	0.50	2.37	0.50	4.65
$B_{21,m}$	0.50	0.94	0.50	0.94
$B_{22,m}$			0.50	1.17

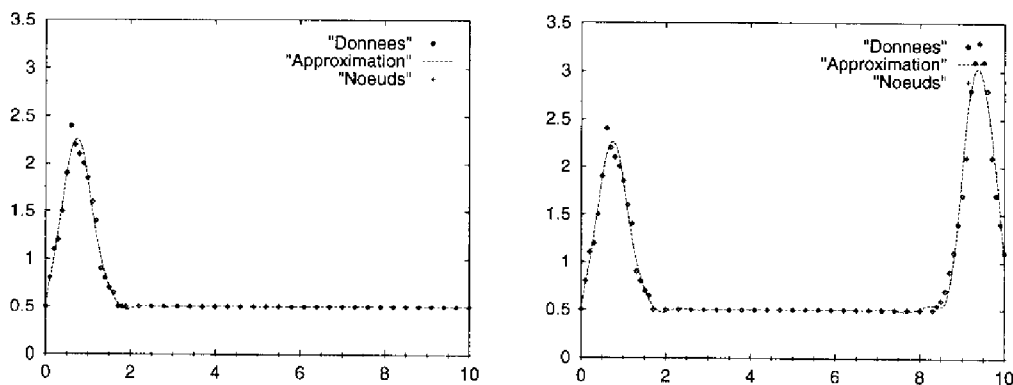


Figure 6.21: Approximation de degré 2 aux moindres carrés

Ce phénomène se généralise directement au cas de l'approximation aux moindres carrés à l'aide de surfaces B-splines de produit tensoriel. La même propriété de localité se manifeste suivant les deux directions  $x$  et  $y$ .

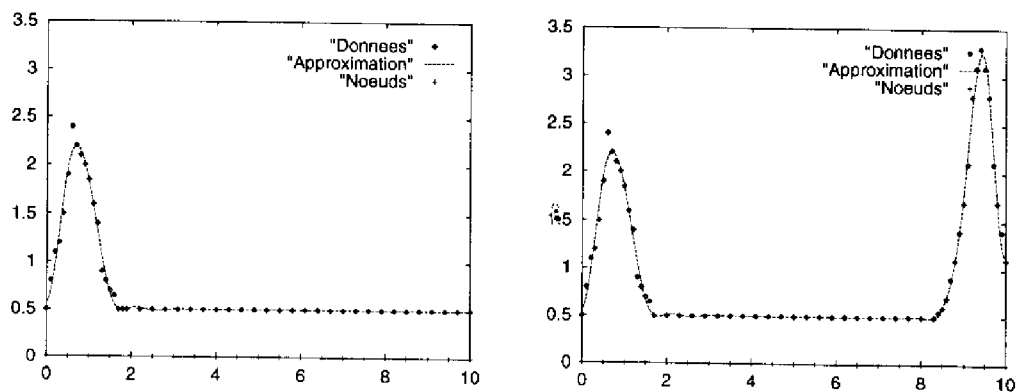


Figure 6.22: Approximation de degré 3 aux moindres carrés

Pour construire incrémentalement la représentation par B-splines du terrain, nous tirons parti de cette propriété pour ne recalculer que localement le modèle local du terrain après une nouvelle acquisition sensorielle. La figure 6.23 illustre l'approche générale adoptée. Sur cette figure, nous avons représenté une acquisition sensorielle instantanée, et la zone correspondante sur laquelle est calculée la carte d'élévation instantanée. Les carreaux qui apparaissent sur la figure sont ceux du modèle B-splines local et du modèle instantané. On suppose que la position du robot a été recalée, et le réseau de carreaux du modèle instantané est choisi de telle sorte qu'il soit superposable avec le réseau de carreaux du modèle local. La zone de fusion représente la zone dans laquelle le modèle B-splines local est remis en cause par l'acquisition courante. Nous prenons par rapport à la zone de mesure une "marge" tenant compte du phénomène très local de propagation évoqué ci-dessus. Cette marge est très peu étendue car l'influence des nouvelles données décroît exponentiellement. Le choix pratique que nous avons fait dans le cadre de nos applications est de prendre une marge de  $m$  intervalles si  $m$  est le degré de la surface; parmi les coefficients sur la base locale des B-splines, seuls ceux des B-splines qui apparaissent sur ces intervalles sont remis en cause. Ce choix est issu des nombreux essais que nous avons effectués.

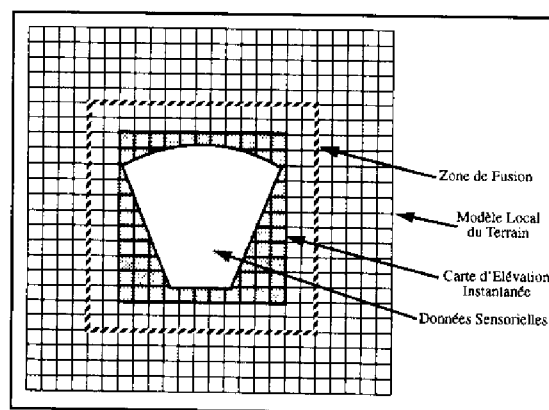


Figure 6.23: Construction incrémentale du modèle de terrain: schéma général

### 6.3.2 Deux approches au problème de modélisation incrémentale

Lorsqu'on vient de faire une acquisition de données, comment remettre à jour le *modèle local* hiérarchique de surfaces B-splines du terrain ? Deux approches possibles se dégagent :

1. Construire un modèle hiérarchique de surfaces B-splines à partir des informations sensorielles *instantanées*, et fusionner ensuite ce modèle instantané avec le modèle de surfaces B-splines local. Lors du processus de fusion, on manipule alors directement deux modèles de surfaces B-splines. Cette approche est présentée plus en détail dans le paragraphe 6.3.4.
2. Incorporer directement l'information contenue dans les données sensorielles de terrain dans le modèle local, sans passer par la construction d'un modèle de surfaces B-splines instantané. Cette approche est présentée dans le paragraphe 6.3.3 suivant.

### 6.3.3 Fusion "données instantanées à spline locale"

Dans cette approche, nous proposons de fusionner les nouvelles données sensorielles (décrites par une carte d'élévation) avec le modèle local de surfaces B-splines, sans passer par la construction d'un modèle de B-splines représentant les seules données instantanées. Cette méthode repose sur l'utilisation d'une structure intermédiaire de type "bitmap", familière chez les "visionnaires" ou les roboticiens, et en particulier dans notre équipe [Nashashibi 92, Lacroix 93]. La procédure proposée est la suivante :

1. On discrétise le modèle local du terrain dans la *Zone de Fusion* (cf. figure 6.23) : on reconstruit un MNT en calculant l'altitude du modèle de B-splines sur une grille régulière.
2. On construit la carte d'élévation (MNT) correspondant aux données sensorielles instantanées (sur une grille de même résolution) ;
3. On fusionne ce MNT avec la discrétisation du modèle B-splines local dans la *Zone de Fusion*, en utilisant la carte de bits donnant pour chaque point de la grille MNT l'information *connu* ou *inconnu*. Si aucune des deux valeurs d'élévation n'est connue, le point reste marqué *inconnu*. Si une seule des deux valeurs est *connue*, elle est retenue. Si les deux valeurs sont *connues*, de nombreuses règles peuvent être appliquées. On peut pondérer les deux valeurs par leur incertitude, ou utiliser quelques heuristiques (par exemple, garder la valeur correspondant à l'acquisition la plus proche, ou retenir la valeur la plus élevée – i.e. la plus pénalisante pour la navigation – si on veut donner une forte priorité à la sécurité du robot).
4. On calcule le modèle hiérarchique de surfaces B-splines sur la "carte d'élévation" (bitmap) provenant de cette fusion. Le réseau instantané de carreaux uniformes choisi

est *superposable* avec un sous ensemble du réseau uniforme choisi pour le modèle local. Ainsi, les deux bases (instantanée et locale) de B-splines comportent des *sous-bases communes*, ce qui permet de recalculer une partie du modèle local. Le niveau supérieur du modèle est d'abord recalculé. Comme nous l'avons expliqué ci-dessus, seuls les coefficients de B-splines (du modèle local) attachés à la *Zone de Fusion* sont recalculés. Les carreaux sur lesquels cette représentation est trop grossière sont raffinés selon la procédure exposée au paragraphe 6.2.

Cette approche présente l'inconvénient de nécessiter le passage par une structure intermédiaire de type "bitmap" ou "carte d'élévation" mais les temps de calcul restent compatibles avec les applications temps réel [Nashashibi 93a]. Par ailleurs, l'utilisation de B-splines uniformes et de degré faible nous permet d'optimiser les temps de calcul pour l'évaluation d'un point situé sur une surface B-splines (les expressions analytiques des B-splines sont bien connues).

### Résultats expérimentaux

Cette approche a été implémentée, et nous présentons les premiers résultats expérimentaux obtenus.

Nous montrons ici le résultat de la construction du modèle de terrain à partir de deux images laser prises dans l'environnement présenté figure 6.14. Le MNT correspondant à la première image est présenté figure 6.24, celui correspondant à la deuxième vue est donné figure 6.16. La figure 6.25 montre le modèle B-splines hiérarchique résultant de la fusion de ces deux vues. Nous présentons aussi, figure 6.26, et à titre de comparaison, le MNT issu de la fusion des MNT correspondant aux deux vues.

Les surfaces B-splines permettent grâce à la compacité de la représentation une grande économie en place mémoire pour le modèle de terrain, par rapport notamment aux MNT. Ceci est un point important puisque le but est d'embarquer l'ensemble des traitements sur le robot. Dans l'expérimentation EDEN, on ne conserve qu'un MNT "glissant" autour de la position courante du robot car cette représentation est coûteuse en place mémoire. L'effet de lissage par les B-splines est lié au degré, à la taille des carreaux, et à la précision demandée pour la représentation du terrain. Si celle-ci est plus élevée, l'effet de lissage est moins sensible.

#### 6.3.4 Fusion "spline locale à spline instantanée"

Dans cette approche, nous étudions le problème de la fusion de deux surfaces B-splines décrites par leurs représentations implicites (matrices de coefficients sur les bases de B-splines). L'idée de base est encore une fois d'utiliser une sous-base commune, dans la zone de terrain où il y a redondance entre la carte locale et la carte instantanée, et de fusionner les coefficients des B-splines de cette sous-base. Mais le problème de la fusion du modèle local et du modèle instantané est aussi celui de la fusion de deux arbres. La méthodologie proposée est la suivante:

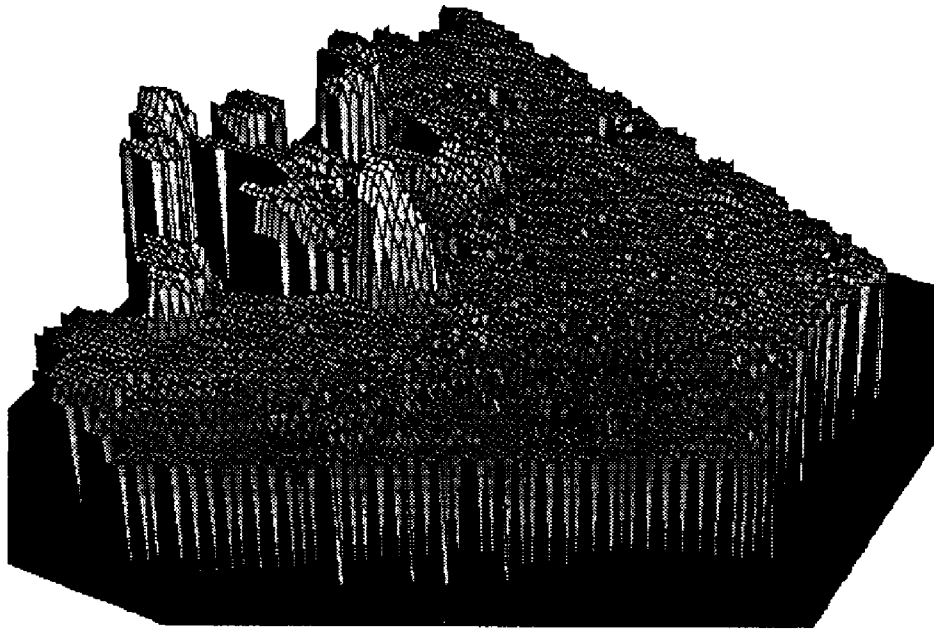


Figure 6.24: *MNT de la vue 1*

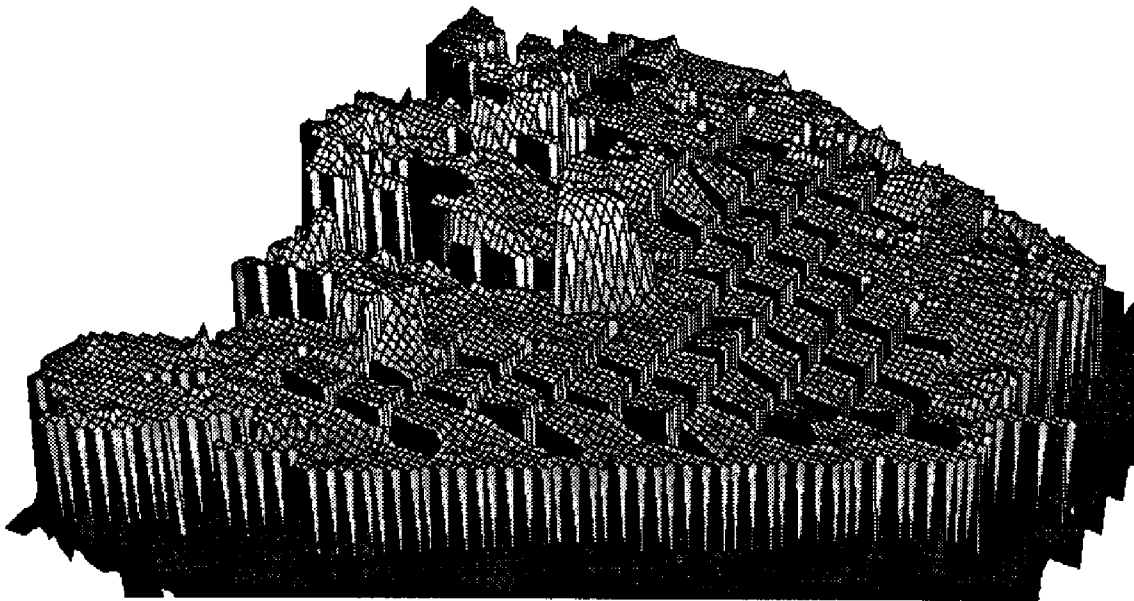


Figure 6.25: *Modèle hiérarchique de terrain fusionnant 2 vues*

1. on construit la représentation hiérarchique de surfaces B-splines dans la *Zone de Fusion*. Les données à modéliser sont constituées par le MNT instantané, plus la marge, nécessaire à la constitution de la zone de fusion, et dans laquelle les élévations proviennent de la représentation B-splines locale.

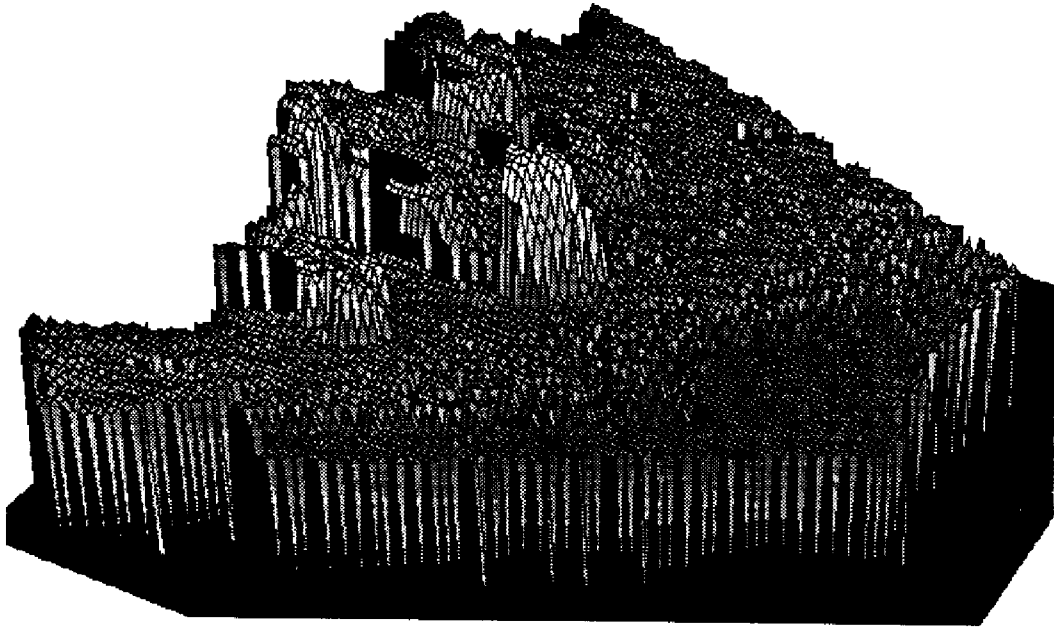


Figure 6.26: *MNT issu de la fusion des 2 mêmes vues*

2. on fusionne les niveaux supérieurs des deux modèles de terrain à base de B-splines (instantané et local), dans la *Zone de Fusion*. Seuls les coefficients des B-splines situées dans cette zone sont recalculés, grâce à une fusion des coefficients sur les deux sous-bases communes (locale et instantanée). Nous connaissons l'incertitude sur les coefficients des B-splines. La fusion tient compte des incertitudes des coefficients du modèle local et du modèle instantané. Plaçons-nous à la  $k^{\text{ième}}$  acquisition. Si  $\sigma_1^2$  et  $\sigma_2^2$  représentent respectivement les variances des coefficients de la B-spline locale et de la B-spline instantanée (qui sont superposables), le nouveau coefficient dans le modèle local est obtenu en pondérant les deux coefficients par leur variance:

$$(\mathbf{a}_{ij-\text{Local}})^k = \frac{\sigma_1^{-2} \cdot (\mathbf{a}_{ij-\text{Local}})^{k-1} + \sigma_2^{-2} \cdot (\mathbf{a}_{kl-\text{Instantane}})^k}{\sigma_1^{-2} + \sigma_2^{-2}} \quad (6.11)$$

où  $a_{ij-\text{Local}}$  est le coefficient de la B-spline de la base locale (indicée  $ij$ ) superposable avec la B-spline de la base instantanée indicée  $kl$ , dont le coefficient est noté  $a_{kl-\text{Instantane}}$ .

3. en général, on conserve entre deux descriptions pour un carreau de terrain la description la plus fine;
4. il faut cependant être capable d'abandonner une description fine sur un carreau si la description plus grossière devient suffisante au fur et à mesure que l'on emmagasine de l'information sur le terrain correspondant à ce carreau; pour cela, on étudie l'évolution

d'une mesure de la différence entre la représentation fine et la représentation plus grossière sur un carreau donné de la surface.

Cette approche n'a pas encore été implémentée. Dans l'équipe, nous poursuivrons le travail sur la représentation à l'aide de B-splines jusqu'à la comparaison des deux approches pour la construction incrémentale du modèle du terrain.

## 6.4 Conclusion et orientations futures

Nous avons présenté dans ce chapitre une méthode de modélisation hiérarchique d'un terrain accidenté à l'aide de B-splines, et souligné les principales caractéristiques de ce type de modèle. On obtient une représentation qui présente les avantages suivants:

- elle constitue une représentation compacte;
- elle possède des propriétés de localité intéressantes pour la construction incrémentale du modèle du terrain;
- elle garantit un ordre de continuité donné sans manipuler explicitement de conditions aux limites des carreaux composant la surface;
- on obtient des représentations analytiques, qui permettent par exemple de générer à tout moment un MNT de résolution quelconque. Nous allons surtout voir dans le chapitre suivant que le modèle analytique obtenu permet de raisonner sur l'information stockée.

L'utilisation du modèle du terrain présentée dans ce chapitre consiste à construire une représentation *géométrique* la plus précise possible du terrain. La décision de raffiner la représentation d'un carreau donné est prise selon des critères géométriques, en fonction de la précision de la représentation obtenue. Mais le nombre de niveaux du modèle, la décision de raffiner ou non, le choix du degré de la surface, etc. doivent surtout être *fonction de la tâche à effectuer*. C'est ce que nous allons discuter dans la suite de l'étude, en introduisant des aspects *sémantiques* dans la modélisation de terrain. Nous allons essentiellement nous intéresser, dans le chapitre suivant, à la *fonction localisation*. Nous allons notamment voir comment le modèle permet d'extraire des caractéristiques de terrain, qui seront utilisées en vue de recalculer la position du robot, au cours de son déplacement en environnement non structuré. Nous discuterons également en conclusion de l'utilisation du modèle pour la fonction *navigation* notamment (nous avons déjà fourni quelques idées dans ce sens au cours de ce chapitre 6). Nous allons ainsi introduire des aspects *sémantiques* dans la modélisation.

## Chapitre 7

# Localisation en Milieu Non-Structuré

Nous avons vu au chapitre 5 des méthodes de recalage lorsque des amers de forme géométrique simple et caractéristique sont présents dans l'environnement, et peuvent être utilisés pour réaliser le recalage de la position du robot [Devy 92b, Fillatreau 93a]. Nous allons dans ce chapitre nous placer dans le cas où le robot est dans un environnement complètement non structuré. Dans un tel contexte, le problème du recalage est un problème très difficile; de nombreuses approches peuvent être trouvées dans la littérature; parmi celles-ci, beaucoup s'appliquent difficilement à des données réelles. Un grand nombre de questions intéressent ici les chercheurs. Quels sont les invariants 3D sur lesquels on peut s'appuyer, pour effectuer des mises en correspondance ? Quel est le degré de stabilité de ces différentes caractéristiques ? Quelles méthodes de mise en correspondance de caractéristiques mettre en oeuvre ?

Nous allons dans un premier temps faire une rapide revue des méthodes de recalage 3D de deux cartes de terrain que l'on trouve dans la littérature, avant de présenter notre démarche pour le choix des invariants 3D utilisés, et la méthode de localisation mise en oeuvre. Rappelons que dans nos applications, la verticale est connue avec une bonne précision. Nous cherchons à recaler le cap  $\theta$  et la position  $(X, Y, Z)$  du robot.

Dans la méthode de recalage proposée, la représentation de terrain par B-splines joue un grand rôle. Dans ce chapitre, ce modèle est *dédié* à la fonction *localisation*.

### 7.1 Recalage en milieu non-structuré: état de l'art

On peut classifier les méthodes de recalage de cartes de terrain en deux grandes catégories, selon le type d'entités mises en correspondance:

- les méthodes que nous appellerons **iconiques** (*iconic matching* selon la terminologie rencontrée dans la littérature anglo-saxonne [Kweon 91b]). La principale caractéristique de ces méthodes est qu'elles s'appuient directement sur toutes les données de



terrain (points d'une carte d'élévation ou points de mesure bruts, obtenus par stéréovision ou télémétrie ...)

- les méthodes basées sur l'**extraction et la mise en correspondance** de "**caractéristiques de recalage**" (*feature matching*), qui peuvent être des *points particuliers*, des *courbes* ou des *surfaces*.

En l'absence d'une estimation de trajectométrie, les méthodes iconiques s'avèrent le plus souvent trop lourdes à mettre en oeuvre: en effet, la quantité d'information manipulée est grande, et la recherche de mises en correspondance doit s'effectuer dans des zones de recherche étendues. La mise en correspondance d'un petit nombre d'entités très particulières peut être plus efficace, et suffisante pour réaliser le recalage de la position du robot.

Si l'estimée de position obtenue par trajectométrie est relativement bonne, alors on peut envisager l'utilisation de ces deux types de méthodes. Des techniques de type iconique peuvent être utiles si les caractéristiques de recalage ne peuvent être extraites avec une précision suffisante, ou si le terrain, trop peu accidenté, ne révèle pas d'amer assez caractéristique pour être adapté à un quelconque processus de reconnaissance.

### Approche iconique

On peut considérer une carte d'élévation (cf. Chapitre 3) comme une image 2D, si on traite l'élévation de manière analogue à l'intensité dans une image. Une approche basée sur une technique de type corrélation ne peut cependant suffire à trouver une transformation 3D (3 paramètres de position et 3 d'orientation) entre deux cartes. L'application de telles techniques suppose connue l'orientation d'une des cartes par rapport à l'autre. C'est une hypothèse qu'il devient réaliste de faire notamment car de plus en plus de robots sont équipés de systèmes inertiels, dont les prix, longtemps prohibitifs, baissent, tandis que les performances d'estimation de l'orientation deviennent de plus en plus intéressantes. Le nombre de degrés de liberté se trouve ainsi souvent réduit aux trois paramètres de position.

C'est le cas pour les travaux de Gennery [Gennery 89], qui a traité le problème de la mise en correspondance entre des données de terrain acquises à l'aide d'un système stéréoscopique embarqué sur le robot (carte d'élévation locale) et des données du terrain obtenues par imagerie orbitale (carte d'élévation globale), les deux cartes étant de résolution différente. L'incertitude sur les points de la carte locale est connue, et celle sur les points de la carte globale est supposée constante. L'orientation étant supposée connue avec une précision suffisante, il détermine la translation qui minimise la somme des écarts sur l'élévation, pondérés par les incertitudes correspondantes. Le critère minimisé tient compte du fait que les résolutions des deux cartes ne sont pas les mêmes, et une stratégie "*coarse to fine*" est mise en oeuvre pour accélérer le processus de mise en correspondance. La méthode a été testée sur des données réelles.

Horn et Harris [Horn 91] ont proposé une approche analogue à l'estimation du flot optique [Horn 81]. S'il y a  $n$  pixels dans la carte de profondeur, un système de  $n$  équations à 6

inconnues permet d'estimer par la technique des moindres carrés les coordonnées des vecteurs vitesse angulaire et linéaire instantanés du capteur. La méthode a été testée sur des données réelles acquises à l'aide du capteur ERIM.

Les chercheurs de CMU [Hebert 89, Kweon 90, Thorpe 89] appariant deux cartes d'élévation par une méthode iconique. Ils proposent d'estimer la transformation  $T$  entre les deux cartes en minimisant le critère d'erreur:

$$\varepsilon = \sum_{(\mathbf{x}, \mathbf{y})} \|z_1(\mathbf{x}, \mathbf{y}) - \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{T})\|^2 \quad (7.1)$$

exprimant la somme des écarts quadratiques entre l'élévation en chacun des points dans la première carte et les élévations sur les mêmes emplacements dans la deuxième carte, transformées par  $T$ . Disposant d'une estimée initiale de  $T$  de précision raisonnable, ils ont utilisé un algorithme de descente de gradient de la forme suivante:

$$\mathbf{T}^{i+1} = \mathbf{T}^i + \kappa \frac{\partial \varepsilon}{\partial \mathbf{T}}(\mathbf{T}^i) \quad (7.2)$$

Citons encore les travaux de Szeliski [Szeliski 88], qui traite un problème similaire. Il minimise à l'aide d'une descente de gradient analogue à celle citée ci-dessus la distance entre deux ensembles de points. Une interpolation par spline surfacique est d'abord réalisée sur le premier ensemble de points, en imposant des contraintes de positionnement (*compatibilité*) et de *lissage* à la surface interpolante. Ensuite, il minimise un critère de distance entre la spline et le deuxième ensemble de points. Hébert [Hebert 92] adopte une approche analogue, et minimise la distance entre un ensemble de points de mesure et des surfaces actives décrivant le deuxième ensemble de points de mesure. Cet algorithme a été testé sur des données synthétiques.

Zhang, dans le cadre du projet VAP, a proposé une méthode de recalage itératif utilisant des histogrammes de distances, qui fournit une estimation de la transformation 3D (translation et rotation) entre deux ensembles de points 3D [Hotz 93]. Les points du 1er ensemble sont appariés avec leurs plus proches voisins dans le 2ème; un seuil de distance maximale est fixé, et une contrainte dite de *consistance d'orientation* est appliquée. Un histogramme des distances permet de sélectionner les appariements correspondant à une distance proche de la moyenne des distances. Les couples de points retenus permettent d'estimer la transformation 3D selon un critère des moindres carrés. On applique la transformation obtenue à tous les points de la 1ère vue, et on réitère le processus.

### Approches basées sur la mise en correspondance de caractéristiques

#### **Approche basée sur des points spéciaux**

Le problème de l'estimation du mouvement entre deux cartes de profondeur (ou deux cartes d'élévation) pose celui de la mise en correspondance de deux ensembles de points 3D. Il

peut être intéressant de sélectionner pour la recherche d'appariements des points présentant des caractéristiques particulières, et discriminantes. Celles-ci doivent correspondre à des invariants 3D, indépendants du point de vue. Un attribut particulièrement intéressant dans ce contexte est la courbure Gaussienne, qui, d'après la géométrie différentielle, est invariante par isométrie 3D, y compris le déplacement rigide (rotation et translation). Les points présentant une forte courbure peuvent donc être utilisés pour la mise en correspondance.

Si l'élévation  $z$  est considérée comme une fonction de la position 2D  $(x, y)$ , une surface peut être décrite sous la forme:

$$\underline{X} = [x, y, z(x, y)]^T \quad \text{avec: } (x, y) \in \mathbf{R}^2$$

Une surface de cette forme est appelée *Surface (Carreau) de Monge*; en tout point d'une telle surface, la courbure gaussienne  $K$  et la courbure moyenne  $H$  sont définies par [Brady 84]:

$$\begin{aligned} K &= \frac{z_{xx}z_{yy} - z_{xy}^2}{(1 + z_x^2 + z_y^2)^2} \\ H &= \frac{z_{xx} + z_{yy} + z_{xx}z_y^2 + z_{yy}z_x^2 - 2z_xz_yz_{xy}}{2(1 + z_x^2 + z_y^2)^{3/2}} \end{aligned} \quad (7.3)$$

l'indice double désignant la dérivée partielle d'ordre 2:

$$z_{xx} = \frac{\partial^2 z(x, y)}{\partial x^2}, z_{xy} = \frac{\partial^2 z(x, y)}{\partial x \partial y} \text{ et } z_{yy} = \frac{\partial^2 z(x, y)}{\partial y^2}$$

La courbure est le plus souvent calculée à partir d'une approximation locale de la surface sous forme de polynômes; les polynômes orthogonaux sont d'un usage classique ici [Bolle 84, Besl 86].

Dans le cadre de cette approche, Goldgof [Goldgof 88] a proposé une méthode pour appairer deux cartes de terrain, en utilisant un algorithme d'appariement développé par Chen et Huang [Chen 88]. Les chercheurs de CMU [Hebert 89], pour leur part, ont proposé une approche basée sur la mise en correspondance de points spéciaux sélectionnés en fonction de la magnitude des deux courbures principales de la surface en ce point.

### Approches basées sur des contours

Les méthodes rencontrées ici supposent une connaissance partielle de la transformation entre deux cartes de profondeur ou d'élévation (connaissance de l'orientation d'une carte par rapport à l'autre par exemple). Le problème de transformation 3D est ainsi ramené le plus souvent à un problème de transformation 2D (mise en correspondance de contours). Parmi les travaux rencontrés ici, beaucoup supposent le modèle local de l'environnement connu [Talluri 90, Kweon 91b, Stein 92]

Radack et Badler [Radack 89] ont cherché à appairer des *contours de distance*, qui sont des contours de points équidistants de points critiques, choisis en fonction de la magnitude de

leur courbure. Mais la représentation d'une surface par de tels contours dépend fortement de l'erreur sur la position de ces points critiques, et celle-ci est très sensible aux bruits. L'algorithme n'a été testé que sur des données synthétiques simples.

Rodriguez et Aggarwal [Rodriguez 89] ont extrait de leurs MNT des contours de falaises (*cliff maps*) en appliquant une convolution avec un filtre LoG continu de la forme:

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^6} (r^2 - 2\sigma^2) \exp(-r^2/2\sigma^2) \text{ où } r^2 = x^2 + y^2$$

Les contours obtenus sont refermés par détection des passages par zéro (*zero crossings*), puis codés par chaîne de Freeman; une procédure d'appariement 2D est menée entre les contours de la carte instantanée et ceux de la carte locale.

Dans une approche analogue, Kweon [Kweon 91b] apparie les lignes de niveau construites à partir de ses cartes d'élévation.

Kamgar-Parsi et Jones [Kamgar-Parsi 91], dans une application sous-marine, codent également les chaînes de contour en fonction des variations d'orientation le long du contour et les apparie à l'aide d'un critère associant la corrélation des codes de deux chaînes (calcul de l'*appariement erroné*), pour trouver leur recouvrement, et une mesure de leur proximité géométrique (calcul du *placement erroné*). Cette méthode a donné d'assez bons résultats sur des données réelles.

On peut encore citer ici les travaux de Stein et Medioni [Stein 92], qui ont utilisé les lignes d'horizon pour recalibrer la position du robot, approche également adoptée par Talluri et Aggarwal [Talluri 90]. Ce type de méthode ne donne pas de résultats très précis, mais peut être suffisant dans certains cas, si on n'a pas besoin d'un recalage précis des coordonnées de position du robot. Enfin, Chen et Medioni [Chen 91] utilisent les lignes de courbure pour la reconstruction d'objets.

### Approches basées sur des surfaces

Les méthodes citées ici reposent sur la mise en correspondance d'éléments de surface présentant des caractéristiques invariantes par isométrie 3D. Elles se décomposent très souvent en trois étapes:

- segmentation de la surface en primitives surfaciques invariantes; très souvent, cette segmentation est basée sur les signes des courbures moyenne et gaussienne [Besl 86];
- mise en correspondance de ces primitives surfaciques;
- estimation du mouvement à partir des correspondances trouvées.

Kehtarnavaz et Mohan [Kehtarnavaz 89] ont ainsi segmenté les surfaces et construit pour chaque surface un graphe d'adjacence des régions; l'appariement de deux surfaces se ramène à l'appariement de deux graphes de ce type (par recherche de cliques maximales). Liang et

Todhunter [Liang 90] ont adopté une approche similaire, avec des contraintes plus fortes lors de l'appariement des graphes. Notons ici aussi les travaux de N. Ayache dans le domaine de l'imagerie médicale.

Ces approches présentent l'inconvénient d'être très sensibles aux bruits et aux occultations; par ailleurs, elles mènent quelquefois à des temps de calcul très coûteux pour obtenir une précision raisonnable.

## 7.2 Quels invariants 3D ?

Les méthodes iconiques étant le plus souvent assez lourdes du point de vue mise en oeuvre et surtout du point de vue des temps de calcul induits, il nous a paru nécessaire tout d'abord de détecter des caractéristiques dans l'environnement. En effet, lorsque le robot se déplace sur un sol accidenté et dans un environnement dépourvu d'objets de forme géométrique simple, il peut cependant se trouver des zones de terrain suffisamment caractéristiques pour être utilisées en tant qu'amers de localisation (rocher de forme particulière par exemple, ...).

Le problème du recalage de la position 3D du robot dans un tel environnement pose donc la question importante du choix des invariants. Passons en revue les différents types d'invariants utilisés dans les approches présentées ci-dessus;

- Les points utilisés sont souvent des points de forte courbure. Celle-ci est souvent calculée à partir d'une reconstruction locale de la surface, dont le choix est très important, puisqu'il doit préserver les propriétés d'invariance de la courbure. De plus, la courbure est un attribut assez sensible aux bruits de mesure, et il faut faire un lissage adéquat des données au préalable (filtrage des données ou choix pertinent de la méthode de reconstruction de la surface). Nous nous sommes intéressés à l'utilisation de points de forte courbure; mais nous considérons que ce sont les régions de points de forte courbure dans leur ensemble, et non les points de courbure très grande ou maximale, qui peuvent offrir une stabilité suffisante pour le recalage de la position du robot. D'autres points spéciaux auxquels nous avons été amenés à penser naturellement sont les maxima d'élévation locaux (ou pics d'altitude) – qui correspondent généralement aussi à des points de forte courbure –; nous leur avons accordé une importance toute particulière dans la suite de l'étude, car leur stabilité est particulièrement bonne.
- Les approches basées sur les mises en correspondance de morceaux de surfaces reposent souvent sur une segmentation basée sur la courbure et les remarques précédentes s'appliquent encore ici. De même, les classes issues d'une telle segmentation ne sont pas toujours appropriées dans le cas de scènes d'extérieur. Enfin, ces méthodes sont notoirement sensibles aux bruits et aux occultations (car le type de surface ne suffit pas pour faire des mises en correspondance, et des critères non invariants comme la position du centre de masse et l'étendue de la primitive surfacique doivent être rajoutées), et requièrent souvent des temps de calcul importants.

- Les approches basées sur les mises en correspondance de contours peuvent présenter un intérêt dans notre cas; nous étudions actuellement la possibilité d'utiliser des lignes de niveaux extraites des cartes d'élévation (travaux actuels de F. Nashashibi dans notre équipe), ou des caractéristiques de type lignes de vallée ou de crête.

Les invariants auxquels nous nous sommes intéressés dans un premier temps sont donc les maxima locaux d'altitude, et les régions constituées de points de forte courbure.

## 7.3 Extraction de caractéristiques de terrain

Dans ce paragraphe, nous montrons comment le modèle hiérarchique de terrain présenté au chapitre 6 peut être utilisé pour extraire des caractéristiques de terrain [Fillatreau 93b] que nous utiliserons ensuite dans une procédure de recalage présentée au paragraphe suivant. Nous tirons parti de l'aspect multi-résolution du modèle de terrain pour focaliser la recherche de caractéristiques, sur les portions intéressantes du terrain.

Dans un souci d'obtenir des temps de calcul rapides, nous utilisons, lorsque cela est possible, les représentations matricielles globales (coefficients de B-splines), implicites et compactes, réduisant la quantité d'information traitée.

### 7.3.1 Pics d'élévation

Examinons tout d'abord les procédures d'extraction des maxima locaux d'élévation. Elle repose sur l'utilisation des représentations matricielles implicites des différents niveaux du modèle hiérarchique de terrain présenté au chapitre 6 [Fillatreau 93b].

#### 7.3.1.1 Extrémum d'une surface B-splines

Il est possible de déterminer, à partir de la représentation implicite B-splines d'une courbe  $y(x) = \sum_i a_i B_{i,m}(x)$ , les positions  $\{x_{ext}\}$  des maxima ou minima de  $y$ , ainsi que les positions des extrema  $\{(x_{ext}, y_{ext})\}$  d'une surface B-splines de produit tensoriel définie par  $z(x, y) = \sum_{i,j} a_{ij} B_{ij}(x, y)$  (cf. chapitre 6). Dans la suite, nous ne considérerons que le cas des maxima, car ce sont ces caractéristiques que nous souhaitons détecter.

Considérons tout d'abord le cas monodimensionnel, avant de généraliser au cas de surfaces de produit tensoriel. Pour une spline exprimée comme combinaison linéaire de B-splines  $P(x) = \sum_i a_i B_i(x)$ , des maxima locaux de  $P(x)$  apparaissent pour chaque maximum local dans la séquence de coefficients  $\{a_i\}$ . Intuitivement, on peut dire que si un extrémum  $a_i$  apparaît, la spline présente un extrémum "près" de la position où la spline correspondante  $B_i$  atteint son maximum. Comme nous connaissons les expressions et la forme des B-splines de base, nous pouvons déterminer les positions exactes de ces maxima.

Ici, nous donnons seulement les résultats obtenus dans les cas où le degré vaut 1 et 2, cas les plus utilisés en pratique (le détail des calculs est donné dans l'annexe D). Considérons une courbe spline définie par  $P(x) = \sum_i a_i B_i(x)$ , et soit  $a_j$  un maximum local dans la suite des coefficients des B-splines. En accord avec les notations définies au début du chapitre 6, la B-spline  $B_j$  est non nulle sur l'intervalle  $[x_j \ x_{j+m_x+1}]$ , et seules  $m_x + 1$  B-splines sont non nulles sur chaque intervalle  $[x_i \ x_{i+1}]$ .

#### Cas du degré 2

On montre que si un maximum  $a_j$  apparaît, la spline atteint un maximum pour:

$$x_{max} = \frac{a_{j-1}x_{j+2} - \frac{a_j}{2}(x_j + x_{j+1} + x_{j+2} + x_{j+3}) + a_{j+1}x_{j+1}}{a_{j-1} - 2a_j + a_{j+1}}$$

Dans le cas uniforme (où la distance entre deux noeuds successifs est une constante égale à  $\Delta x$ ), on obtient:

$$x_{max} = x_{j+2} - \frac{\Delta x}{\frac{a_j - a_{j-1}}{a_j - a_{j+1}} + 1}$$

qui confirme la propriété intuitive que  $x_{max} \in [(x_{j+1} + x_{j+2})/2 \ x_{j+2}]$  si  $a_{j+1} > a_{j-1}$ ,  $x_{max} = \frac{x_{j+1} + x_{j+2}}{2}$  si  $a_{j-1} = a_{j+1}$ , et  $x_{max} \in [x_{j+1} \ (x_{j+1} + x_{j+2})/2]$  si  $a_{j+1} < a_{j-1}$ .

#### Cas du degré 1

Dans ce cas, le résultat est beaucoup plus simple, comme on pouvait s'y attendre. Un maximum  $a_i$  dans la suite de coefficients de la spline correspond à un maximum situé au noeud  $x_{i+1}$ , où la B-spline  $B_i(x)$  atteint son maximum.

#### Cas d'une surface B-splines de produit tensoriel

Les résultats précédents se généralisent directement au cas des surfaces. Si nous voulons trouver les maxima locaux d'une surface, il nous suffit de rechercher les maxima locaux parmi les coefficients de B-splines  $\{a_{ij}\}$ . Dans le cas d'une surface bi-linéaire par exemple, un maximum local  $a_{kl}$  parmi les coefficients des B-splines correspond à un maximum local de la surface situé en  $(x_{k+1}, y_{l+1})$ .

### **7.3.1.2 Extraction de sommets**

Nous allons maintenant utiliser ces résultats pour détecter les maxima d'élévation. Nous utilisons alors le modèle hiérarchique de terrain pour extraire ces caractéristiques par une méthode multi-résolution ("coarse to fine").

L'idée générale est d'utiliser le niveau supérieur (description la plus grossière) du modèle pour trouver la position approximative de telles caractéristiques. On utilise ici la propriété que la représentation B-spline grossière réalise un filtrage qui lisse les données surfaciques,

éliminant les bruits. Ainsi, on sélectionne des régions “intéressantes” de la surface du terrain, dans lesquelles il est pertinent de rechercher la position exacte d’un maximum local d’altitude.

La première sélection est faite en recherchant des maxima locaux parmi les coefficients des B-splines de la description grossière du terrain, par application d’une technique simple de type gradient sur la matrice  $(a_{ij})$ . Comme la surface est exprimée sur une base uniforme de B-splines, nous mesurons ainsi ce que l’on peut appeler un “coefficient local d’étirement de la surface”, en utilisant un *double filtrage* (filtrage passe-bas à l’aide des B-splines, et application d’un gradient). Une caractéristique détectée à ce niveau de description a de très bonnes chances de présenter un intérêt car 1) les bruits de mesure ont été filtrés et 2) des maxima locaux (et non pas seulement absolus) ont été extraits.

Cependant, après cette analyse à un niveau grossier, nous ne connaissons pas avec une grande précision les positions de ces maxima. De plus, l’application d’une technique de type gradient sur le premier niveau du modèle hiérarchique peut conduire à la détection d’artefacts, dus à des oscillations de la spline. Ainsi, dans une deuxième phase, nous utilisons les niveaux inférieurs de la représentation hiérarchique pour déterminer avec une plus grande précision la position des pics, et pour éliminer les artefacts. Pour chaque maximum détecté au niveau grossier, on recherche les positions des maxima dans les carreaux raffinés voisins. La construction du modèle hiérarchique de surfaces B-splines est ici “orientée localisation”; nous ne construisons de représentation raffinée que dans ces carreaux-là. Le modèle ne comprend que 2 niveaux: le niveau supérieur, et le niveau raffiné pour lequel on adopte directement une très bonne résolution (des carreaux de 10cm x 10cm ou 20cm x 20 cm constituent un choix typique). Les positions raffinées correspondent aux maxima dans les coefficients de B-splines des représentations raffinées. Ainsi, l’analyse sur les niveaux raffinés du modèle permet:

- d’extraire les positions raffinées des maxima, dont la précision dépend de la nouvelle taille des carreaux de la spline;
- d’éliminer les artefacts par une analyse très locale du terrain;
- de calculer pour chaque pic confirmé un “coefficient de proéminence” correspondant à la valeur d’un gradient dans la matrice de coefficients de B-splines. Cet attribut peut être utile pour choisir les meilleures caractéristiques pour réaliser le recalage de la position du robot.

L’utilisation des représentations matricielles (compactes) et du caractère multi-résolution du modèle hiérarchique de terrain, conduit à une procédure d’extraction de caractéristiques très rapide. La première phase revient à appliquer un gradient sur une matrice (ou “image”) de coefficients de petite taille; les opérations de la deuxième phase sont encore très rapides, et ne sont menées que dans les zones d’intérêt du terrain.



### 7.3.1.3 Résultats expérimentaux

Les figures 7.1 à 7.4 montrent tout d'abord un exemple d'extraction de pics dans une carte d'élévation générée à partir d'images de profondeur réelles obtenues dans l'environnement présenté figure 6.14. La figure 7.1 montre la vue de dessus de la carte d'élévation initiale, et la figure 7.2 la plus grande partie du réseau de carreaux de la description la plus grossière de la surface à l'aide de B-splines. Sur la figure 7.3, on voit les positions approximatives des pics, déduites de l'analyse sur le niveau le plus grossier du modèle hiérarchique. Pour l'ensemble des résultats présentés ici, les modèles hiérarchiques de surfaces B-splines construits comportent deux niveaux bilinéaires. Les positions grossières des sommets correspondent donc à des intersections de la grille de carreaux du premier niveau. Sur la figure 7.4, après analyse sur les niveaux raffinés du modèle, les positions des pics ont été raffinées et quelques artefacts supprimés.

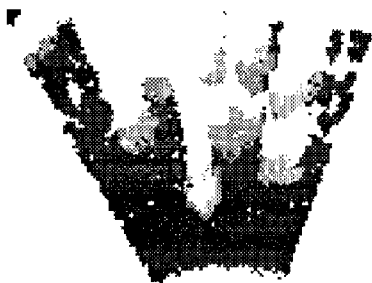


Figure 7.1: Carte d'élévation

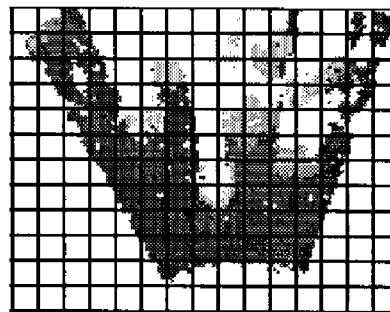


Figure 7.2: Réseau de carreaux grossier

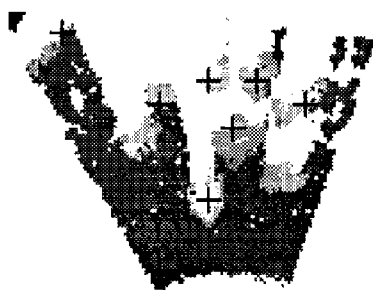


Figure 7.3: MNT 1: Positions approximatives des pics

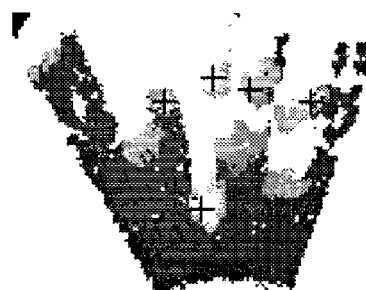
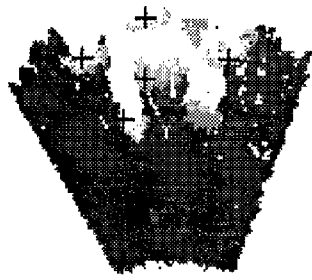
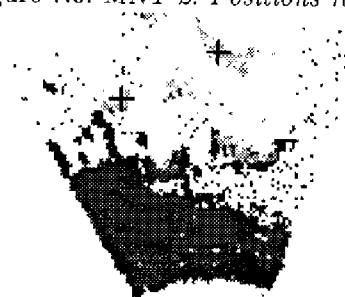
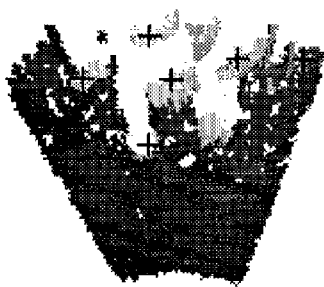
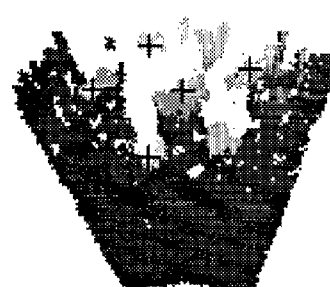


Figure 7.4: MNT 1: Positions raffinées des pics

Sur une Sparc 2, le temps de calcul de la représentation du niveau supérieur est d'environ 1s, et de l'ordre de 10ms pour chacune des représentations raffinées. L'application de techniques de type gradient sur des petites images de coefficients (environ 20x20 coefficients de B-splines pour le niveau supérieur du modèle) est très rapide.

Finalement, les figures 7.5 à 7.10 montrent plusieurs exemples d'extraction de maxima d'élévation sur une séquence de cartes d'élévation acquises sur le même environnement expérimental. Pour chaque image, nous montrons les positions de sommets déduites de l'analyse

du modèle au niveau grossier, puis les positions raffinées après analyse des niveaux plus fins. D'une image à l'autre nous pouvons constater qu'une partie des caractéristiques extraites correspondent aux mêmes maxima "physiques" d'élévation, ce qui rend légitime de les utiliser dans une procédure de recalage analogue au recalage par points présenté au chapitre 5. Enfin, les résultats montrent que cette méthode d'extraction de sommets est efficace même en présence de fortes occultations, avec la représentation que nous avons choisie pour les zones inconnues (cf. chapitre 6).

Figure 7.5: *MNT 2: Positions approximatives*Figure 7.6: *MNT 2: Positions raffinées*Figure 7.7: *MNT 3: Positions approximatives*Figure 7.8: *MNT 3: Positions raffinées*Figure 7.9: *MNT 4: Positions approximatives*Figure 7.10: *MNT 4: Positions raffinées*

### 7.3.2 Régions de forte courbure

Nous avons vu aux paragraphes 7.1 et 7.2 que la courbure est un invariant 3D qui a souvent été utilisé pour réaliser le recalage de 2 cartes de terrain (par mise en correspondance de points de forte courbure). En pratique, la valeur de cet attribut différentiel risque d'être très sensible au point de vue, et surtout aux bruits de mesure. Il faut donc accorder une grande importance à la reconstruction de la surface utilisée pour calculer ensuite la courbure; d'une part, elle doit préserver l'invariance de la courbure, et d'autre part elle doit réaliser un lissage adéquat des mesures [Guéziec 92].

Tirant profit de la différentiabilité<sup>1</sup> du modèle de terrain que nous avons construit, nous nous sommes intéressés au calcul de la courbure en un point quelconque de la surface. Nous pouvons utiliser les attributs de courbure pour:

- Mettre en évidence des zones de forte courbure, correspondant à des zones de terrain accidenté utilisables comme caractéristiques de recalage; on peut ainsi essayer de réaliser la mise en correspondance d'ensembles de points de forte courbure.
- Classifier la surface du terrain, et en réaliser une segmentation en fonction du signe des courbures moyenne et gaussienne (cf. [Parvin 91] pour une segmentation en portions de surface de type plane, cylindrique, hyperboloïde ou ellipsoïde, ou [Besl 86]).

#### 7.3.2.1 Calcul de la courbure en un point d'une surface B-splines

Le calcul de la courbure en un point quelconque d'une surface B-splines passe par le calcul intermédiaire de l'expression explicite analytique polynômiale de la surface  $z = f(x, y)$ , qui change d'un carreau de la surface à l'autre.

Pour cela, il faut tout d'abord repérer à quel carreau de la surface appartient le point pour lequel nous voulons calculer la courbure. On en déduit ensuite l'ensemble des B-splines qui sont non-nulles sur ce carreau-là, et dont nous connaissons les expressions analytiques. Il suffit alors d'utiliser les coefficients correspondants de la surface globale sur la base de B-splines pour retrouver l'expression polynômiale explicite de la surface sur le carreau considéré. Les courbures gaussienne et moyenne peuvent alors être déterminées à l'aide des formules 7.3.

Si nous voulons calculer la courbure en un point quelconque de la surface B-splines, le degré de la surface dans chaque direction doit être 2 au moins (et 3 si nous voulons que la courbure soit continue). Nous avons donc choisi de représenter le terrain à l'aide de B-splines bi-cubiques pour évaluer dans le cadre de notre étude l'intérêt de l'extraction de caractéristiques surfaciques définies à partir de la courbure. Nous donnons dans l'annexe E le détail du calcul de la courbure. La connaissance de ces expressions analytiques, que l'on peut "câbler hors ligne" permet d'accélérer grandement le calcul de la courbure (voir résultats expérimentaux).

<sup>1</sup>Rappel: l'ordre de différentiabilité dépend du degré de la surface et du choix de la base de B-splines (cf. chapitre 6)

Nous avons pu mettre en évidence l'importance des points suivants:

#### Le facteur résolution

Kishon [Kishon 90, Kishon 91] a montré l'influence du facteur "*résolution*" (en terme du nombre de noeuds plus ou moins grand pour approximer le même ensemble de données). Le choix de cette résolution a une très grande influence sur le niveau de lissage des données, et donc de la courbure. Une trop faible résolution filtre les caractéristiques, mais une résolution trop forte conduit à des courbures trop bruitées pour espérer les utiliser dans un processus d'appariement ou (surtout) de classification.

Nous avons effectué le calcul des courbures à partir d'un modèle hiérarchique comme celui présenté au chapitre 6, en calculant pour chaque point de la surface la courbure à l'aide de l'expression polynômiale de la surface au niveau de résolution le plus fin.

Nous avons obtenu des résultats cohérents dans les zones de la surface où la description grossière avait suffi à modéliser les données de terrain; ici, une classification de la surface selon les signes des courbures gaussienne et moyenne a bien permis de distinguer les zones presque planes des autres. Par contre, la courbure dans les zones raffinées devient vite très bruitée, et notamment inexploitable pour une classification "*fine*" [Besl 86, Parvin 91], si la taille des carreaux est trop petite (inférieure à  $5 \times 5$  pixels du MNT dans les essais que nous avons menés).

#### Energie interne et lissage de la courbure

Un autre problème important a été rapporté notamment par Guéziec et Ayache dans [Guéziec 92]. Leur objectif est de mettre en correspondance des courbes 3D suivant une méthode basée sur celle de Kishon *et al* [Kishon 91]; étant donnée une courbe déterminée par un ensemble de données bruitées, quelle approximation construire pour lisser correctement des attributs comme la courbure, et les rendre exploitables par une procédure de mise en correspondance ?

Les auteurs montrent que le choix d'un réseau uniforme n'est pas optimal, et que les régions de la courbe où la courbure varie fortement nécessitent une répartition plus serrée des noeuds. Par ailleurs, une approximation avec un simple critère de minimisation d'erreur de position conduit à des courbures assez instables en général (ceci est à mettre en balance avec l'influence de la résolution). Les auteurs obtiennent de très bons résultats pour la stabilité de la courbure avec les améliorations suivantes du critère à optimiser pour déterminer la meilleure approximation par une courbe B-splines cubique:

- améliorer le placement des noeuds en utilisant une approximation polygônale de ces points (qui comporte plus de sommets près des zones de forte courbure) pour guider le placement des noeuds. Rappelons que dans le cas des surfaces B-splines de produit tensoriel, on ne peut rajouter un noeud suivant une direction  $x$  ou  $y$  donnée sans affecter tous les carreaux suivant l'autre direction (cf. paragraphe 6.2); le besoin d'une plus grande densité de noeuds a été pris en compte grâce à notre approche hiérarchique.

- minimiser l'énergie de courbure par rajout d'un critère de minimisation d'énergie interne d'oscillation de la courbe;
- améliorer le calcul de la distance d'un point à une courbe;
- incorporer une mesure de la violation de la contrainte suivante: la courbe étant supposée appartenir à une surface dont les normales sont connues, la tangente à la courbe en un point doit être perpendiculaire à la normale à la surface en ce même point.

Ainsi, pour obtenir une plus grande stabilité de la courbure, des critères plus sévères qu'une simple minimisation d'erreurs aux données à modéliser devraient être mis en oeuvre. Les approches de type *snakes ou contours actifs* [Kass 87, Amini 88, Cohen 90, Williams 92], ou *surfaces actives* pour les représentations surfaciques [Cohen 91], entrent dans le cadre de cette approche. Ces méthodes exigent souvent des temps de calculs plus importants, et l'introduction de mécanismes plus complexes pour garantir une bonne convergence du contour ou de la surface. L'application de telles méthodes aurait permis d'atteindre une plus grande stabilité de la courbure que celle que nous avons obtenue; mais, en l'état actuel de la technologie, elle est difficilement envisageable dans un contexte robotique.

### 7.3.2.2 Résultats expérimentaux

Nous avons calculé les courbures sur l'ensemble des points des modèles surfaciques B-splines construits à partir de cartes d'élévation réelles. Le calcul analytique permet d'aboutir à des temps de calcul rapides: moins de 5s sur une Sparc Station 2 pour évaluer la courbure sur tous les points étiquetés "connu" d'une carte d'élévation.

Nous présentons figures 7.11 à 7.14 des résultats obtenus pour le calcul des courbures gaussienne et moyenne à l'aide du modèle hiérarchique de surfaces B-splines, dont les paramètres (résolution, ...) ont été choisis en fonction des remarques ci-dessus: un choix intermédiaire de carreaux bicubiques de taille 40cmx40cm (1 noeud est confondu avec 1 donnée d'élévation sur 5) permet de filtrer les bruits de mesure sans éliminer les caractéristiques du terrain.

La figure 7.11 présente une image laser brute du terrain d'expérimentation d'Eden. On distingue deux rochers (un à gauche et un à droite), et une "cassure" de pente à l'arrière plan. Les 3 cartes de la figure 7.12 représentent respectivement la carte d'élévation obtenue (de résolution 10x10cm), et les points de cette carte où les courbures gaussienne puis moyenne sont fortes (supérieures à un certain seuil); les courbures ont été calculées en tout point de la surface (suivant une grille  $(x, y)$  régulière) situé en zone connue, à l'aide du modèle hiérarchique de surfaces B-splines de la carte d'élévation. Les figures 7.13 et 7.14 donnent la classification obtenue à partir de la magnitude des courbures. Chaque pixel de l'image est codé suivant sa classe (region "inconnue" en blanc, "plane" en noir, "accidentée" en gris). La figure 7.13 donne les étiquettes pixel par pixel. La figure 7.14 donne la classification carreau par carreau: on affecte à un carreau de la surface l'étiquette majoritaire des pixels qu'il

contient. Ce critère n'est pas le plus adapté; nous allons également appliquer des opérateurs morphologiques pour obtenir une classification en régions à partir de la classification pixel à pixel.

Les attributs sont calculés au niveau de résolution le plus fin. L'énergie interne de la surface n'est pas optimisée. La classification binaire obtenue suffit cependant amplement pour être utilisée à des fins de recalage: les principales caractéristiques sont détectées lors de la classification binaire.

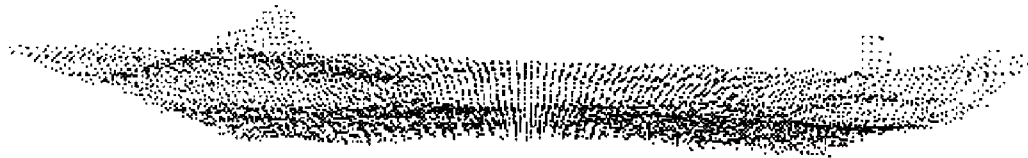


Figure 7.11: *Image laser brute*



Figure 7.12: *Carte d'élévation, courbures gaussiennes et courbures moyennes*



Figure 7.13: *Classification des points*

Figure 7.14: *Classification des carreaux*

### 7.3.2.3 Conclusion: utilisation actuelle de la courbure

Dans son état actuel, notre méthode de modélisation de terrain ne tient pas compte de critères de minimisation d'énergie interne dans les processus d'approximation surfacique; seule la résolution des noeuds, et les propriétés de lissage des splines bi-cubiques, permettent de filtrer les bruits et de lisser la courbure; elles ne suffisent pas pour garantir la meilleure stabilité. Cependant, on obtient une stabilité suffisante pour extraire les régions accidentées (à forte courbure), utilisables pour le recalage (cf. paragraphe 7.4.2), et pour réaliser une classification zones planes / zones accidentées. La stabilité n'est pas suffisante pour classifier suivant les classes proposées par Besl et Jain, ou Parvin et Medioni [Besl 86, Parvin 91]. Toutefois, le type de segmentation qu'ils proposent est bien adapté pour la segmentation de la surface d'objets manufacturés. Pour les applications qui nous intéressent, elle serait plus appropriée à la segmentation d'images orbiteur (dans lesquelles des cols, vallées, ... apparaissent) qu'à la segmentation d'images acquises par le robot, les régions obtenues alors pouvant être utilisées pour l'aide à la navigation par exemple (comme l'étaient les cols et les vallées dans [Morlans 92]).

Pour espérer cependant faire une classification plus fine du terrain, il faudrait sans doute utiliser des surfaces actives permettant un meilleur lissage de la courbure par minimisation de critères comme l'énergie interne de flexion. Celles-ci se révèlent particulièrement bien adaptées dans les cas où on dispose de données denses que l'on veut recalquer avec une grande précision (imagerie médicale par exemple). Mais on peut aussi se demander si dans le cadre de nos applications, où les données sont bruitées et les occultations nombreuses, les surfaces actives sont bien appropriées. Se prêtent-elles bien aux types de traitements et raisonnements qui nous intéressent ? Enfin, nous visons des applications temps réel, ce qui nous interdit pour l'instant ces méthodes trop coûteuses en temps de calcul.

## 7.4 Recalage: approche adoptée

Nous allons maintenant présenter la méthode que nous avons développée pour le recalage, qui est basé sur l'extraction de caractéristiques ponctuelles: points de forte courbure et pics d'élévation.

### 7.4.1 Recalage par mise en correspondance de points

Une première possibilité pour réaliser le recalage du robot est d'utiliser directement les caractéristiques ponctuelles extraites dans l'environnement, et représentées par des points 3D  $(x, y, z)$ . Dans une procédure en tout point analogue à celle décrite au chapitre 5, nous pouvons construire incrémentalement un modèle de localisation contenant par exemple les pics détectés au cours du déplacement du robot, et remis à jour à chaque acquisition, après mise en correspondance des pics contenus dans le modèle instantané de localisation, et ceux contenus dans le modèle local.

La mise en correspondance de points 3D peut permettre théoriquement d'estimer les 6 paramètres de position et d'orientation du robot. Mais nous supposons la verticale connue par trajectométrie. Nous recalons donc les paramètres de position 2D et de cap  $(x, y, \theta)$ , de la même façon qu'avec les lignes de discontinuité verticale (cf. chapitre 5). La troisième coordonnée de position  $z$  peut être recalée par différence entre les altitudes des points mis en correspondance (pondérées par les incertitudes sur l'altitude de chaque sommet).

#### 7.4.2 Recalage par coopération avec un appariement iconique

Cependant, nous devons faire les observations suivantes:

- Les caractéristiques ponctuelles que nous détectons sont des invariants 3D. Cependant, en pratique, comme pour la courbure, le calcul de ces attributs est sensible aux erreurs de mesure, aux occultations, au point de vue, ... Nous avons souligné ce point pour la courbure. Pour un *même* pic perçu plusieurs fois, le résultat de la chaîne de traitements (*perception / modélisation / extraction de pics*) conduit à des résultats très proches, mais qui varient en fonction du point de vue, à cause notamment des occultations, et du fait que la précision de la position détectée est limitée par la taille des carreaux de la surface B-splines au niveau de résolution le plus fin (et l'effet de lissage associé).
- une simple mise en correspondance de caractéristiques ponctuelles peut conduire à des appariements ambigus, comme nous l'avons vu au chapitre 5, surtout si de telles caractéristiques sont présentes en grand nombre [Fillatreau 93a] (ce qui est par exemple le cas dans les résultats expérimentaux présentés figures 7.3 à 7.10). Une telle méthode est très rapide, et se montre particulièrement adaptée lors de déplacements rapides en présence d'amers peu nombreux; sans doute est-elle moins adaptée lorsque de tels amers sont très nombreux, ce qui se produit par exemple lorsque le robot se déplace sur un terrain très accidenté; le mode de déplacement doit alors être beaucoup plus prudent, nécessite souvent l'étude des contacts avec le sol, une modélisation très fine de celui-ci, et un recalage précis (alors que les conditions sont défavorables à une trajectométrie précise);
- les cartes d'élévation [Nashashibi 92, Nashashibi 93a] que nous construisons par ailleurs (ou une carte d'élévation issue de la discrétisation du modèle hiérarchique de surfaces B-splines) contiennent une information dense sur les zones caractéristiques du terrain, dont il serait dommage de se priver, car la richesse de cette information peut nous permettre de réaliser un recalage bien plus précis de la position du robot !

Nous proposons une approche faisant coopérer l'extraction de caractéristiques avec des méthodes de type iconique, appliquées seulement dans les zones d'intérêt désignées par l'extraction des caractéristiques (ici, les zones situées autour des pics ou les régions de points à forte courbure par exemple). En appliquant une telle technique iconique localement seulement, nous nous affranchissons de la lourdeur de ces techniques lorsqu'elles sont utilisées



sur l'ensemble des mesures surfaciques, tout en tirant parti de l'information dense fournie par les MNT sur les zones caractéristiques.

Dans un premier temps, on extrait les amers présents dans la scène instantanée (pics d'élévation ou points de forte courbure), sur la base de la modélisation par B-splines. Ces caractéristiques (ou *points de contrôle*) sont projetées dans la carte d'élévation locale grâce à l'estimation de position fournie par trajectométrie. Nous appliquons alors une fonction de corrélation entre la carte d'élévation instantanée  $z = f(x, y)$  exprimée dans le repère robot redressé, et  $Z = f(X, Y)$  la carte locale. La corrélation n'est appliquée que dans un voisinage étroit des points de contrôle (on choisit typiquement des fenêtres  $\Omega$  de taille (3x3) ou (5x5) pixels de taille 10cm, autour des caractéristiques ponctuelles extraites). Elle revient à apparier les fenêtres  $\Omega$  dans le MNT instantané avec une fenêtre de même taille dans le MNT local. Nous utilisons pour la fonction de corrélation la forme classique:

$$C(i, j) = \frac{\sum_{i_0=0}^{I_\Omega} \sum_{j_0=0}^{J_\Omega} f(\hat{i} + i_0, \hat{j} + j_0) \cdot F(i + \hat{i} + i_0, j + \hat{j} + j_0)}{\sqrt{\sum_{i_0=0}^{I_\Omega} \sum_{j_0=0}^{J_\Omega} f(\hat{i} + i_0, \hat{j} + j_0)^2 \cdot \sum_{i_0=0}^{I_\Omega} \sum_{j_0=0}^{J_\Omega} F(i + \hat{i} + i_0, j + \hat{j} + j_0)^2}} \quad (7.4)$$

Pour un point de contrôle donné  $(\hat{i}, \hat{j})$ , on recherche dans un voisinage de taille  $\Delta i_0 \times \Delta j_0$  autour de la prédiction trajectométrique, la position  $(i, j)$  telle que le coefficient de corrélation  $C(i, j)$  soit maximum. La taille  $\Delta i_0 \times \Delta j_0$  de cette zone de recherche dépend de l'incertitude sur l'estimée trajectométrique de la position (elle est typiquement comprise entre 7x7 et 11x11 pixels). L'application d'une telle méthode suppose que *l'attitude du robot est connue au préalable avec une bonne précision*. Nous connaissons la verticale avec une bonne précision, et le cap peut être recalé par la mise en correspondance des caractéristiques ponctuelles détectées selon la procédure de recalage décrite au chapitre 5).

Afin de tenir compte du décalage en  $z$  des deux cartes d'élévation à mettre en correspondance, nous avons injecté dans le critère de corrélation ci-dessus la *différence d'élévation moyenne* autour des caractéristiques de terrain, qui s'exprime dans une fenêtre  $\Omega$  de taille  $I_\Omega \times J_\Omega$  par:

$$\Delta Z_{0est} = \frac{1}{N_P} \sum_{i_0=0}^{I_\Omega} \sum_{j_0=0}^{J_\Omega} f(\hat{i} + i_0, \hat{j} + j_0) - F(\hat{i} + i + i_0, \hat{j} + j + j_0) \quad (7.5)$$

Dans la formule 7.4, nous remplaçons donc l'expression  $f(\hat{i} + i_0, \hat{j} + j_0)$  par  $f(\hat{i} + i_0, \hat{j} + j_0) - \Delta Z_{0est}$ .

Si nous avons extrait un ensemble de points de contrôle  $\{P_k\}$ , le résultat de la corrélation pour chaque point de contrôle donne  $(i_k, j_k)$  qui fournit une estimation partielle du mouvement  $(T_x^k, T_y^k)$ , avec un coefficient de corrélation  $C_k$ . On obtient également une estimation partielle de  $T_z^k$  par la différence moyenne d'élévation dans le voisinage du point de contrôle,

entre les 2 cartes. Les incertitudes sur les valeurs d'élévation dans chacune des cartes fournit directement l'incertitude sur la moyenne des différences.

Plus  $C_k$  est proche de la valeur 1, meilleur est l'appariement iconique réalisé à l'aide du point de contrôle  $P_k$ . Ceci se produit si la zone de terrain est bien caractéristique (pic proéminent par exemple). Pour tenir compte dans l'estimation du mouvement de la qualité des appariements partiels réalisés, l'estimation finale est obtenue en pondérant toutes les estimations partielles avec les coefficients de corrélation partiels:

$$\mathbf{T} = \sum_k \frac{C_k}{\sum_k C_k} \mathbf{T}_k \quad (7.6)$$

Les appariements correspondant à des coefficients de corrélation trop faibles ne sont pas retenus. Par ailleurs, si pour un point de contrôle  $P_k$  donné, le maximum de la fonction de corrélation est obtenu pour une position en bordure de la zone de recherche  $(\Delta i_0, \Delta j_0)$ , on considère que la procédure d'appariement a échoué. Enfin, on prend en compte implicitement les positions relatives des différents amers. En effet, on étudie la dispersion de l'ensemble des transformations  $T_k$ , et on écarte les  $T_k$  éloignés de la moyenne si la dispersion est importante.

L'algorithme de recalage, que nous avons mis au point et développé avec F. Nashashibi [Nashashibi 92, Fillatreau 93c], est finalement le suivant:

1. Extraire les points de contrôle de la vue instantanée,
2. Utiliser  $T_0$  (odométrie ou centrale) pour calculer les prédictions trajectométriques des points de contrôle dans la carte locale,
3. Pour un voisinage  $(\Delta i_0, \Delta j_0)$  autour de chaque point de contrôle  $(\hat{i}_k, \hat{j}_k)$  faire:
  - choisir une fenêtre  $\Omega$  de taille  $N \times N$  pour le calcul de  $C_k(i, j)$ ,
  - calculer la différence d'élévation moyenne:  $\Delta Z_{0est}$ ,
  - calculer  $C_k(i, j)$  pour tout  $(i, j) \in \Omega$  (en injectant  $\Delta Z_{0est}$ ),
  - trouver  $C_k(i, j)_{max}$  qui maximise la fonction de corrélation,
  - calculer la transformation  $T_k$  correspondante,
4. Calculer la transformation  $T$  en pondérant les transformations  $T_k$  par les coefficients de corrélation  $C_k$  correspondants.

Les différents types de caractéristiques ponctuelles que nous savons extraire à l'aide de la modélisation par B-splines peuvent être utilisés comme points de contrôle de l'algorithme ci-dessus. Si on extrait des pics, les fenêtres  $\Omega$  sont centrées sur ces points. Si on veut utiliser une région de points de forte courbure, on peut utiliser par exemple une fenêtre rectangulaire englobant la région considérée, ou une fenêtre de taille donnée centrée sur le centre de gravité des points 3D composant cette région.

## 7.5 Resultats expérimentaux

La figure 7.15 montre un exemple de recalage de deux cartes d'élévation, de résolution 5cmx5cm, de l'environnement expérimental présenté figure 6.14. On montre successivement les pics d'élévation détectés à l'aide de la représentation par B-splines instantanée, puis les points corrélés locaux correspondants, avec les fenêtres de corrélation correspondantes, qui sont utilisées pour l'appariement iconique. Les meilleurs coefficients de corrélation obtenus correspondent aux rochers les plus "caractéristiques" (par exemple "les plus pointus"), qui donnent les estimations les plus fiables du déplacement. Enfin, le MNT local obtenu après fusion est présenté (cf. [Nashashibi 92] pour plus de détails sur cette procédure de fusion, qui est effectuée "pixel à pixel" et tient compte de l'incertitude sur les 2 valeurs d'élévation fusionnées). La figure 7.16 montre une vue 3D de ce MNT.

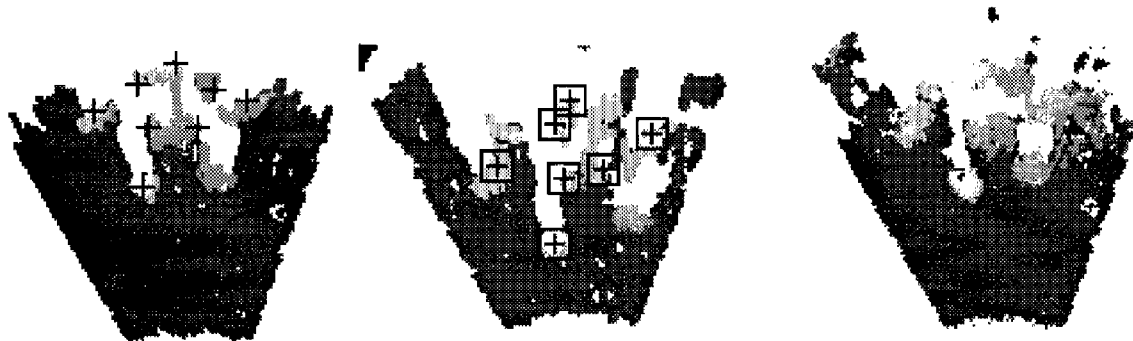


Figure 7.15: *Pics détectés, points de corrélation locaux correspondants et MNT après fusion*

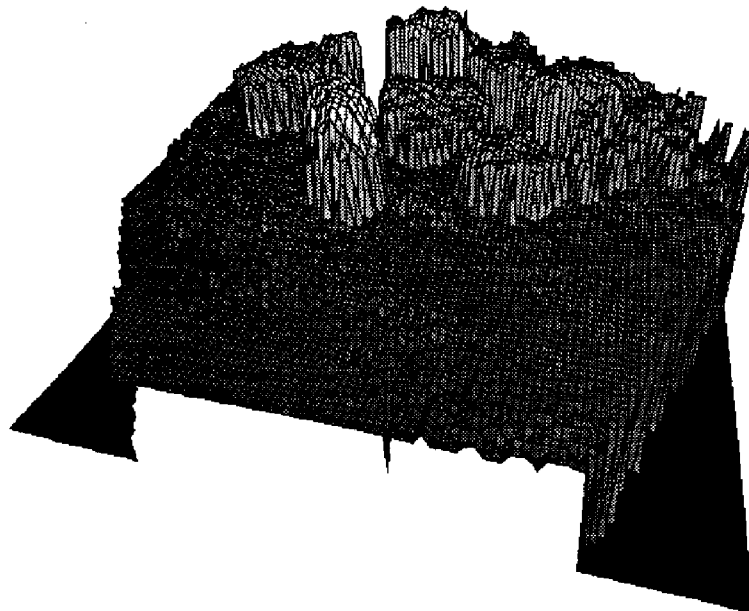


Figure 7.16: *MNT local après fusion: vue 3D*

L'ensemble de la procédure de corrélation prend quelques secondes de temps de calcul. La précision du recalage effectué est de 1 à 2 pixels, soit dans notre cas inférieure à 10cm.

Nous présentons maintenant des résultats obtenus dans le cadre de l'expérimentation EDEN. Le robot ADAM se déplace dans un environnement naturel présenté figure 7.17. Les résultats du recalage sont utilisés pour construire incrémentalement le MNT local du terrain (de résolution 10cm x 10cm).

Les figures 7.18 à 7.20 montrent la construction incrémentale du MNT local à partir de 5 images de profondeur.



Figure 7.17: *Environnement naturel d'ADAM*

Le tableau suivant donne les résultats obtenus pour les recalages de la position du robot durant son déplacement. La 1ère colonne donne l'étape en cours, la 2ème colonne donne les estimations trajectométriques des 6 paramètres de position ( $x, y, z$ ) et d'orientation (roulis  $\phi$ , tangage  $\psi$ , cap  $\theta$ ). La 3ème colonne contient les corrections effectuées sur les paramètres de position  $x, y$  et  $z$ . Enfin, la 4ème colonne donne le nombre de pics ayant conduit à des appariements intervenant dans le calcul du recalage de position (appariements iconiques couronnés de succès).

Étape	Estimées de trajectométrie (m., degrés)		recalages (m.)	Nombre d'appariements
	$(\hat{x}, \hat{y}, \hat{z})$	$(\phi, \psi, \theta)$	$(\delta x, \delta y, \delta z)$	
Fusion 1	8.96	5.28	0.04	6
	15.55	1.16	0.02	
	0.08	-6.89	-0.09	
Fusion 2	10.15	6.75	-0.23	4
	15.83	1.06	-0.42	
	0.09	-17.09	-0.06	
Fusion 3	12.57	5.42	0.03	1
	14.59	-6.97	0.78	
	0.57	-47.57	-0.02	
Fusion 4	15.27	-0.82	-0.03	3
	11.62	4.71	0.75	
	-0.20	-92.26	-0.04	

Les deux premiers recalages effectués sont peu importants en  $x$  et  $y$ , car le robot se déplace à ce moment-là sur un terrain peu accidenté. Par contre, pour Fusion 3 et Fusion 4, le robot est entrain de franchir le "col" entre les 2 amas rocheux, et se déplace sur un sol très accidenté d'une part, et glissant (à cause des conditions météorologiques ayant précédé cette expérimentation) d'autre part. Sur ce type de terrain, les performances de la trajectométrie sont beaucoup moins bonnes, et les recalages obtenus sont plus importants.

La figure 7.18 présente le recalage et la fusion de MNT réalisés à l'étape 2. Elle montre successivement les pics extraits à l'aide du modèle de B-splines instantané, les points corrélés correspondants (et les fenêtres de corrélation) dans le MNT local, et le nouveau MNT local après fusion. Pour apprécier la qualité du recalage, remarquons notamment que la zone de quelques "pixels", située en zone inconnue et derrière le groupe de rochers sur la gauche du champ de vue du robot (il s'agit de la représentation d'un amer vertical de type colonne ou tronc d'arbre), n'a pas été dupliquée; la précision du recalage est de l'ordre de 10cm ou mieux. La figure 7.19 montre le MNT local après 4 fusions, et les positions successives d'ADAM pour les différentes acquisitions. La figure 7.20 donne une vue 3D du même MNT; la position du robot est la position où a été effectuée la dernière acquisition, après recalage. Pour donner une idée de la taille de ce MNT, notons que le robot a été représenté à l'échelle, et rappelons que ses dimensions sont: 3m x 2m x 2m.

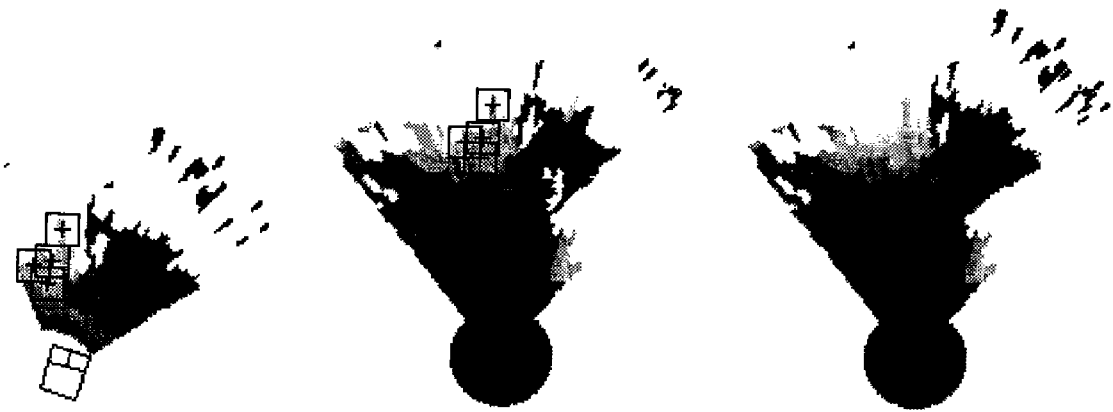


Figure 7.18: *Fusion 2: Pics détectés, points corrélés correspondants et MNT après fusion*

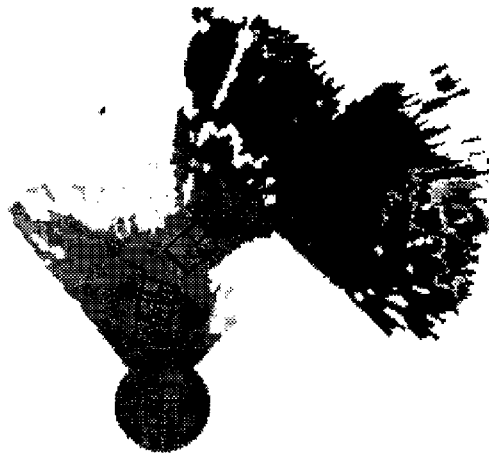


Figure 7.19: *MNT local après 4 fusions*



Figure 7.20: *MNT local après 4 fusions*

Nous montrons figures 7.21 à 7.27 les résultats correspondant à une nouvelle expérimentation (cette fois, 9 images de profondeur ont été acquises).

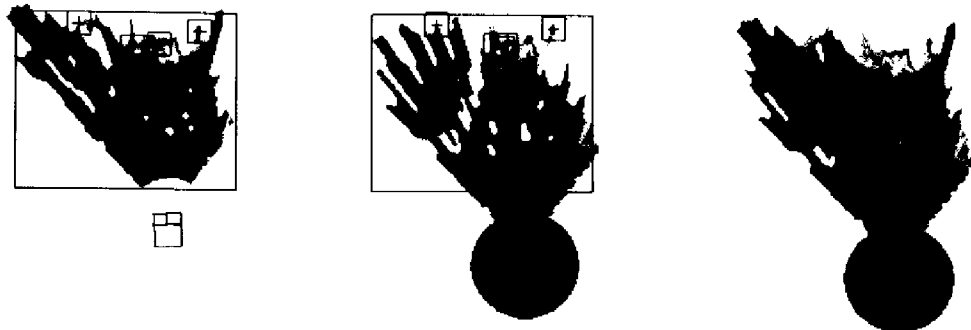


Figure 7.21: *Recalage-Fusion 1*

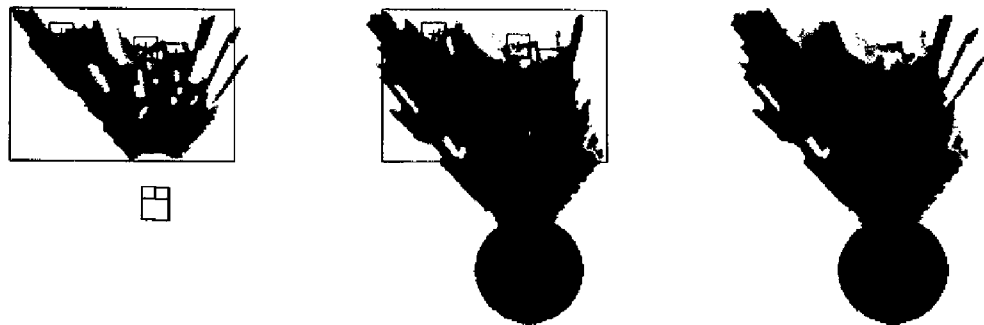


Figure 7.22: *Recalage-Fusion 2*

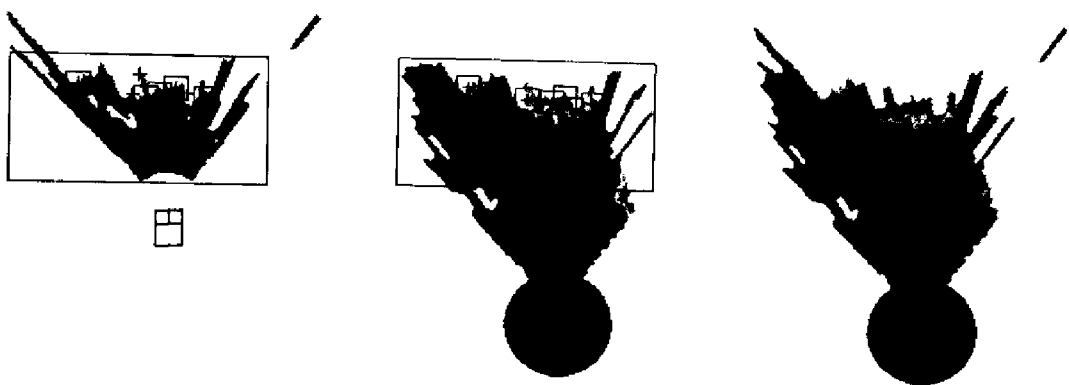
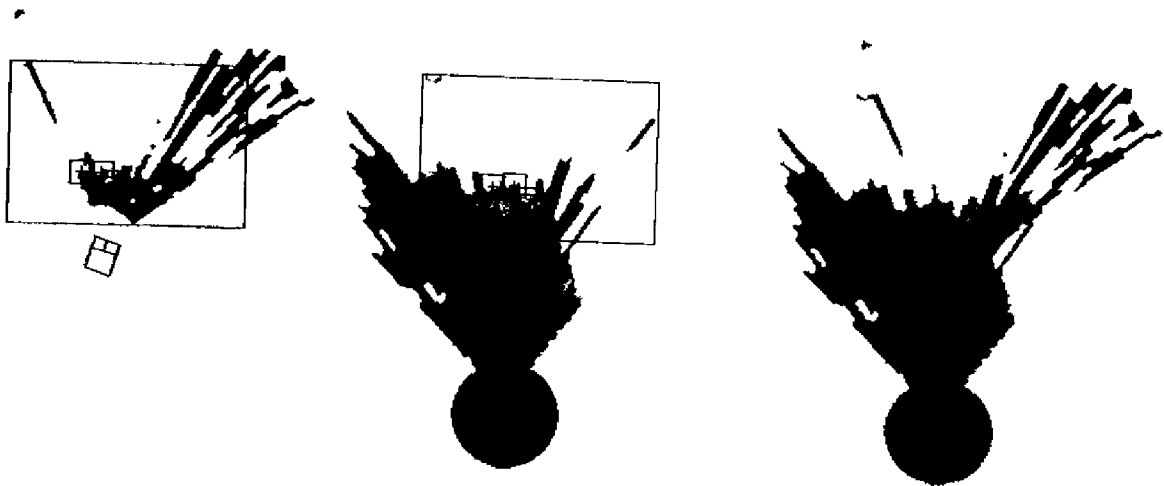
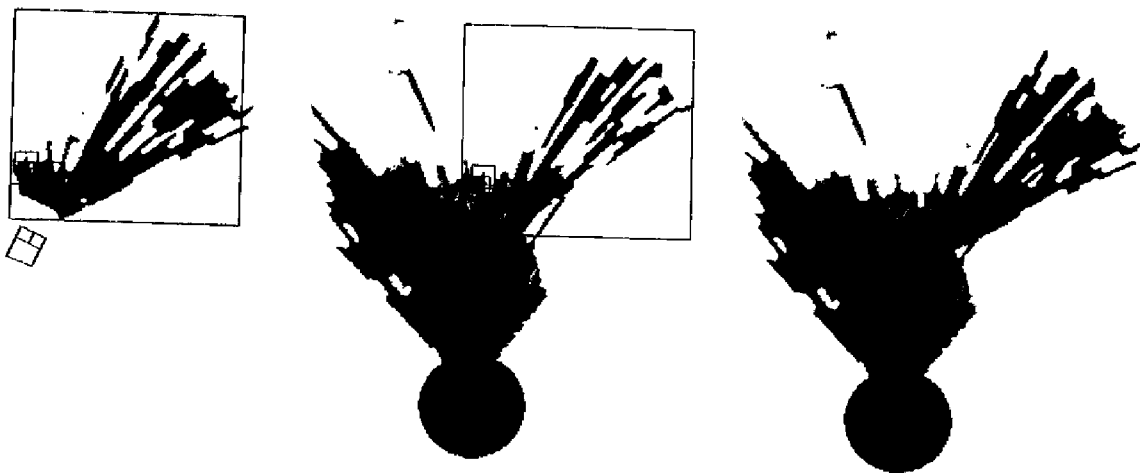
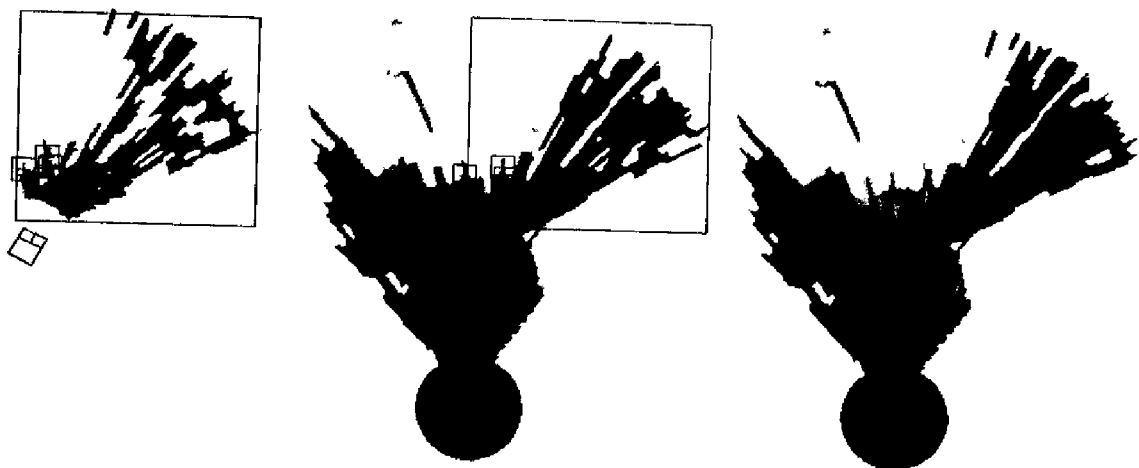


Figure 7.23: *Recalage-Fusion 3*

Figure 7.24: *Recalage-Fusion 4*Figure 7.25: *Recalage-Fusion 5*Figure 7.26: *Recalage-Fusion 6*



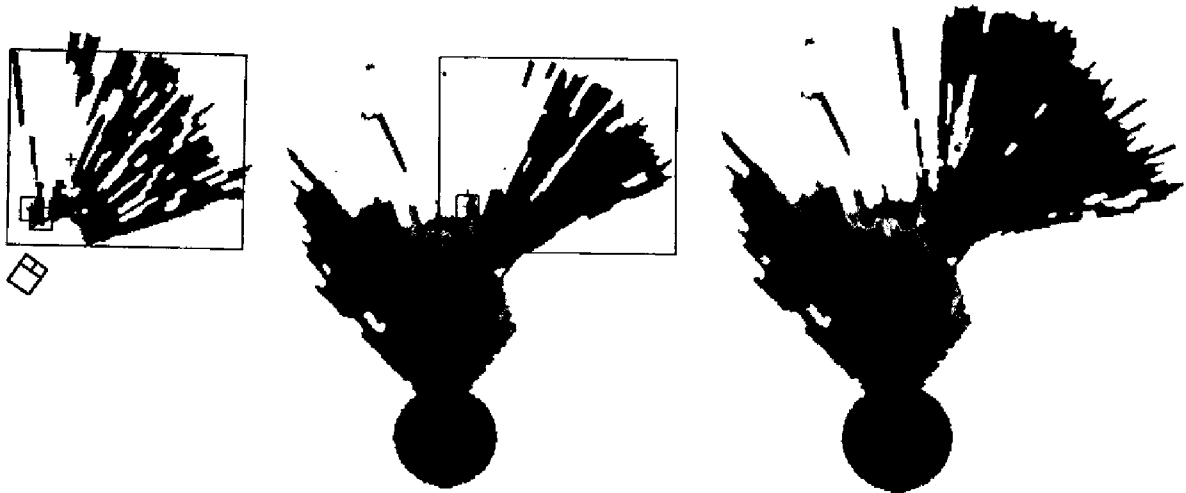


Figure 7.27: *Recalage-Fusion 7 et MNT local après 8 fusions*

## 7.6 Conclusion et orientations futures

Dans le chapitre 6, nous avons présenté une méthode de modélisation hiérarchique du terrain à l'aide de surfaces B-splines. Dans ce chapitre, nous avons montré comment ce modèle peut être utilisé pour le recalage de la position du robot à l'aide de caractéristiques de terrain.

Dans un premier temps, nous avons vu comment utiliser ce modèle pour une recherche hiérarchique d'amers (en focalisant la recherche sur des zones intéressantes du terrain), et détecter des caractéristiques comme des pics d'élévation ou des points de forte courbure. Pour la recherche de pics, on manipule les représentations implicites, très compactes, des surfaces B-splines; les temps de calcul sont ainsi très rapides. Pour le calcul d'attributs comme la courbure, on utilise les représentations analytiques explicites de la surface sur les différents carreaux qui la composent, et on fait un calcul analytique direct; les temps de calcul induits sont également assez rapides, relativement aux méthodes classiques.

Pour réaliser le recalage du robot, une focalisation sur les zones caractéristiques permet une coopération efficace avec les Modèles Numériques de Terrain pour réaliser des appariements iconiques de caractéristiques du terrain, conduisant à un recalage performant et rapide du robot, même en milieu fortement non structuré. On tient compte, dans l'estimation du recalage, de la qualité des appariements iconiques effectués à partir des différentes caractéristiques de terrain extraites (à travers les coefficients de corrélation obtenus).

Comme c'est le cas pour de très nombreux travaux concernant la localisation dans la littérature, les méthodes que nous avons proposées au chapitre 5 reposaient sur la détection de certaines caractéristiques, et sur leur mise en correspondance d'une image à l'autre. Une technique de fusion numérique permettait alors de construire incrémentalement un modèle de localisation contenant une *description aussi précise que possible des amers* rencontrés

dans l'environnement, et de recalculer la position du robot. Ces méthodes sont bien adaptées dans le cas où le robot se déplace dans un environnement semi-structuré. Mais dans ce chapitre, pour un robot évoluant dans un environnement non structuré, nous avons adopté une philosophie toute autre. Pour recalculer la position du robot, il n'est plus nécessaire de conserver avec la meilleure précision possible la position des amers et mettre en correspondance ces caractéristiques extraites d'une vue à l'autre. C'est une technique iconique avec focalisation de la recherche qui effectue les "appariements", en utilisant localement une description dense du terrain. Cette approche nous paraît bien adaptée en milieu non structuré, car les caractéristiques de terrain ne sont pas extraites avec une aussi grande précision que des caractéristiques de type structuré. Notons cependant que construire incrémentalement une carte des amers du terrain (pics par exemple) selon la méthode présentée au paragraphe 5.1 [Fillatreau 93a] garde un intérêt certain: la carte des pics, donnant leurs positions approximatives, peut permettre de décider à une position donnée du robot dans quelle direction il doit effectuer une acquisition (i.e. déterminer les *paramètres d'acquisition*) pour trouver dans son champ de vue des zones de terrain caractéristiques; la carte des pics peut ainsi permettre de générer des *plans de perception "orientés localisation"*<sup>2</sup>. Mais remettre à jour une position extrêmement précise de chaque amer dans le modèle local n'est plus un problème critique; ces positions approximatives donnent simplement des "pointeurs" vers des zones accidentées ou intéressantes de l'environnement.

Quelques extensions peuvent être apportées à la méthode de localisation que nous avons proposée dans ce chapitre. En particulier, comme cela a déjà été discuté en conclusion de la partie II, on pourrait prendre en compte de manière explicite les positions relatives des amers, et utiliser la notion de "groupe" de sommets ou de points de forte courbure, pour un meilleur contrôle de la fonction localisation. Actuellement, ces relations sont prises en compte implicitement uniquement (cf. paragraphe 7.4.2). Enfin, nous pourrions contraindre plus fortement la procédure d'appariement iconique: la construction incrémentale d'une carte des amers permet d'apparier au préalable ces amers d'une image à l'autre (cf. paragraphe 7.4.1), par des techniques de type recherche arborescente (cf. paragraphe 5.1.2); on pourrait envisager de n'effectuer d'appariement iconique qu'au voisinage des amers pour lesquels ces appariements préalables ont été réalisés.

Les travaux en cours ou futurs dans notre équipe tendent vers la mise au point de techniques de recalage s'appuyant sur d'autres types de caractéristiques, ou permettant un recalage des 6 paramètres de position et d'orientation du robot.

F. Nashashibi travaille actuellement sur une approche "contours", visant à extraire des MNT et mettre en correspondance des lignes de niveau. Dans une approche de ce type, nous souhaitons également étudier la possibilité d'extraire des lignes de crête ou de vallée (suivant des méthodes analogues à la recherche de pics) à partir de la représentation par B-splines. La recherche de telles caractéristiques est cependant plus légitime sur des représentations

<sup>2</sup> Plus généralement, la stratégie de perception doit évidemment tenir compte également de la position du but à atteindre, de la nature du terrain traversé, de l'attitude du robot, de la tâche à réaliser: localisation, navigation, ..., etc.

construites à partir d'images orbiteur (par exemple, pour l'aide à la navigation) que sur des représentations construites à partir d'acquisitions effectuées par le robot.

On peut penser à d'autres types de méthodes, pour recalculer les 6 paramètres de position et d'attitude du robot. Une première voie possible est de minimiser la distance entre les points de mesure correspondant à une acquisition instantanée, et la représentation surfacique du terrain; cette approche est similaire à celles adoptées par Szeliski [Szeliski 88] ou Hebert [Hebert 92] (cf. paragraphe 7.1). Une autre voie possible est d'envisager des approches plus iconiques. Nous envisageons notamment d'étudier la mise en correspondance de deux nuages de points, constitués par exemple des points de forte courbure de 2 MNT non redressés suivant l'information de verticale, et qui pourrait permettre d'éviter la lourdeur de méthodes iconiques consistant à mettre en correspondance *tous* les points de mesure (cf. par exemple [Hotz 93]).

Une autre voie explorée actuellement dans l'équipe est celle d'associer une plus grande information sémantique et topologique aux entités de recalage (nous rejoignons ici un souci général, déjà évoqué pour le recalage en milieu structuré au chapitre 5). Les travaux actuels de S. Betge Brezetz, R. Chatila et M. Devy [Betge-Brezetz 94] visent à isoler des objets "posés" sur le sol (rocher caractéristique par exemple), et à définir des relations topologiques entre eux. La reconnaissance de ces objets devrait permettre de réaliser le recalage de la position du robot, et les relations topologiques de guider la recherche des objets-amers servant de caractéristiques de recalage.

## Conclusion Générale

Dans ce travail de thèse, nous nous sommes intéressés aux fonctions *modélisation* et *localisation*, deux fonctionnalités du système de perception fondamentales pour un robot mobile autonome d'intervention ou d'exploration planétaire. Le problème de la localisation, en particulier, occupe une place centrale dans notre étude.

La réalisation de ces différentes fonctionnalités s'appuie sur l'utilisation d'informations multisensorielles. Nous avons utilisé des données fournies par des capteurs proprioceptifs de déplacement (odométrie + capteurs inertiels), et un télémètre laser.

Les représentations de l'environnement construites (et les primitives sur lesquelles elles sont basées) doivent être adaptées au type d'environnement rencontré, ainsi qu'à la tâche que l'on désire effectuer. Le type de robot qui nous intéresse doit affronter des environnements très variés, qui peuvent être localement fortement non structurés. Nous avons ainsi construit:

- Un modèle de localisation basé sur des primitives hétérogènes lorsque le robot évolue dans un environnement semi-structuré. L'utilisation de primitives hétérogènes permet de prendre en compte la diversité des amers rencontrés, et de rendre le recalage plus performant. Une stratégie de choix des amers a été proposée pour réduire l'*incertain*. La cohérence globale du modèle doit être vérifiée à travers une coopération numérique et sémantique des différents capteurs logiques (cf. chapitre 5).
- Un modèle de terrain basé sur l'utilisation hiérarchique de primitives non planes, les B-splines. On obtient un modèle analytique et compact, présentant d'intéressantes propriétés de localité et de continuité. Nous avons montré que la construction de ce modèle peut être dédiée à diverses tâches: la *construction incrémentale d'un modèle géométrique compact de l'environnement*, la *navigation*, la *classification de terrain*, et surtout la *localisation*. Les paramètres du modèle (nombre de niveaux du modèle hiérarchique, degré des surfaces polynômiales, critère de raffinage de la représentation d'un carreau d'une surface B-splines) sont choisis en fonction de la tâche à accomplir.

Dans le chapitre 6, l'algorithme de modélisation proposé vise à construire incrémentalement une représentation géométrique de la surface du terrain avec une certaine précision. La décision de raffiner la représentation sur un carreau dépend de l'erreur de modélisation des données de terrain.

Nous avons aussi souligné la possibilité d'utiliser le modèle de terrain pour la *classification* (notamment à travers une étude de la courbure, cf. chapitre 7). Le modèle de terrain proposé pourrait également être utilisé pour la *navigation* d'un robot mobile. On peut utiliser les expressions explicites de la surface polynomiale sur ses différents carreaux pour la planification 3D [Shiller 91, Siméon 93]. L'aspect multi-résolution pourrait permettre une *navigation adaptative*. On pourrait ainsi utiliser les carreaux du niveau supérieur, de grande taille, dans les régions peu accidentées où ces carreaux n'ont pas été raffinés (cf. chapitre 6), et utiliser les carreaux des niveaux inférieurs dans les régions où la représentation a été raffinée (régions où le terrain est accidenté). Dans le chapitre 7, enfin, l'utilisation du modèle est dédiée à la *localisation*. Pour cette dernière fonction, une approche multirésolution permet de focaliser la recherche de caractéristiques de terrain utilisables pour le recalage. Cette procédure d'extraction d'amers est extrêmement rapide car les représentations manipulées sont très compactes. Le recalage est réalisé par coopération avec une technique iconique. La focalisation de la recherche au voisinage des caractéristiques extraites permet d'obtenir des temps de calcul très rapides par rapport aux méthodes classiques dans la littérature.

Pour la fonction *localisation*, l'analyse des performances du recalage réalisé par les différents systèmes que nous avons réalisés, ainsi que celle des performances de trajectométrie selon le type de terrain traversé, fait apparaître clairement le besoin d'*adapter* le type de recalage mis en oeuvre à la nature des terrains traversés, comme c'est le cas pour la navigation. Nous pouvons établir un parallèle entre les différents modes de navigation nécessaires que nous avons définis dans le cadre de l'expérimentation EDEN (cf. paragraphe 1.4.3), et les "modes de localisation" à mettre en oeuvre:

- Si le terrain est essentiellement plat et "vide d'obstacles" (terrain libre), le robot évolue en mode de navigation *réactif*. Le terrain comporte peu d'amers potentiels, mais la centrale inertielle fournit des informations assez précises, car le terrain est plat et les trajectoires générées sont des lignes droites en direction du but. On peut par exemple se contenter d'un recalage approximatif sur la base d'amers lointains (amers, ligne d'horizon perçue par vision, ...).
- Si le terrain est "plat avec obstacles", le robot évolue en mode *planifié 2D*. Les déplacements du robot restent rapides. Les amers sont présents en petit nombre dans l'environnement. Les trajectoires générées sont plus complexes à cause de la présence d'obstacles (on utilise des méthodes de type "bitmap"). La trajectométrie est moins performante que dans le cas précédent (il y a des glissements et des patinages, notamment lors des virages) mais reste assez précise. Un recalage par mise en correspondance d'amers de type arbres, colonnes ou sommets, selon la procédure décrite au chapitre 5 (partie II) est adapté. Le recalage est rapide et la précision obtenue suffisante. Les amers sont peu nombreux et peu d'appariements ambigus seront trouvés car la trajectométrie est assez performante. Le recalage effectué est 2D (position  $(x, y)$  et cap  $\theta$ ).

La coordonnée  $z$  peut être recalée si on détecte des amers de type pics d'élévation. On peut aussi utiliser ici des amers que l'on distingue du sol (comme un rocher caractéristique), entre lesquels on définit des relations topologiques [Betge-Brezetz 94].

- Si le terrain est accidenté, le robot évolue en mode *planifié 3D*. Un modèle précis du terrain est nécessaire (du type MNT ou modèle de B-splines très précis); un planificateur 3D génère les trajectoires [Siméon 93, Dacre Wright 93]. Sur ce type de terrain, nos expérimentations montrent que la trajectométrie est peu performante. Par ailleurs, les amers de type pics d'élévation ou points de forte courbure peuvent être nombreux. Une procédure comme celle proposée au chapitre 5 conduirait à de nombreux appariements ambigus et serait d'une précision insuffisante. Un recalage 3D précis est nécessaire; il faut utiliser de préférence les amers situés à proximité du robot. Une précision à l'échelle du MNT est requise, et la méthode de recalage que nous avons proposée au chapitre 7 est plus adaptée.

Enfin, il est nécessaire d'introduire la notion de *contrôle de perception* dans les différentes fonctionnalités de la perception, et en particulier la localisation. Il s'agit là d'un travail très important, qui pourrait constituer un travail de thèse (ou plusieurs). Quelques idées peuvent déjà être avancées:

- Un robot mobile d'intervention doit pouvoir affronter différents types d'environnements, ce qui nous a conduits à développer deux modules de localisation distincts, l'un approprié pour un environnement de type semi-structuré, l'autre adapté à un environnement non structuré (terrain accidenté). Il faut pouvoir décider, à un moment donné, quel module de localisation doit être mis en oeuvre. Nos travaux futurs devront étudier la mise au point d'un opérateur spécialisé permettant de résoudre ce problème, en tenant compte du type d'environnement traversé. Pour le moment, une solution intermédiaire pourrait être rapidement développée. En effet, les temps de traitement induits par les méthodes de détection d'amers que nous avons proposées permettent un renvoi très rapide d'indices. On pourrait choisir, en fonction du nombre d'amers d'un type donné détectés, quel est la méthode de recalage la plus appropriée. La question de la fusion des résultats de recalage retournés par plusieurs capteurs logiques mérite également d'être posée, dans le cas où des amers de type structuré et non structuré sont présents. Le problème de l'élaboration d'une stratégie de choix des amers se posera alors. Quelques éléments de réponse ont été apportés dans la partie II.
- Une autre question intéressante, enfin, est posée par l'analyse de la relation entre le nombre de perceptions d'un amer d'une part, et son incertitude et son imprécision d'autre part (cf. paragraphes 1.3 et 5.1.3). Cette analyse mérite d'être approfondie. Il serait notamment intéressant d'établir une relation entre la perception et le but du déplacement. On pourrait ainsi par exemple décider de faire des acquisitions plus

nombreuses dans les zones du terrain comprenant des configurations d'amers discriminantes, ou chercher à déterminer la position de tous les amers avec une précision donnée (perception dirigée par les incertitudes), ...

## Annexes





## Annexe A

# Calibrage intrinsèque du télémètre HILARE II / ADAM

### A/ Mode de déflexion

Le modèle géométrique du capteur doit dans un premier temps tenir compte du mode de déflexion mécanique permettant le balayage de la scène par le faisceau laser. Deux modes d'acquisitions, illustrés par la figure A.1, sont possibles:

- le mode *non corrélé* (i.e. balayages axiaux non corrélés): ici, le balayage en azimut est effectué à l'aide d'une rotation autour de l'axe vertical (direction de la pesanteur), quelle que soit la valeur de la visée en site.
- le mode *corrélé*; ici, le balayage en azimut se fait par une rotation autour de l'axe  $z_\phi$  correspondant à un angle de visée en site égal à  $\phi$ .

Parmi les capteurs que nous avons utilisés au LAAS pour expérimenter, le capteur d'HILARE I fonctionne en mode corrélé, alors que le capteur IBEO, développé par KAE <sup>1</sup> et utilisé dans le cadre du projet européen SKIDS, le capteur ALIS, développé par le LETI et que nous avons utilisé dans le cadre du projet AMR (où il équipa le démonstrateur de phase 2c ADAM), et les capteurs équipant actuellement HILARE II et ADAM, fonctionnent en mode non corrélé.

### B/ Système mécanique de déflexion

De nombreux offsets mécaniques (translations et rotations), dus à la complexité du système mécanique de balayage, doivent ici être modélisés.

- la platine azimut a une certaine orientation par rapport au robot, modélisée par les deux angles  $(\alpha, \beta)$ .

---

<sup>1</sup>Krupp Atlas Elektronik

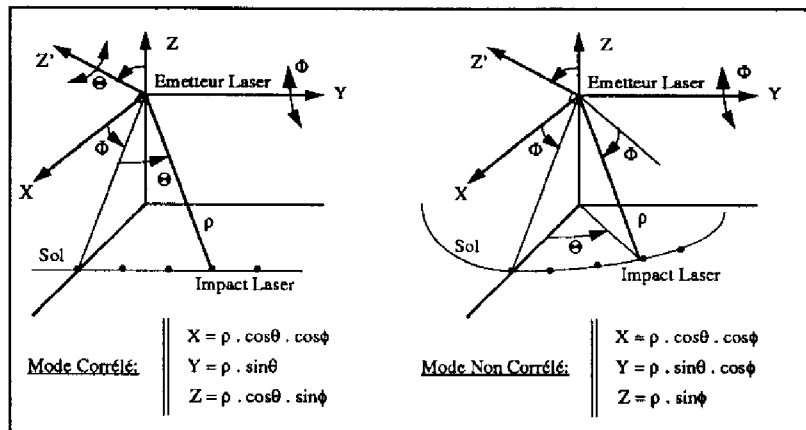


Figure A.1: Modes d'acquisition d'une image laser

- les axes de balayage en site et azimuth ne sont pas concourants, ce que nous modélisons par les offsets de translation  $(x_e, y_e)$ .
- le miroir site possède un offset angulaire  $\delta$ .
- les droites porteuses du faisceau laser et de l'axe de rotation du miroir site sont séparés d'une distance  $S$  à identifier. Par contre, nous négligeons l'angle  $\mu$  qu'elles forment.

### C/ Imprécision de la mesure laser

Nous savons que le faisceau laser est divergent. Par ailleurs, nous avons pu observer expérimentalement la présence d'un offset quasi-systématique et d'une dérive en fonction de la distance de mesure, sur la mesure de profondeur retournée par le capteur. Nous avons donc choisi pour modèle de la mesure de profondeur:

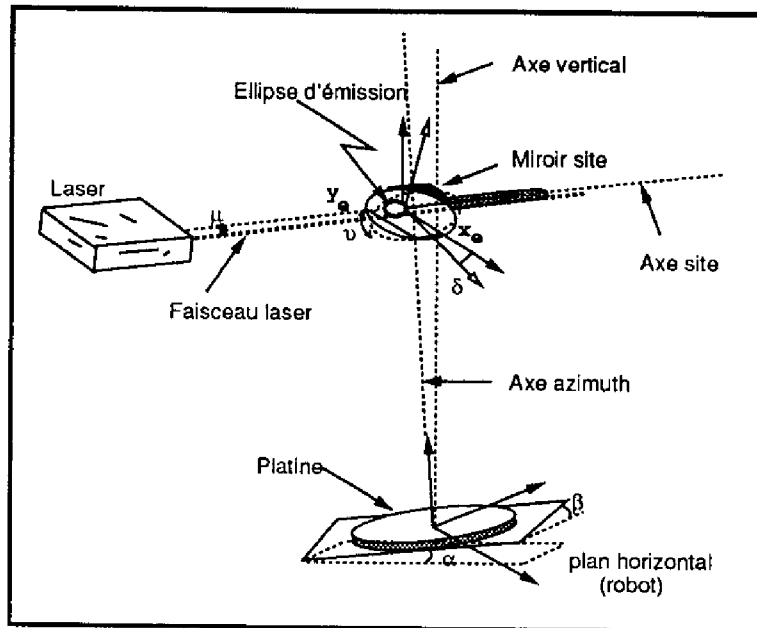
$$\rho_{mesure} = a \cdot \rho_{theorique} + \rho_0$$

où  $\rho_{mesure}$  est la profondeur retournée par le capteur et  $\rho_{theorique}$  la profondeur réelle qu'aurait théoriquement renvoyée le capteur en l'absence de toute imperfection mécanique ou optique. Nous avons déterminé expérimentalement les valeurs des paramètres du modèle, et avons obtenu [Nashashibi 93a]:

$$\begin{cases} a & = & 0.99641 \\ \rho_0 & = & 0.20237(m.) \end{cases}$$

Au total, le modèle géométrique élaboré pour le télémètre d'HILARE II est donc celui décrit par la figure A.2.

Les équations de passage des coordonnées sphériques en coordonnées cartésiennes sont:

Figure A.2: *Modèle géométrique du télémètre du robot HILARE II*

$$\begin{cases} X = (a \cdot \rho + \rho_0) [ (\cos \beta \cdot \cos \phi + \sin \phi \cdot \cos \alpha \cdot \sin \beta) \cos \theta + \sin \phi \cdot \sin \alpha \cdot \sin \theta ] + \sqrt{X_e^2 + Y_e^2} \cdot \sin \theta \\ Y = (a \cdot \rho + \rho_0) [ (\cos \beta \cdot \cos \phi + \sin \phi \cdot \cos \alpha \cdot \sin \beta) \sin \theta - \sin \phi \cdot \sin \alpha \cdot \cos \theta ] + \sqrt{X_e^2 + Y_e^2} \cdot \cos \theta \\ Z = (a \cdot \rho + \rho_0) [ -\cos \phi \cdot \sin \beta + \sin \phi \cdot \cos \alpha \cdot \cos \beta ] \end{cases}$$

Nous avons utilisé pour le calibrage une planche verticale comme support expérimental, et le filtre de Kalman comme outil d'estimation. Les résultats obtenus pour les paramètres intrinsèques du modèle simplifié finalement mis en oeuvre sont présentés dans le tableau suivant:

<i>paramètre (type)</i>	<i>symbole</i>	<i>valeur</i>	
Offsets électroniques	$x_e, y_e$ (mm)	115	-323
Offsets géométriques	$S_x, S_y$	10	0
Offsets angulaires / ( $O_x, O_y$ )	$\alpha, \beta$	$0.5^\circ$	$-5.28^\circ$

F. Nashashibi a décrit en annexe de sa thèse [Nashashibi 93a] la procédure et les expérimentations que nous avons mises en oeuvre pour le calibrage du télémètre laser d'HILARE II, et le télémètre de même type équipant ADAM actuellement. Cette étude statistique, à la fois qualitative et quantitative, nous a permis (à partir d'un très grand nombre de mesures !!) d'une part d'estimer les paramètres du modèle élaboré ci-dessus, et leur incertitude, mais aussi de fournir une description statistique de la dispersion des mesures en fonction de la distance, en fonction de l'angle d'attaque (pour donner une mesure des phénomènes de specularité), en fonction du type de matériau sur lequel se fait l'impact laser, de la specularité

au sol, ...

## Annexe B

# Calibrage extrinsèque robot-capteur par l'approche de Tsai et Lenz

### B.1 La procédure de calibrage

Notre but ici est de déterminer la transformation homogène 3D  $T_{rc}$  entre un repère cartésien lié au capteur et un repère cartésien lié au robot (nous avons choisi par commodité le repère odométrique, dont l'origine se trouve entre les deux roues motrices centrales dans le cas d'HILARE II (cf. figure 2.2).

Classiquement, le problème est formalisé en posant, pour deux positions 3D données du robot mobile (figure 2.2), la relation:

$$\mathbf{T}_{rc} \times \mathbf{T}_{cc'} = \mathbf{T}_{rr'} \times \mathbf{T}_{rc} \quad (\text{B.1})$$

où  $T_{cc'}$  et  $T_{rr'}$  décrivent respectivement les mouvements effectués par le capteur et par le robot d'une position à l'autre de celui-ci. Ces descriptions sont faites à l'aide de matrices homogènes [Paul 81].

L'approche de Tsai et Lenz découple les phases de détermination de la rotation et de la translation. On notera  $R_{rc}$  et  $t_{rc}$  la matrice de rotation 3x3 et le vecteur translation 3x1 décrivant la transformation  $T_{rc}$ . Le principe de base est de faire effectuer au robot des déplacements 3D connus. Si on peut reconnaître à chaque position du robot un objet ou des éléments de l'environnement permettant de déterminer le déplacement du capteur, alors la connaissance des déplacements du capteur et du déplacement du robot permettent de calculer  $T_{rc}$  à l'aide de la relation B.1. Au moins deux paires de positions du robot sont nécessaires.

Dans la suite, étant donnée une matrice de rotation  $R$ , nous noterons  $P$  le vecteur de dimension 3 définissant l'axe de la rotation (et de norme  $|2 \cdot \sin \frac{\theta}{2}|$ ) - où  $\theta$  est l'angle de

rotation -, choisi ainsi par commodité et parce-qu'il permet d'exprimer  $R$  de manière simple en fonction de  $P$ .  $P'$ , pour sa part, désigne un autre vecteur intermédiaire colinéaire à  $P$ . L'utilisation des deux vecteurs intermédiaires  $P$  et  $P'$  permet de simplifier les équations qui suivent.

La procédure proposée par Tsai et Lenz se décompose alors selon les 6 étapes suivantes:

1. Détermination de  $P_c$  et  $P_r$  associés aux matrices  $R_{cc'}$  et  $R_{rr'}$ , à partir des vecteurs propres associés à la valeur propre 1 de ces deux matrices.
2. Calcul du vecteur  $P'_{rc}$ , en résolvant, par la méthode des moindres carrés linéaires, le système suivant:

$$(P_r + P_c) \wedge P'_{rc} = P_c - P_r$$

3. Calcul de l'angle de rotation robot-capteur  $\theta$ , qui vérifie:

$$\theta = 2. \arctan \| P'_{rc} \|$$

4. Calcul du vecteur  $P_{rc}$  avec:

$$P_{rc} = \frac{2.P'_{rc}}{\sqrt{1 + |P'_{rc}|^2}}$$

5. Détermination de  $R_{rc}$  à partir du vecteur normal  $(n_1, n_2, n_3)$  colinéaire à  $P_{rc}$ , selon:

$$R_{rc} = \begin{pmatrix} n_1^2 + (1 - n_1^2) \cos \theta & n_1 n_2 (1 - \cos \theta) - n_3 \sin \theta & n_1 n_3 (1 - \cos \theta) - n_2 \sin \theta \\ n_1 n_2 (1 - \cos \theta) + n_3 \sin \theta & n_2^2 + (1 - n_2^2) \cos \theta & n_2 n_3 (1 - \cos \theta) - n_1 \sin \theta \\ n_1 n_3 (1 - \cos \theta) - n_2 \sin \theta & n_2 n_3 (1 - \cos \theta) + n_1 \sin \theta & n_3^2 + (1 - n_3^2) \cos \theta \end{pmatrix}$$

6. Détermination du vecteur translation robot-capteur  $t_{rc}$  par résolution, aux moindres carrés, du système:

$$(R_{rr'} - Id).t_{rc} = R_{rc}.t_{cc'} - t_{rr'}$$

## B.2 Dispositif expérimental et résultats

Dans notre cas, les déplacements du robot sont mesurés par trajectométrie:

- odomètre + inclinomètres dans le cas d'HILARE II;
- odomètres + centrale inertielle dans le cas d'ADAM.

Les mesures de déplacement du capteur peuvent être effectuées de deux manières:

- Par reconnaissance et localisation, après chaque déplacement, d'un objet polyédrique dont on connaît parfaitement le modèle, à l'aide d'un système multisensoriel composé d'un télémètre laser et d'une caméra monochrome (travaux de J. Colly et M. Devy [Colly 91a]).
- Par segmentation des images laser successives en régions planes, et par appariement des plans d'une image à l'autre en utilisant l'information trajectométrique (travaux de F. Nashashibi et M. Devy [Nashashibi 93b]).

Nous avons expérimenté avec les deux approches. La figure B.1 montre l'expérimentation avec la deuxième approche dans le cas d'HILARE II, dans l'environnement de notre salle de robotique. Dans le cas d'ADAM, nous avons adopté la première, car l'expérimentation se déroulait alors à l'extérieur!

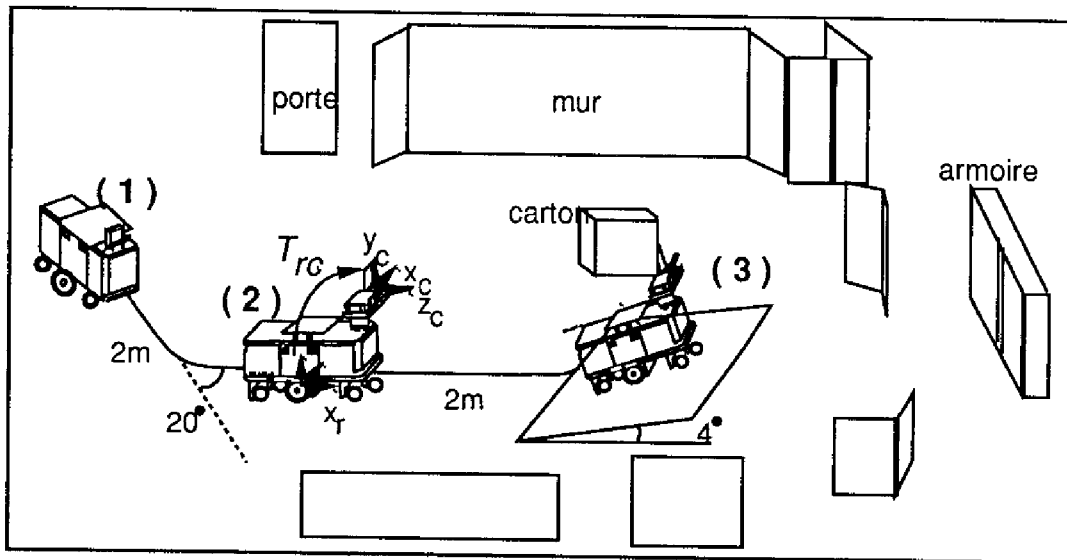


Figure B.1: *Calibrage extrinsèque: scénario expérimental*

Les résultats que nous avons obtenus pour la transformation robot-capteur dans le cas d'HILARE II sont finalement:

$$T_{RC} = \begin{pmatrix} 0.9957568 & -0.0000226 & 0.092023 & 0.4235507 \\ 0.0007802 & 0.9961946 & -0.0086895 & 0.0013401 \\ -0.0920197 & 0.0087244 & 0.9957189 & 0.9506673 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matrice de rotation est proche de la matrice identité, ce qui correspond au fait que le capteur est posé "à peu près" horizontalement sur le robot, et "regarde" dans l'axe devant lui. Le vecteur de droite donne en mètres la hauteur et la translation horizontale par rapport au repère odométrique (le capteur se trouve à une hauteur approximative de 95cm, et 42cm environ "devant" le centre des deux roues motrices).



## Annexe C

# Le filtre de Kalman généralisé: l'outil YAFF

Mis au point et développé dans notre équipe par Ph. Moutarlier et R. Chatila [Moutarlier 91], l'outil de filtrage numérique *YAFF*, basé sur le filtre de Kalman, peut être utilisé dans le cas de non linéarités, et en présence de bruits corrélés et colorés. Il prend en compte explicitement les corrélations entre un nombre quelconque de primitives, reliées entre elles par une équation de mesure de la forme:

$$z = f(x_1, x_2, \dots, x_n)$$

La valeur théorique  $z$  est en fait une connaissance sur l'état du système qui induit une réestimation des primitives mises en jeu. L'ancienne estimée d'un vecteur  $\hat{x}$  est corrigée d'une valeur proportionnelle à l'écart de la mesure  $\delta$  entre la prédiction  $z$ , connaissant l'état courant, et la valeur théorique.

$$\hat{x}' = \hat{x} + \Gamma_{xz} \Gamma_{zz}^{-1} \delta$$

avec

$$\delta = z - \hat{z}$$

$$\Gamma_{xz} = E[\varepsilon_x \cdot \delta^T] \quad \Gamma_{zz} = E[\delta \cdot \delta^T]$$

La variance de la nouvelle estimée vaut :

$$\sigma_{x'} = \sigma_x - \Gamma_{xz} \Gamma_{zz}^{-1} \Gamma_{xz}^T$$

Et sa corrélation avec un autre vecteur  $x_i$  devient :

$$\sigma_{x',x_i} = \sigma_{x,x_i} - \Gamma_{x_i z} \Gamma_{zz}^{-1} \Gamma_{xz}^T$$

Le filtre est l'estimateur *linéaire* qui donne l'estimation non biaisée de variance minimale. Les équations précédentes en donnent la formulation exacte sans restriction sur la forme

de la densité de probabilité (en particulier, pas de supposition gaussienne). Dans le cas de distributions gaussiennes, le filtre est, de *tous* les estimateurs, celui qui donne l'estimation de variance minimale, et il correspond à l'estimateur Bayésien du maximum de vraisemblance *a posteriori*.

La formulation ci-dessus montre que pour réestimer un vecteur  $x$ , il suffit qu'il soit corrélé à au moins un vecteur de la mesure (i.e.  $\Gamma_{xz} \neq 0$ ), même s'il ne fait pas partie lui-même directement de la mesure. De même, si on réestime un vecteur quelconque, il devient corrélé à tous les vecteurs corrélés à la mesure (ou sa corrélation évolue s'il l'était déjà).

Le filtre offre une très grande souplesse d'utilisation; il permet par exemple de ne réestimer que certains vecteurs, ou d'utiliser plusieurs fois les mêmes mesures pour estimer des vecteurs différents, tout en obtenant le même résultat qu'après une réestimation simultanée, sous réserve de mettre à jour toutes les corrélations après chaque réestimation [Grandjean 91].

## Annexe D

# Extrémum d'une courbe ou surface B-splines

Les calculs sont menés ici dans le cas monodimensionnel, les résultats obtenus pour une courbe B-splines se généralisant directement au cas des surfaces de produit tensoriel. Nous donnons le principe du calcul dans le cas où le degré vaut 1 ou 2; ces cas sont les cas les plus couramment utilisés en pratique, avec le cas du degré 3 que nous ne présentons pas ici, afin d'alléger la présentation.

Pour alléger la présentation encore, nous nous plaçons dans des intervalles de "régime permanent" où tous les points d'arrêt sont de multiplicité 1; les cas en bordure de domaine, un peu différents à cause des noeuds extrêmes qui sont de multiplicité supérieure à 1, ont été également traités, mais ne sont pas présentés ici.

### D.1 Cas du degré 2

Soit une courbe B-splines polynômiale de degré 2 par morceaux, définie par:

$$y = P(x) = \sum_i a_i B_{i,2}(x)$$

Supposons que  $a_j$  soit un maximum local dans la suite de coefficients de B-splines, c'est à dire que  $a_{j-1} < a_j$  et  $a_{j+1} < a_j$

Plaçons-nous dans l'intervalle  $[x_{j+1} \quad x_{j+2}]$  où les 3 B-splines  $B_{j-1}(x)$ ,  $B_j(x)$  et  $B_{j+1}(x)$  sont non nulles.

Sur cet intervalle, nous avons [Pavlidis 81]:

$$B_{j-1}(x) = \frac{(x_{j+2} - x)^2}{(x_{j+2} - x_j) \times (x_{j+2} - x_{j+1})}$$

$$B_j(x) = \frac{(x - x_j) \times (x_{j+2} - x)}{(x_{j+2} - x_j) \times (x_{j+2} - x_{j+1})} + \frac{(x_{j+3} - x) \times (x - x_{j+1})}{(x_{j+3} - x_{j+1}) \times (x_{j+2} - x_{j+1})}$$

$$B_{j+1}(x) = \frac{(x - x_{j+1})^2}{(x_{j+3} - x_{j+1}) \times (x_{j+2} - x_{j+1})}$$

et la spline prend sur ce même intervalle l'expression:

$$y = P(x) = a_{j-1}B_{j-1,2}(x) + a_j B_{j,2}(x) + a_{j+1}B_{j+1,2}(x)$$

On obtient alors:

$$\frac{\partial P(x)}{\partial x} = (a_{j-1} - 2a_j + a_{j+1}) \cdot x - a_{j-1} \cdot x_{j+2} + \frac{a_j}{2} \cdot (x_j + x_{j+1} + x_{j+2} + x_{j+3}) - a_{j+1} \cdot x_{j+1}$$

qui s'annule pour:

$$x_{max} = \frac{a_{j-1}x_{j+2} - \frac{a_j}{2}(x_j + x_{j+1} + x_{j+2} + x_{j+3}) + a_{j+1}x_{j+1}}{a_{j-1} - 2a_j + a_{j+1}}$$

Dans le cas uniforme (où la distance entre deux points d'arrêt successifs est une constante égale à  $\Delta x$ ), on obtient:

$$x_{max} = x_{j+2} - \frac{\Delta x}{\frac{a_j - a_{j-1}}{a_j - a_{j+1}} + 1}$$

qui confirme la propriété intuitive que  $x_{max} \in [(x_{j+1} + x_{j+2})/2 \quad x_{j+2}]$  si  $a_{j+1} > a_{j-1}$ ,  $x_{max} = \frac{x_{j+1} + x_{j+2}}{2}$  si  $a_{j-1} = a_{j+1}$ , et  $x_{max} \in [x_{j+1} \quad (x_{j+1} + x_{j+2})/2]$  si  $a_{j+1} < a_{j-1}$ .

En particulier, si  $a_{j-1} = a_{j+1}$ , on retrouve que  $x_{max} = x_{j+2} - \frac{\Delta x}{2}$ , c'est à dire que le maximum est atteint au milieu de l'intervalle  $[x_{j+1} \quad x_{j+2}]$ , point où la B-spline  $B_{j,2}(x)$  atteint son maximum.

## D.2 Cas du degré 1

Le résultat obtenu dans ce cas est bien plus simple encore; la figure suivante montre l'allure obtenue pour une spline d'ordre 1 pour laquelle un maximum  $a_j$  apparaît dans la séquence de coefficients sur les B-splines (nous nous plaçons par exemple ici dans le cas où  $a_{j-1} < a_{j+1} < a_j$ ). Il vient de façon immédiate que la spline présente un maximum local en  $x_{j+1}$ .

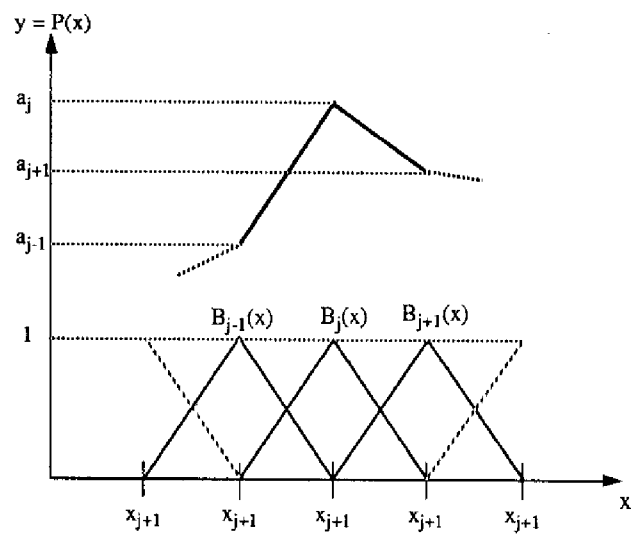


Figure D.1: *Maximum d'une courbe B-splines de degré 1*

## Annexe E

# Courbure en un point d'une surface B-splines

Soit une surface B-splines quelconque, définie comme combinaison linéaire de  $(k_x + m_x)(k_y + m_y)$  B-splines de base:

$$z = S(x, y) = \sum_{i=0}^{k_x+m_x-1} \sum_{j=0}^{k_y+m_y-1} a_{i,j} B_{i,m_x}(x) B_{j,m_y}(y)$$

ou encore:

$$z = S(x, y) = \sum_{i=0}^{k_x+m_x-1} \sum_{j=0}^{k_y+m_y-1} a_{i,j} B_{i,j}(x, y)$$

où  $B_{i,j}(x, y) = B_{i,m_x}(x) \times B_{j,m_y}(y)$ .

Les expressions analytiques des B-splines de degré quelconque, deux vecteurs de noeuds (et leurs multiplicités) étant donnés (et définissant un réseau de carreaux sur la surface - cf. chapitre 6-), peuvent être déterminées à partir des équations 6.1 et 6.2. Dans le cas uniforme, en particulier, les expressions analytiques des B-splines de faible degré sont bien connues. Sur un carreau donné de la surface, l'expression analytique explicite  $z = f(x, y)$  de la surface peut être déterminée à partir des B-splines qui sont non nulles sur ce domaine et des coefficients de la combinaison linéaire. Les équations 7.3 permettent alors de calculer les courbures Gaussienne et Moyenne en un point quelconque de la surface.

Nous souhaitons que la courbure de la surface soit continue, et nous utilisons donc ici des B-splines bicubiques. Les multiplicités des noeuds extrêmes suivant  $x$  et  $y$  sont égales à 4. La distribution des noeuds est uniforme. Les 3 B-splines extrêmes suivant chaque direction sont particulières, les autres ayant toutes la même allure (voir figure 6.2). Sur un carreau  $i, j$  (délimité par les noeuds  $x_i$  et  $x_{i+1}$  suivant  $x$ ,  $y_i$  et  $y_{i+1}$  suivant  $y$ ) où seules des B-splines de ce dernier type sont non nulles, les expressions obtenues pour les courbures gaussienne et moyenne sont les suivantes:

$$K = \frac{f_{xx}f_{yy} - f_{xy}^2}{(1 + f_x^2 + f_y^2)^2}$$

$$H = \frac{(1 + f_x^2)f_{yy} + (1 + f_y^2)f_{xx} - 2f_x f_y f_{xy}}{(1 + f_x^2 + f_y^2)^{3/2}}$$

avec:

$$f_x = (a_{i,j}.P20(u).P30(v) + a_{i,j-1}.P20(u).P31(v) + a_{i,j-2}.P20(u).P32(v) + a_{i,j-3}.P20(u).P33(v) + a_{i-1,j}.P21(u).P30(v) + a_{i-1,j-1}.P21(u).P31(v) + a_{i-1,j-2}.P21(u).P32(v) + a_{i-1,j-3}.P21(u).P33(v) + a_{i-2,j}.P22(u).P30(v) + a_{i-2,j-1}.P22(u).P31(v) + a_{i-2,j-2}.P22(u).P32(v) + a_{i-2,j-3}.P22(u).P33(v) + a_{i-3,j}.P23(u).P30(v) + a_{i-3,j-1}.P23(u).P31(v) + a_{i-3,j-2}.P23(u).P32(v) + a_{i-3,j-3}.P23(u).P33(v)) \times u_x$$

$$f_y = (a_{i,j}.P30(u).P20(v) + a_{i,j-1}.P30(u).P21(v) + a_{i,j-2}.P30(u).P22(v) + a_{i,j-3}.P30(u).P23(v) + a_{i-1,j}.P31(u).P20(v) + a_{i-1,j-1}.P31(u).P21(v) + a_{i-1,j-2}.P31(u).P22(v) + a_{i-1,j-3}.P31(u).P23(v) + a_{i-2,j}.P32(u).P20(v) + a_{i-2,j-1}.P32(u).P21(v) + a_{i-2,j-2}.P32(u).P22(v) + a_{i-2,j-3}.P32(u).P23(v) + a_{i-3,j}.P33(u).P20(v) + a_{i-3,j-1}.P33(u).P21(v) + a_{i-3,j-2}.P33(u).P22(v) + a_{i-3,j-3}.P33(u).P23(v)) \times v_y$$

$$f_{xx} = (a_{i,j}.P10(u).P30(v) + a_{i,j-1}.P10(u).P31(v) + a_{i,j-2}.P10(u).P32(v) + a_{i,j-3}.P10(u).P33(v) + a_{i-1,j}.P11(u).P30(v) + a_{i-1,j-1}.P11(u).P31(v) + a_{i-1,j-2}.P11(u).P32(v) + a_{i-1,j-3}.P11(u).P33(v) + a_{i-2,j}.P12(u).P30(v) + a_{i-2,j-1}.P12(u).P31(v) + a_{i-2,j-2}.P12(u).P32(v) + a_{i-2,j-3}.P12(u).P33(v) + a_{i-3,j}.P13(u).P30(v) + a_{i-3,j-1}.P13(u).P31(v) + a_{i-3,j-2}.P13(u).P32(v) + a_{i-3,j-3}.P13(u).P33(v)) \times u_{xx}$$

$$f_{yy} = (a_{i,j}.P30(u).P10(v) + a_{i,j-1}.P30(u).P11(v) + a_{i,j-2}.P30(u).P12(v) + a_{i,j-3}.P30(u).P13(v) + a_{i-1,j}.P31(u).P10(v) + a_{i-1,j-1}.P31(u).P11(v) + a_{i-1,j-2}.P31(u).P12(v) + a_{i-1,j-3}.P31(u).P13(v) + a_{i-2,j}.P32(u).P10(v) + a_{i-2,j-1}.P32(u).P11(v) + a_{i-2,j-2}.P32(u).P12(v) + a_{i-2,j-3}.P32(u).P13(v) + a_{i-3,j}.P33(u).P10(v) + a_{i-3,j-1}.P33(u).P11(v) + a_{i-3,j-2}.P33(u).P12(v) + a_{i-3,j-3}.P33(u).P13(v)) \times v_{yy}$$

$$f_{xy} = (a_{i,j}.P20(u).P20(v) + a_{i,j-1}.P20(u).P21(v) + a_{i,j-2}.P20(u).P22(v) + a_{i,j-3}.P20(u).P23(v) + a_{i-1,j}.P21(u).P20(v) + a_{i-1,j-1}.P21(u).P21(v) + a_{i-1,j-2}.P21(u).P22(v) + a_{i-1,j-3}.P21(u).P23(v) + a_{i-2,j}.P22(u).P20(v) + a_{i-2,j-1}.P22(u).P21(v) + a_{i-2,j-2}.P22(u).P22(v) + a_{i-2,j-3}.P22(u).P23(v) + a_{i-3,j}.P23(u).P20(v) + a_{i-3,j-1}.P23(u).P21(v) + a_{i-3,j-2}.P23(u).P22(v) + a_{i-3,j-3}.P23(u).P23(v)) \times u_x v_y$$

où:

$$\begin{array}{lll} P30(u) = u^3/6 & P20(u) = u^2/2 & P10(u) = u \\ P31(u) = (-3u^3 + 3u^2 + 3u + 1)/6 & P21(u) = (-3u^2 + 2u + 1)/2 & P11(u) = (-3u + 1) \\ P32(u) = (3u^3 - 6u^2 + 4)/6 & P22(u) = (3u^2 - 4u)/2 & P12(u) = (3u - 2) \\ P33(u) = (-u^3 + 3u^2 - 3u + 1)/6 & P23(u) = (-u^2 + 2u - 1)/2 & P13(u) = (-u + 1) \end{array}$$

Dans ces expressions,  $u$  et  $v$  désignent les variables réduites:  $u = \frac{x-x_{i-1}}{x_i-x_{i-1}}$  et  $v = \frac{y-y_{i-1}}{y_i-y_{i-1}}$ .

Par ailleurs, les indices  $x$  ou  $y$  désignent les dérivées partielles: par exemple,  $f_x = \frac{\partial f}{\partial x}$ ,  $f_{xx} = \frac{\partial^2 f}{\partial x^2}$ ,  $f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$ .

Pour les carreaux situés près des frontières de la surface B-splines, sur lesquels des B-splines particulières interviennent dans le calcul de la courbure, nous avons tenu compte des expressions particulières de ces B-splines.

## Références bibliographiques

- [Achour 87] K. Achour. *Localisation absolue et relative d'un robot mobile dans un univers à deux dimensions*. Thèse de docteur ingénieur, Institut National des Sciences Appliquées de Rennes, Janvier 1987.
- [Aggarwal 86] J.K. Aggarwal & Y.C. Kim. *Rectangular parallelepiped coding : A volumetric representation of three-dimensionnal objects*. International Journal of Robotics and Automation - IASTED, vol. RA2, no. 3, pages 127-134, Septembre 1986.
- [Ahlberg 67] J. H. Ahlberg, E. N. Nilson & J. L. Walsh. *The Theory of Splines and their Applications*. Academic Press, New York, NY, 1967.
- [Amini 88] A. A. Amini, S. Tehrani & T. E. Weymouth. *Using Dynamic Programming for Minimizing the Energy of Active Contours in the Presence of Hard Constraints*. In Second International Conference on Computer Vision, Tampa, Florida (USA), pages 95-99, Décembre 1988.
- [Argouarc'h 89] J.R. Argouarc'h, F.X. Pôtel & F. Richard. *Mechanical models for the locomotion of a four-articulated-track robot*. In SPIE Mobile Robots IV, Novembre 1989.
- [Asada 88] M. Asada. *Building a 3-D World Model for a Mobile Robot from Sensory Data*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), pages 918 - 923, Avril 1988.
- [Ayache 87a] N. Ayache & O. Faugeras. *Maintaining representations of the environment of a mobile robot*. In Robotics Research : The Fourth International Symposium, Santa Cruz (USA), Août 1987.
- [Ayache 87b] N. Ayache & F. Lustman. *Fast and reliable passive trinocular stereovision*. In First International Conference on Computer Vision, London (U.K.), pages 422-427, Juin 1987.
- [Ayache 87c] N. Ayache & F. Lustman. *Trinocular Stereovision*. In 10th International Joint Conference on Artificial Intelligence (IJCAI), Milan (Italy), Août 1987.
- [Ayache 88] N. Ayache. *Construction et Fusion de Représentations Visuelles 3D. Applications à la Robotique Mobile*. Thèse d'état, Université de Paris-Sud, 1988.
- [Badler 79] N. Badler & J. O'Rourke. *Decomposition of 3-Dimensional Objects into Spheres*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, no. 3, Juillet 1979.
- [Bahnu 90] B. Bahnu, B. Roberts & J. Ming. *Inertial Navigation Sensor Integrated Motion Analysis for Obstacle Detection*. In IEEE International Conference on Robotics and Automation, Cincinnati, (USA), pages 954-959, Mai 1990.
- [Barr 81] A.H. Barr. *Superquadrics and Angle-Preserving Transformations*. IEEE Computer Graphics and Applications, Janvier 1981.



- [Barral 91] J.C. Barral, C. Barrouil, M. Llibre, R. Mampey & T. Fualdes. *Rapport Intermédiaire n. 1/7792.02 - DERA - Projet VAP Phase 4: Localisation et Navigation*. Rapport technique, CERT-DERA, Décembre 1991.
- [Bartels 87] R. H. Bartels, J. C. Beaty & B. A. Barsky. *B-splines*. In *Mathématiques et CAO*. Hermes, 1987.
- [Bauzil 91] G. Bauzil, R.Ferraz De Camargo, C.Lemaire & G.Vialaret. *Robot mobile HILARE II : description du matériel*. Rapport Interne 91234, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), Juillet 1991.
- [Bauzil 92] G. Bauzil, R.Ferraz De Camargo, C.Lemaire & G.Vialaret. *Robot mobile HILARE II. Matériel et bibliothèques associées*. Rapport Interne 92265, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), Juillet 1992.
- [Besl 86] Paul V. Besl & Ramesh C. Jain. *Invariant Surface Characteristics for 3D Object Recognition in Range Images*. *Computer Vision, Graphics and Image Processing*, vol. 33, no. 1, pages 33–80, Janvier 1986.
- [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, USA, Mai 1994.
- [Binford 71] T.O. Binford. *Visual perception by a computer*. In *IEEE Conference on Systems and Controls*, 1971.
- [Bohm 84] W. Bohm, G. Farin & J. Kahmann. *A Survey of Curve and Surface Methods in CAGD*. *Computed Aided Geometric Design*, vol. 1, pages 1–60, 1984.
- [Boissier 85] L. Boissier. *Modélisation de l'environnement et localisation du robot mobile HILARE par télémétrie laser*. Thèse de troisième cycle, Université Paul Sabatier, Décembre 1985.
- [Boissier 91] L. Boissier & G. Giralt. *Autonomous Planetary Rover (VAP): the robotics concepts*. In *I.A.R.P. 91 on Robotics in Space*, Pisa (Italy), Juin 1991.
- [Boissonnat 87] J.D. Boissonnat, O.D. Faugeras & E. Le Bras-Mehlan. *Représentation de données stéréo par la triangulation de Delaunay*. In *6ème Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle*, Antibes (France), pages 687–701, Novembre 1987.
- [Bolle 84] R. M. Bolle & D. B. Cooper. *Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pages 418–429, 1984.
- [Bolles 82] R.C. Bolles & R.A. Cain. *Recognising and locating partially visible objects : the local-feature-focus method*. *International Journal of Robotics Research*, vol. 1, no. 3, pages 57–82, 1982.
- [Boumaza 94] R. Boumaza & M. Devy. *Reconnaissance d'objets 3D non-polyédriques par un système multisensoriel Laser/Caméra*. In *AFCET 9ème Congrès Reconnaissance des Formes et Intelligence Artificielle*, Janvier 1994.
- [Bowyer 81] A. Bowyer. *Computing Dirichlet Tessellations*. *Computer Journal*, vol. 24, no. 2, pages 162–166, May 1981.
- [Brady 84] M. Brady, J. Ponce, A. Yuille & H. Asada. *Describing Surfaces*. In *Robotics Research : The Second International Symposium*, Kyoto (Japan), pages 434 – 445, Août 1984.

- [Brooks 79] R.A. Brooks, R. Greiner & T.O. Binford. *The ACRONYM model-based vision system*. In 6th International Joint Conference on Artificial Intelligence (IJCAI), Tokyo (Japan), pages 105–113, Août 1979.
- [Brooks 81] R.A. Brooks. *Symbolic Reasoning among 3D Models and 2D Images*. Artificial Intelligence, vol. 17, pages 285–348, 1981.
- [Bulata 92] H. Bulata. *Modélisation 3D de l'Environnement à l'aide de Primitives non-planes*. DEA, Université Paul Sabatier, Toulouse (France), Juin 1992.
- [Champleboux 92] G. Champleboux, S. Lavallée, P. Sautot & P. Cinquin. *Accurate calibration of cameras and range imaging sensors : the NPBS method*. In IEEE International Conference on Robotics and Automation, pages 1552–1557, Nice (France), Mai 1992.
- [Chatila 84] R. Chatila. *L'expérience du robot mobile HILARE*. Rapport Interne N. 3188, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), 1984.
- [Chen 88] H. Chen & T. Huang. *Maximal matching of 3-D points for multiple-object motion estimation*. In IEEE Conference on Computer Vision and Pattern Recognition, Ann Arbor, Michigan (USA), pages 75 – 90, Juin 1988.
- [Chen 91] Y. Chen & G. Medioni. *Object Modeling by Registration of Multiple Range Images*. In IEEE International Conference on Robotics and Automation, Sacramento, (USA), pages 2724 – 2729, Avril 1991.
- [Cohen 90] L.D. Cohen & I. Cohen. *A Finite Element Method Applied to New Active Contour Models and 3D Reconstruction from Cross Sections*. Rapport de recherche 1245, Institut National de Recherche en Informatique et en Automatique (I.N.R.I.A.), Rocquencourt (France), Juin 1990.
- [Cohen 91] I. Cohen, L. D. Cohen & N. Ayache. *Introducing Deformable Surfaces to Segment 3D Images and Infer Differential Structures*. Rapport de recherche 1403, Institut National de Recherche en Informatique et en Automatique (I.N.R.I.A.), Rocquencourt (France), Mars 1991.
- [Colly 91a] J. Colly. *Reconnaissance d'Objets Articulés Tridimensionnels à Partir d'Informations Multisensorielles pour des Applications Robotiques*. Thèse de l'Université Paul Sabatier, Toulouse (France), Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Octobre 1991.
- [Colly 91b] J. Colly & M. Devy. *Reconnaissance d'un Objet 3D par un Système Multisensoriel Laser/Caméra*. In 9ème congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, Novembre 1991.
- [Crowley 85] J.L. Crowley. *Navigation for an intelligent mobile robot*. IEEE Journal of Robotics and Automation, vol. 1, 1985.
- [Crowley 86] J.L. Crowley. *Mathematical Tools for Representing Uncertainty in Perception*. Rapport technique, L.I.F.I.A., Grenoble (France), Décembre 1986.
- [Crowley 90] J.L. Crowley, P. Bobet & K. Sarachik. *Dynamic World Modeling Using Vertical Lines Stereo*. In 1st European Conference on Computer Vision, Antibes (France), pages 241–246, Avril 1990.
- [Dacre Wright 93] B. Dacre Wright. *Planification de Trajectoires pour un Robot Mobile sur Terrain Accidenté*. Thèse de Doctorat de l'École Nationale Supérieure des Télécommunications. Rapport Interne N. 93510, L.A.A.S.-C.N.R.S., Octobre 1993.

- [Daily 88] M. Daily, J. Harris, D. Keirse, K. Olin, D. Payton, K. Reiser, J. Rosenblatt, D. Tseng & V. Wong. *Autonomous Cross-Country Navigation with the ALV*. In IEEE International Conference on Robotics and Automation, Philadelphia (USA), pages 718 – 726, Avril 1988.
- [De Boor 78] C. De Boor. *A Practical Guide to Splines*. Springer Verlag, 1978.
- [Del Pobil 92] A. P. Del Pobil, M. A. Serna & J. Llovet. *A New Representation for Collision Avoidance and Detection*. In IEEE International Conference on Robotics and Automation, Nice (France), pages 246–251, Mai 1992.
- [Demko 77] S. Demko. *Inverses of Band Matrices and Local Convergence of Spline Projections*. SIAM Journal of Numerical Analysis, vol. 14, no. 4, pages 616–619, Septembre 1977.
- [Devy 91] M. Devy, J. Colly, P. Grandjean & T. Baron. *Environment Modelling from a Laser / Camera Multisensor System*. In IARP 2nd Workshop on Multi-Sensor Fusion and Environment Modelling, Oxford (U.K.), Septembre 1991.
- [Devy 92a] M. Devy, J. Colly & R. Boumaza. *Localization of a satellite by a multisensory perceptual system*. International Symposium Artificial Intelligence Robotics and Automation, in Space, Toulouse (France), pp.117-128, Cepadues Editions, Octobre 1992.
- [Devy 92b] M. Devy, P. Fillatreau & M.C. Tatard. *Localization of an autonomous mobile robot from 3D depth images in a structured environment*. 23rd International Symposium on Industrial Robots (ISIR'92), Barcelona (Spain), pp.407-413, Octobre 1992.
- [Durrant-Whyte 87a] H.F. Durrant-Whyte. *Consistent Integration and Propagation of Disparate Sensor Observations*. In International Journal of Robotics Research, 1987.
- [Durrant-Whyte 87b] H.F. Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Robot Systems*. Kluwer Academic Publ., Boston (USA), 1987.
- [Elfes 86] A. Elfes. *A Sonar-Based Mapping and Navigation System*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), April 7-10 1986.
- [Elfes 89] A. Elfes. *Using Occupancy Grids for Mobile Robot Perception and Navigation*. In IEEE Transactions on Computers, 1989.
- [Faugeras 83] O.D. Faugeras & M. Hebert. *A 3-D recognition and positionning algorithm using geometrical matching between primitive surfaces*. In 8th International Joint Conference on Artificial Intelligence (IJCAI), Karlsruhe (RFA), pages 996–1002, 1983.
- [Faugeras 86] O.D. Faugeras, N. Ayache & B. Faverjon. *Building Visual Maps by combining noisy stereo measurements*. In IEEE International Conference on Robotics and Automation, San Francisco (USA), Avril 1986.
- [Faurre 71] P. Faurre. *Navigation Inertielle Optimale et Filtrage Statistique*. In Méthodes Mathématiques de l'Informatique No 1. Dunod, 1971.
- [Faux 79] I.D. Faux & M.J.Pratt. *Computational geometry for design and manufacture*. Ellis Horwood Limited, 1979.
- [Ferrand 91] A. Ferrand. *Conception et Mise en Oeuvre d'un Système de Capteurs Proprioceptifs Destiné à la Localisation Relative des Robots Mobiles*. Thèse de troisième cycle, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), Mai 1991.

- [Ferrand 92] A. Ferrand. *Mesure de l'inclinaison d'un robot mobile l'aide d'un odomètre associé des capteurs inertiels*. Rapport Interne 92440, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), Novembre 1992.
- [Fillatreau 91] P. Fillatreau & R. Prajoux. *Revue de Méthodes de Modélisation Adaptées à la Représentation de Formes 3D*. Rapport de Recherche N. 91468, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), Décembre 1991.
- [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), Juillet 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), Novembre 1993.
- [Fillatreau 93c] Ph. Fillatreau, F. Nashashibi, M. Devy & R. Prajoux. *Modélisation de Terrain Accidenté pour un Robot Mobile Autonome en Milieu Naturel*. In 4èmes Journées ORASIS. Pole Vision du GDR-PRC Communication Homme-Machine. Mulhouse (France), Octobre 1993.
- [Frappier 90] G. Frappier. *Systèmes Inertiels de Navigation pour Robots Mobiles*. In Séminaire *Les Robots Mobiles*, 1990.
- [Fua 91] P. Fua. *A parallel stereo algorithm that produces dense depth maps and preserves image features*. Rapport de Recherche 1369, Institut National de Recherche en Informatique et en Automatique (I.N.R.I.A.), Janvier 1991.
- [Gennery 89] D.B. Gennery. *Visual Terrain Matching for a Mars Rover*. In IEEE International Conference on Computer Vision and Pattern Recognition, San Diego (USA), pages 483 - 490, Juin 1989.
- [Giralt 92] G. Giralt & L. Boissier. *THE FRENCH PLANETARY ROVER VAP: Concept and Current Developments*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '92), Raleigh (USA), pages 1391-1398, Juillet 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, Juillet 1993.
- [Goldgof 88] D.B. Goldgof, T.S. Huang & H. Lee. *Feature Extraction and Terrain Matching*. In IEEE Conference on Computer Vision and Pattern Recognition, Ann Arbor (USA), pages 899 - 904, Juin 1988.
- [Grandjean 88] P. Grandjean, P. Moutarlier & A. Robert De Saint Vincent. *Calibration Odomètre - Platine Vision pour un Robot Mobile*. Rapport de recherche 89419, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), Décembre 1988.
- [Grandjean 89] P. Grandjean & A. Robert de Saint-Vincent. *3-D modeling of indoor scenes by fusion of noisy range and stereo data*. In IEEE International Conference on Robotics and Automation, Scottsdale (USA), pages 681-687, Mai 1989.
- [Grandjean 91] P. Grandjean. *Perception Multisensorielle et Interpretation de Scènes*. Thèse de l'Université Paul Sabatier, Toulouse (France), Rapport Interne N. 91305, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Octobre 1991.
- [Grosse 80] E. Grosse. *Tensor spline approximation*. In Linear Algebra and its Applications. 34,29-41, 1980.

- [Guéziec 92] A. Guéziec & N. Ayache. *Smoothing and Matching of 3-D Space Curves*. In 2nd European Conference on Computer Vision, Santa Margherita Ligure (Italy), Mai 1992.
- [Hamlin 92] G. J. Hamlin, R. B. Kelley & J. Tornerio. *Efficient Distance Calculation using the Spherically-Extended Polytope (S-tope) Model*. In IEEE International Conference on Robotics and Automation, Nice (France), pages 2502-2507, Mai 1992.
- [Hebert 89] 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, Mai 1989.
- [Hebert 92] M. Hebert. *3-D Landmark Recognition from Range Images*. In IEEE Conference on Computer Vision and Pattern Recognition, Champaign (USA), pages 360-365, Juin 1992.
- [Hermeline 82] F. Hermeline. *Triangulation automatique d'un Polyèdre de Dimension N*. RAIRO Numerical Analysis, vol. 16, no. 3, pages 211-242, 1982.
- [Herrb 89] M. Herrb, F. Noreils & P. Grandjean. *Coordination perception-action en robotique mobile : poursuite d'un objet par vision 3D*. In 7ème Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, Paris (France), pages 1343-1357, Décembre 1989.
- [Herrb 91] M. Herrb. *Vision en mouvement pour la robotique mobile*. Thèse de l'Université Paul Sabatier, Toulouse (France), Rapport Interne N. 91048, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Février 1991.
- [Horn 81] B.K.P. Horn & B. G. Shunk. *Determining optical flow*. Artificial Intelligence, vol. 17, pages 185-203, 1981.
- [Horn 91] B. Horn & J. Harris. *Rigid Body Motion from Range Image Sequences*. In CVGIP : Image Understanding, pages 1 - 13, Janvier 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 (France), Septembre 1991.
- [Hotz 93] B. Hotz, Z. Zhang & P. Fua. *Incremental Construction of Local DEM for an Autonomous Planetary Rover*. In INRIA-ESA Workshop on Computer Vision for Space Applications, Antibes (France), pages 33-43, Septembre 1993.
- [Ikeuchi 81] K. Ikeuchi. *Recognition of 3D objects using the Extended Gaussian Image*. In 7th International Joint Conference on Artificial Intelligence (IJCAI), Vancouver (Canada), pages 595-600, Août 1981.
- [Izaguirre 87] A. Izaguirre, J. Summers & P. Pu. *A New Development in Camera Calibration - Calibrating a Pair of Mobile TV Cameras*. In International Journal of Robotics Research, 1987.
- [Jaffard 90] S. Jaffard. *Propriétés des matrices "bien localisées" près de leur diagonale et quelques applications*. Annales de l'Institut Henri Poincaré. Série Analyse non linéaire, vol. 7, no. 5, pages 461-476, 1990.
- [Kamgar-Parsi 91] B. Kamgar-Parsi, J.L. Jones & A. Rosenfeld. *Registration of multiple overlapping range images: Scenes without distinctive features*. In IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 857-871, 1991.

- [Kass 87] M. Kass, A. Witkin & D. Terzopoulos. *Snakes: Active Contour Models*. In First International Conference on Computer Vision, London (U.K.), pages 259–268, Juin 1987.
- [Kehtarnavaz 89] N. Kehtarnavaz & S. Mohan. *A framework for estimation of motion parameters from range images*. In Computer Vision, Graphics, and Image Processing, pages 88 – 105, 1989.
- [Kemotsu 93] K. Kemotsu & T. Kanade. *Uncertainty in Object Pose Determination with Three Light-Stripe Range Measurements*. In IEEE International Conference on Robotics and Automation, Atlanta (USA), pages 128–134, Mai 1993.
- [Khoumsi 87] A. Khoumsi. *Un système de pilotage pour le contrôle du mouvement d'un robot mobile à travers des informations sensorielles*. Rapport de Recherche LAAS, Toulouse (France), 1987.
- [Khoumsi 88] A. Khoumsi. *Pilotage, asservissement sensoriel et localisation d'un robot mobile autonome*. Thèse de l'Université Paul Sabatier, Toulouse (France), Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Toulouse (France), Juin 1988.
- [Kishon 90] E. Kishon, T. Hastie & H. Wolfson. *3D Curve Matching Using Splines*. In 1st European Conference on Computer Vision, Antibes (France), Avril 1990.
- [Kishon 91] E. Kishon, T. Hastie & H. Wolfson. *3D Curve Matching Using Splines*. Journal of Robotic Systems, vol. 8, no. 6, pages 723–743, 1991.
- [Kweon 90] I. Kweon & T. Kanade. *High Resolution Terrain Map from Multiple Data*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '90), Tsuchiura (Japan), Juillet 1990.
- [Kweon 91a] I. Kweon, R. Hoffman & E. Krotkov. *Experimental Characterization of the Perceptron Laser Rangefinder*. Rapport technique CMU-RI-TR-91-1, Robotics Institute, Carnegie Mellon University, Janvier 1991.
- [Kweon 91b] I. S. Kweon. *Modeling Rugged Terrain by Mobile Robots with Multiple Sensors*. PhD thesis, Robotics Institute, Carnegie Mellon University, Février 1991.
- [Lacombe 91] J. L. Lacombe & T. Blais. *PROGRAMME AMR DE ROBOTS MOBILES AVANCES. 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 (ORIA), Marseille (France), pages 223–230. Institut International de Robotique et d'Intelligence Artificielle de Marseille, Décembre 1991.
- [Lacroix 93] S. Lacroix, P. Fillatreau, F. Nashashibi, R. Chatila & M. Devy. *Perception for Autonomous Navigation in a Natural Environment*. In Workshop on Computer Vision for Space Applications, Antibes (France), Septembre 1993.
- [Lebegue 92] X. Lebegue & J. K. Aggarwal. *Extraction and Interpretation of Semantically Significant Line Segments for a Mobile Robot*. In IEEE International Conference on Robotics and Automation, Nice (France), Mai 1992.
- [Leonard 90] J. Leonard, H. Durrant-Whyte & I. J. Cox. *Dynamic Map Building for an Autonomous Mobile Robot*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '90), Tsuchiura (Japan), pages 89–96, Juillet 1990.
- [Liang 90] P. Liang & J.S. Todhunter. *Representation and recognition of surface shapes in range images: a differential geometry approach*. In Computer Vision, Graphics, and Image Processing, pages 78–109, 1990.

- [Manaoui 88] F.O. Manaoui. *Etude et Simulation d'Algorithmes de Navigation pour Robots Mobiles Autonomes sur Terrain inégal*. Thèse, de l'Université des Sciences et Techniques du Languedoc, Montpellier (France), 1988.
- [Matthies 88] L. 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), pages 727 – 733, Avril 1988.
- [Milios 92] E. E. Milios & T. J. Horton. *A Combined Extended Circular Image and Spatial Occupancy Approach to 2-D Contour Matching*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '92), Raleigh (USA), Juillet 1992.
- [Mohr 88] R. Mohr. *Sur l'appariement modèle - perception*. In 2ème Atelier Scientifique : "Traitement d'Images : du Pixel à l'Interprétation" (TIPI), Aussois (France), pages XXIX-1 – XXIX-17. C.N.R.S., Avril 1988.
- [Morlans 92] R. Morlans. *Génération des Trajectoires d'un Robot d'Exploration Planétaire utilisant un Modèle Numérique de Terrain issu d'Images prises par un Satellite*. Thèse de l'Université des Sciences et Techniques du Languedoc, Montpellier (France), Juin 1992.
- [Mortenson 85] M.E. Mortenson. *Geometric Modeling*. John Wiley & Sons, 1985.
- [Moutarlier 89a] P. Moutarlier & R. G. Chatila. *An Experimental System for Incremental Environment Modelling by an Autonomous Mobile Robot*. In International Symposium on Experimental Robotics, Montreal (Canada), Juin 1989.
- [Moutarlier 89b] P. Moutarlier & R. G. Chatila. *Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modelling*. In International Symposium on Robotics Research, Tokyo (Japan), 1989.
- [Moutarlier 89c] P. Moutarlier, P. Grandjean & R. Chatila. *Multisensory Data Fusion for Mobile Robot Location and 3D Modeling*. In IARP 1st Workshop on Multi-Sensor Fusion and Environment Modeling, Toulouse (France), Octobre 1989.
- [Moutarlier 91] P. Moutarlier. *Modélisation Autonome de l' Environnement par un Robot Mobile*. Thèse de l'Université Paul Sabatier, Toulouse (France) Rapport Interne N. 91381, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Octobre 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, Mai 1992.
- [Nashashibi 93a] 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), Rapport Interne N. 93032, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Février 1993.
- [Nashashibi 93b] F. Nashashibi & M. Devy. *3D Incremental Modeling and Robot Localization in a Structured Environment using a Laser Range Finder*. In IEEE Transactions on Robotics and Automation, Atlanta (USA), Mai 1993.
- [Nashashibi 93c] F. Nashashibi & M. Devy. *Combining Terrain Maps and Polyhedral Models for Robot Navigation*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '93), Yokohama (Japan), Juillet 1993.
- [Nashashibi 94] F. Nashashibi, Ph. Fillatreau, B. Dacre-Wright & T. SimÉOn. *3D Autonomous Navigation in a Natural Environment*. In IEEE International Conference on Robotics and Automation, San Diego (USA), Mai 1994.

- [Noreils 89] F. Noreils. *Contrôle d'Exécution de Plans d'Actions et Architecture pour Robots Mobiles*. Thèse de l'Université Paul Sabatier, Toulouse (France) 560, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Novembre 1989.
- [Pampagnin 90] L.H. Pampagnin. *Reconnaissance d'objets tridimensionnels en perception monoculaire et multisensorielle — Application à la robotique spatiale —*. Thèse de l'Université Paul Sabatier, Toulouse (France), Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), Octobre 1990.
- [Parvin 91] B. Parvin & G. Medioni. *A Layered Network for the Correspondance of 3D Objects*. In IEEE International Conference on Robotics and Automation, Sacramento, (USA), Avril 1991.
- [Parvin 92] B. Parvin & G. Medioni. *B-rep from Unregistered Multiple Range Images*. In IEEE International Conference on Robotics and Automation, Nice (France), Mai 1992.
- [Paul 81] R.P. Paul. *Robot manipulators: mathematics, programming and control*. MIT Press, 1981.
- [Pavlidis 81] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer science press, 1981.
- [Pouzyreff 91] S. Pouzyreff. *Description de données de télémétrie laser à l'aide de primitives volumiques*. Rapport de D.E.A. d'informatique, Université Paul Sabatier de Toulouse (France), 1991.
- [Puskorius 87] G. Puskorius & L. Feldcamp. *Global Calibration of a Robot / Vision System*. In IEEE International Conference on Robotics and Automation, Raleigh (USA), pages 190 – 195, Avril 1987.
- [Radack 89] G.M. Radack & N.I. Badler. *Local matching of surfaces using a boundary-centered radial decomposition*. In CVGIP : Image Understanding, pages 380 – 396, 1989.
- [Radix 72] J. C. Radix. *Techniques inertielles*. Masson, 1972.
- [Radix 78] J. C. Radix. *Gyromètres et gyroscopes*. Cepadues Editions, 1978.
- [Radix 80] J. C. Radix. *Systèmes inertiels à composants liés 'strap-down'*. Cepadues Editions, 1980.
- [Rao 89] B. Rao & H. Durrant-White. *A fully decentralized camera-data fusion algorithm for tracking*. In IARP 1st Workshop on Multi-Sensor Fusion and Environment Modeling, Toulouse (France), Octobre 1989.
- [Risler 91] J. J. Risler. *Méthodes Mathématiques pour la CAO*. In Recherche en Mathématiques Appliquées No 18. Masson, 1991.
- [Robert 91] L. Robert & O.D. Faugeras. *Curved-Based Stereo: Figural Continuity And Curvature*. In IEEE International Conference on Computer Vision and Pattern Recognition, Hawai, 1991.
- [Rodriguez 89] J.J. Rodriguez & J.K. Aggarwal. *Navigation using image sequence analysis and 3-D terrain matching*. In Workshop on Interpretation of 3-D scenes, pages 200 – 207, Novembre 1989.
- [Samson 94] J.F. Samson & O. Mercier. *Locomotion Aid to the Teleoperation of an All Terrain Mobile Robot with a 3D Laser Range Finder*. Submitted to RI/SME Fifth World Conference on Robotics Research, 1994.



- [Schalit 92] E. Schalit. *ARCANE: Towards Autonomous Navigation on rough terrains*. In IEEE International Conference on Robotics and Automation, Nice, (France), pages 2568 – 2574, Mai 1992.
- [Shiller 91] Z. Shiller & Y. R. Gwo. *Dynamic Motion Planning of Autonomous Vehicles*. IEEE Transactions on Robotics and Automation, vol. 7, no. 2, Avril 1991.
- [Shirai 87] Y. Shirai. *Three dimensional computer vision*. Springer-Verlag, Berlin, 1987.
- [Shiu 87] Y. C. Shiu & S. Ahmad. *Finding the mounting position of a sensor by solving a homogeneous transform equation of the form  $AX=XB$* . In IEEE International Conference on Robotics and Automation, pages 1666–1671, Raleigh (USA), Avril 1987.
- [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), Yokohama (Japan), Juillet 1993.
- [Smith 87] R.C. Smith & P. Cheeseman. *On representation and estimation of spatial uncertainty*. International Journal of Robotics Research, vol. 5, no. 4, Winter 1987.
- [Soucy 92] M. Soucy, A. Croteau & D. Laurendeau. *A Multi-Resolution Surface Model for Compact Representation of Range Images*. In IEEE International Conference on Robotics and Automation, Nice (France), pages 1701–1706, Mai 1992.
- [Stein 92] F. Stein & G. Medioni. *Map-Based Localization using the Panoramic Horizon*. In IEEE International Conference on Robotics and Automation, Nice (France), pages 2631 – 2637, Mai 1992.
- [Szeliski 88] R. Szeliski. *Estimating Motion from Sparse Range Data without Correspondance*. In Second International Conference on Computer Vision, Tampa, Florida (USA), pages 207 – 216, Décembre 1988.
- [Talluri 90] R. Talluri & J.K. Aggarwal. *Position Estimation for a Mobile Robot in an Unstructured Environment*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '90), pages 159–166, Tsuchiura (Japan), Juillet 1990.
- [Taniguchi 91] Y. Taniguchi, M. Asada & Y. Shirai. *World Model Representation Based on Planar Patch from Range and Video Images*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '91), Osaka (Japan), pages 409 – 414, Novembre 1991.
- [Thorpe 89] C. Thorpe & T. Kanade. *1988 Year End Report for Road Following at Carnegie Mellon*. Rapport technique CMU-RI-TR-89-5, Robotics Institute, Carnegie Mellon University, Pittsburgh (USA), Février 1989.
- [Tornerio 91] J. Tornerio, J. Hamlin & R.B. Kelley. *Spherical-Object Representation and Fast Distance Computation for Robotics Applications*. In IEEE International Conference on Robotics and Automation, Sacramento (USA), Avril 1991.
- [Tsai 87] R.Y. Tsai & R.K. Lenz. *A New Technique for Autonomous and Efficient 3D Robotics Hand/Eye Calibration*. Research Report RC13212, IBM, Septembre 1987.
- [Vaganay 93] J. Vaganay, M.J. Aldon & A. Fournier. *Mobile Robot Attitude Estimation by Fusion of Inertiel Data*. In IEEE International Conference on Robotics and Automation, Atlanta (USA), pages 277–282, Mai 1993.
- [Wang 92a] C. C. Wang. *Extrinsic calibration of a vision sensor mounted on a robot*. In IEEE trans. on Robotics and Automation, pages 161–175, Avril 1992.

- [Wang 92b] Y.F. Wang, Z. Yang & J.F. Lee. *On 3D Model Construction by Fusing Heterogeneous Sensor Data*. In IEEE International Workshop on Intelligent Robots and Systems (IROS '92), Raleigh (USA), Juillet 1992.
- [Watson 81] D.F. Watson. *Computing the N-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes*. Computer Journal, vol. 24, no. 2, pages 167-172, May 1981.
- [Williams 92] D. J. Williams & M. Shah. *A Fast Algorithm for Active Contours and Curvature Estimation*. CVGIP : Image Understanding, vol. 55, no. 1, pages 14-26, Janvier 1992.
- [Zhang 91] Z. Zhang & O. D. Faugeras. *Three-Dimensional Motion Computation and Object Segmentation in a Long Sequence of Stereo Frames*. Rapport de recherche 1438, Institut National de Recherche en Informatique et en Automatique (I.N.R.I.A.), Sophia Antipolis (France), Juillet 1991.

## Thèse de Philippe Fillatreau

\* \* \*

### Localisation et Modélisation Tridimensionnelles pour un Robot Mobile Autonome Tout Terrain

#### Résumé

Cette thèse traite du problème de la localisation dans un environnement tridimensionnel (3D) d'un robot mobile autonome, ainsi que de celui de la modélisation de terrain. Deux exemples de domaines d'application de ce travail sont la robotique mobile d'intervention (sécurité civile, ...) et l'exploration planétaire. La contribution de la thèse se situe surtout au niveau de la localisation du robot dans un environnement semi-structuré ou non structuré.

Un état de l'art concernant d'une part les principaux capteurs utilisables, et d'autre part la modélisation, est présenté. L'accent est mis sur les capteurs inertiels et sur la modélisation de formes non structurées; le choix des capteurs utilisés, ainsi que celui des primitives géométriques retenues, sont justifiés.

Dans un premier temps, une approche pour la localisation du robot à l'aide d'amers de type structuré, mais éventuellement naturels, est présentée. Le problème de la fusion incrémentale d'un modèle de localisation basé sur des primitives hétérogènes, à partir de la détection de lignes verticales (arbres, colonnes) ou de murs, est traité. Plusieurs approches pour le recalage de la position du robot et la fusion des données hétérogènes sont comparées. Une stratégie de choix des différents amers est enfin proposée.

Dans un deuxième temps, le problème de la modélisation de terrain accidenté et du recalage sur des caractéristiques non structurées de l'environnement est abordé. Une méthode de modélisation du terrain par hiérarchie de B-splines est proposée, et la construction incrémentale du modèle de terrain est traitée. Le modèle analytique obtenu permet d'extraire divers invariants 3D, comme des maxima d'altitude ou des points de forte courbure; l'aspect multi-résolution permet de focaliser progressivement la recherche de telles caractéristiques. Finalement, une méthode de localisation faisant coopérer l'extraction de caractéristiques avec des techniques de type corrélation est proposée.

Les différentes méthodes développées ont fait l'objet d'une validation sur plusieurs robots mobiles expérimentaux.

**Mots clés :** *Perception, Robots Mobiles, Localisation, Modélisation 3D, Modèles Hétérogènes, Surfaces B-splines, Corrélation, Fusion 3D.*

### 3D Localization and Modelling for an Autonomous All Terrain Mobile Robot

#### Abstract

This thesis deals with the problem of localization in a three-dimensional (3D) environment for an autonomous mobile robot, as well as terrain modelling. Two examples of applications are intervention mobile robotics (public safety, ...) and planetary exploration. The contribution of the thesis concerns mainly the localization of the robot in a mid-structured or unstructured environment.

The state of the art, related to, on one hand, the main useable sensors, and, on the other hand, modelling, is surveyed. Inertial sensors and the modelling of unstructured shapes, are emphasized; the choice for the sensors involved, and for the geometric primitives used, is justified.

First, an approach for localizing the robot by using structured (but possibly natural) landmarks, is presented. The problem of the incremental fusion of a localization model based on heterogeneous primitives, corresponding to the detection of vertical lines (columns, trees) or walls, is tackled. Several approaches for the update of the position of the robot and for the fusion of heterogeneous features, are compared. A strategy for choosing the landmarks is finally proposed.

Second, the problem of rugged terrain modelling, and updating the robot position by using unstructured environment features, is dealt with. A method for terrain modelling, involving a hierarchy of B-spline surfaces, is proposed, and incremental terrain modelling is tackled. The analytic model obtained allows to extract various 3D invariants, like elevation maxima or high-curvature points; the multi-resolution aspect allows to progressively focus the search for such features. Finally, a localization method involving a cooperation between feature and iconic (correlation-like) matching is proposed.

The different proposed methods have been validated on several experimental mobile robots.

**Keywords :** *Perception, Mobile Robots, Localization, 3D Modeling, Heterogeneous Models, B-spline Surfaces, Correlation, 3D Fusion.*