POLYPUBLIE

Système d'arpentage automatisé pour les mines souterraines

Philippe Debanné

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	2000
Type	Mémoire ou thèse / Dissertation or Thesis Mémoire de maîtrise/Master's thesis
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/8824/
Conditions d'utilisation Terms of use	© Tous droits réservés / All rights reserved
Référence Citation	Debanné, P. (2000). Système d'arpentage automatisé pour les mines souterraines [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8824/

Détails de la thèse
Thesis details

Directeurs de recherche Advisors	Paul Cohen, & Jean-Yves Hervé
Programme Program	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

Système d'arpentage automatisé pour les mines souterraines

PHILIPPE DEBANNÉ
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE)
MARS 2000

© Philippe Debanné, 2000.

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-53570-3

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

Système d'arpentage automatisé pour les mines souterraines

présenté par: DEBANNÉ Philippe,

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. GAMACHE Michel, Ph.D., président

M. COHEN Paul, Ph.D., membre et directeur de recherche

M. HERVÉ Jean-Yves, Ph.D., membre et codirecteur de recherche

M. LAGANIÈRE Robert, Ph.D., membre

À mon père.

Remerciements

Tout d'abord, je voudrais remercier M. Paul Cohen de m'avoir accueilli dans le Groupe de Recherche en Perception et Robotique et de m'avoir encadré tout au long de ma maîtrise. Je remercie également M. Jean-Yves Hervé, qui a lui-aussi suivi mon travail et contribué à l'élaboration de ce projet. Je remercie aussi les membres du jury pour le temps qu'ils auront consacré à l'évaluation de ce travail.

De plus, je suis reconnaissant envers le Fonds FCAR pour la bourse de recherche qui m'a été attribuée au début de mes études, et envers la chaire CRSNG-Noranda pour l'appui financier qui m'a été accordé par la suite.

J'aimerais exprimer ma reconnaissance envers tous les membres (passé et présent) du GRPR pour leur aide, leurs encouragements et leur amitié. En particulier, je tiens à remercier Geneviève Girard, qui a toujours donné un visage rassurant aux maintes procédures administratives requises. Je lève mon chapeau à deux "vétérans" du laboratoire : Vladimir Polotski, qui a donné généreusement de son temps pour m'aider dans mon cheminement et pour parler de divers sujets (le camping!) : et Hai Nguyen, qui s'est occupé de mes requêtes techniques avec patience. Merci aussi à Frédéric Labrosse pour sa collaboration à mon projet et pour les nombreuses questions en informatique auxquelles il a répondu.

Enfin, je témoigne ma reconnaissance envers ma famille proche, qui m'ont toujours encouragé dans les moments d'épreuve. Malgré les délais et les périodes difficiles qui l'ont marquée, je retire de cette maîtrise des connaissances élargies et des leçons importantes. Je remercie Dieu pour en être venu à bout.

Résumé

Parmi les projets existants en robotique mobile destinés aux mines souterraines, aucun des travaux n'est adapté à la tâche d'arpenter les excavations dans une mine. Ainsi, dans le contexte de la recherche en automatisation minière à la chaire CRSNG-Noranda, nous étudions la faisabilité de l'arpentage d'une mine souterraine par une plate-forme mobile autonome (robot). L'objectif de ce projet est de démontrer le concept d'un système adapté à la fois à l'arpentage automatisé et à la navigation autonome et qui emploie un robot capable de naviguer sans l'aide de structures de guidage. Il s'agit d'un véhicule muni d'un senseur odométrique, de télémètres latéraux fournissant des mesures denses de la forme des murs et du toit et d'un télémètre frontal balayant un demi-plan horizontal devant le véhicule. Notre travail consiste à développer les *capacités d'analyse sensorielle* de ce "robot-arpenteur". La réalisation physique du système est au-delà de la portée de ce projet ; par conséquent, nous évaluons nos stratégies d'analyse sensorielle à l'aide d'un banc expérimental en *mine virtuelle* simulée sur ordinateur avec les méthodes de l'infographie.

Nous nous basons sur un projet de recherche connexe, portant sur la modélisation générique d'objets, pour établir un *modèle* de mine souterraine : celle-ci est décomposable en sous-régions qualitativement distinctes que nous appelons ses *éléments topologiques*. Nous en considérons deux types : les *galeries* et les *intersections* de galeries. (Le modèle peut aussi inclure des *chambres*, qui représentent des espaces ouverts, par exemple des zones d'extraction.) Nous faisons de plus une supposition importante : que le robot possède *a priori* un *graphe topologique* décrivant la région

de la mine à parcourir. Dans ce graphe, les arcs symbolisent les galeries et les nœuds symbolisent les intersections (et les chambres, le cas échéant). En outre, vu la grande complexité du problème de départ, nous faisons quelques simplifications. D'une part, nous développons notre approche dans l'optique d'arpenter des zones de transport dans la mine plutôt que des zones à excavations denses (*e.g.* les zones de chambres et piliers). D'autre part, nous faisons abstraction des problèmes de bruit de mesure dans l'odométrie et dans les senseurs télémétriques.

Le développement de notre système s'appuie sur une architecture hiérarchique qui sépare la tâche totale en trois *activités principales* menées en parallèle : la navigation globale, la navigation locale et l'arpentage. La navigation globale est responsable de localiser le robot par rapport au graphe topologique; ici, le problème classique de distinguer globalement tous les lieux rencontrés est remplacé par la tâche moins complexe de reconnaître les *transitions* entre les éléments topologiques. La navigation locale s'occupe de contrôler les déplacements du robot en exploitant l'information sensorielle plutôt qu'une trajectoire précise pré-établie. L'arpentage doit récolter les mesures provenant des télémètres latéraux dans un format exploitable pour la construction d'un modèle géométrique de la mine. Notre architecture se veut complète bien que certaines de ses composantes ne soient pas implantées. Notamment, comme le contrôle local de véhicule basé sur les senseurs a été étudié dans le milieu minier souterrain, nous laissons de côté cet aspect du système au profit des aspects de l'arpentage et de la navigation globale.

L'activité d'arpentage est divisée en trois modules. Le module collecteur de mesures produit et maintient à jour un ensemble de *cartes métriques locales* des différents éléments topologiques traversés. Nous appelons *jonction* l'union des deux morceaux de parois qui décrivent l'endroit où une galerie et une intersection, adjacentes dans le graphe topologique, se rattachent physiquement dans la mine. La carte métrique locale est constituée des parois mesurées, sous forme de listes de points (coordonnées et hauteur), et de référentiels cartésiens rattachés aux jonctions débouchant

dans l'élément topologique. Le module relocalisateur métrique intervient lorsque le robot rentre de nouveau dans une galerie ou une intersection déjà traversée ; ce module détermine la pose du robot dans la carte métrique existante afin d'y fusionner les nouvelles mesures. La relocalisation exploite les *référentiel locaux* (RLs) détectés dans les jonctions (par le troisième module), ainsi que les mesures de la forme des parois. Ces deux premiers modules de l'arpentage ne sont pas évalués expérimentalement.

Le troisième module est le détecteur de RLs ; il s'occupe de localiser précisément les RLs à l'intérieur des jonctions. Idéalement, nous voulons extraire de la forme des parois de jonction une caractéristique géométrique qui soit invariante face à la répartition des points mesurés et à la rugosité des parois. Nous modélisons d'abord les deux parois par des courbes polynomiales : cette étape réduit les effets des deux facteurs mentionnés et de l'occlusion partielle de parois. Ensuite, le RL est déterminé par la minimisation sous contraintes de la distance entre les deux courbes. Une série de tests dans plusieurs cas de traversée d'intersection, en faisant varier pour chaque cas la trajectoire précise suivie, démontrent que le RL est une caractéristique significative et géométriquement stable de la structure de l'environnement aux transitions entre éléments topologiques.

L'activité de navigation globale est divisée en deux modules : le navigateur topologique et le détecteur de transition. Le navigateur topologique maintient un pointeur vers la position courante du robot dans le graphe topologique. À chaque transition qui lui est signalée par le détecteur de transition, il met à jour les étiquettes (provenant du graphe) de l'élément topologique actuel et des éléments adjacents. Normalement, il faudrait aussi que ce module s'occupe de la replanification de chemin et la correction d'erreurs dans le graphe lorsque surviennent des événements perceptuels inattendus. Ce sont là des problèmes relativement complexes et pour limiter l'envergure de ce projet nous laissons ce module à l'état conceptuel.

Le détecteur de transition alterne entre l'analyse des mesures frontales et celle des mesures latérales afin de notifier puis confirmer aux autres modules les changements

d'éléments topologiques. Afin d'assurer la fiabilité de ce processus, ce module exploite la redondance temporelle des mesures sensorielles. L'analyse frontale détecte les *branches* (c'est-à-dire les occlusions et les régions à la limite de visibilité) dans chaque profil frontal instantané, pour laisser accumuler les profils traités et en former une *image binaire*. Celle-ci est composée de régions blanches séparées par la *trace* des profils qui indique (en noir) les branches détectées. L'évolution de la trace des profils à mesure que le robot se déplace est marquée par des événements importants. Le module reconnaît l'arrivée d'une intersection devant le robot par l'augmentation du nombre de régions blanches dans l'image. Inversement, lorsque le robot sort de l'intersection, ce nombre redescend à une valeur stable.

L'analyse des mesures latérales réutilise la notion de RL de manière dynamique afin de confirmer la présence dans la structure du milieu parcouru des transitions signalées auparavant par l'analyse frontale. Plus précisément, il s'agit de faire la détection de RLs *périodiquement* sur une section de parois qui suit le véhicule, jusqu'à ce que les RLs atteignent une position stable derrière le véhicule. Nos tests sur la détection des transitions démontrent la validité des événements perceptuels que nous avons définis.

Ce travail apporte plusieurs contributions innovatrices aux domaines de l'automatisation minière et de la robotique mobile. D'une part, notre système se distingue par la modélisation topologique explicite de la mine. Le modèle d'environnement et le graphe topologique *a priori* nous permettent de formuler de manière originale les problèmes de la localisation globale, de la relocalisation métrique et de l'arpentage de la mine. D'autre part, ce projet introduit des idées originales quant à son développement des capacités sensorielles du "robot-arpenteur". Le concept des RLs, qui caractérisent les changements importants dans la structure locale de la mine, est certainement nouveau par rapport à ce qui existe dans la littérature. L'image binaire des profils frontaux est elle aussi une représentation originale de l'environnement local mesuré. Cette représentation riche et compacte du milieu perçu pourrait éventuellement trouver d'autres applications, par exemple dans le contexte du contrôle local de véhicule.

Abstract

Within the existing body of work in mobile robotics for underground mines, none of the projects are adapted to the task of surveying mine excavations. Thus, in the context of mining automation research at the NSERC-Noranda chair, we are studying the feasibility of underground mine surveying performed by an autonomous mobile platform (robot). The objective of this project is to demonstrate the concept of a system which is adapted both to automated surveying and to autonomous navigation, and which uses a robot capable of navigating without guiding structures. The robot consists of a vehicle equipped with an odometry sensor, lateral range sensors providing dense shape measurements of the drift walls and roof, and a frontal range sensor scanning a horizontal half-plane in front of the vehicle. Our work focuses on developing the *sensory analysis capabilities* of this "surveyor-robot". The physical realization of the system is beyond the scope of this project; therefore, we evaluate our sensory analysis strategies by means of a *virtual mine* simulation test-bed designed with the use of computer graphics tools.

On the basis of a related research project dealing with generic object modeling, we have established an *underground mine model*: the mine can be divided into qualitatively distinct sub-regions which we call its *topological regions*. Of these, we consider two kinds: the *drifts* and the drifts' *intersections*. (The mine model can also include *rooms*, which represent open spaces such as stopes or equipment storage areas). At this point, we make an important assumption: that the robot possesses initial knowledge of the region of the mine to be explored, in the form of a *topological graph*. In

this *a priori* graph, edges symbolize drifts and nodes symbolize intersections (and optionally, rooms). Furthermore, given that our initial problem is quite complex, we make certain simplifications. Firstly, we develop our approach with the goal of surveying mine regions such as ore haulage zones rather than regions where excavations are more tightly spaced, *e.g.* room and pillar zones. Secondly, we disregard those problems relating to measurement noise in the odometry and the range sensors.

The development of our system rests upon a hierarchical architecture that separates the entire task into three concurrent *activities*: global navigation, local navigation and surveying. Global navigation is responsible for localizing the robot with respect to the topological graph; here, the common exploration problem of mutually distinguishing all the places encountered is substituted for the simpler task of recognizing the *transitions* between topological regions. Local navigation deals with controlling the robot's movements by using sensory information rather than a preplanned, geometrically precise path. Surveying must gather the measurements coming from the lateral range sensors in a format that is suitable for constructing a geometrical model of the mine. Our architecture is meant to be complete, though some of its components are not wholly implemented. In particular, since sensor-based control of a vehicle in the underground mine environment has been studied, we leave aside this aspect of the system and develop instead the surveying and global navigation aspects.

The surveying activity is divided into three modules. The measurement collector module produces and maintains a set of *local metrical maps* of the traversed topological regions. The term *junction* refers to two wall sections describing the place at which a drift and an intersection, being adjacent in the topological graph, are joined physically in the mine. A local metrical map comprises the measured walls as lists of points (coordinates and height), and local coordinate systems linked with the junctions leading into the topological region. The metrical relocalization module comes into play when the robot re-enters a drift or intersection that has already been

traversed; this module determines the robot's pose in the existing metrical map so as to allow registration of the new surveying data. Relocalization makes use of *local reference frames* (LRFs) detected within the junctions (by the third module), as well as the measured wall points. These two first modules of the surveying activity are not implemented nor tested.

The third module is the LRF detector; it must precisely locate the LRFs within the junctions. Ideally, we want to extract from the shape of the junction walls a geometrical feature that is invariant to the distribution of measured points and to surface roughness. We first model the two walls by fitting polynomial curves; this step reduces the effects of the two factors just mentioned and of partial wall occlusions. The LRF is then calculated by a constrained minimization of the distance between the two curves. A series of tests in several intersection traversal cases, with the vehicle's trajectory varying in each case, show that the LRF is a relevant and geometrically stable feature of the environment's structure at the transitions between topological regions.

The global navigation activity is divided into two modules: the topological navigator and the transition detector. The topological navigator maintains a pointer to the current position of the robot in the graph. Each time that a transition is signaled by the transition detector, the navigator updates the labels (from the graph) of the current and adjacent topological regions. Ordinarily, this module would also have to perform path re-planning and correction of errors in the graph whenever the robot makes unexpected observations. However, these relatively complex problems are beyond the scope of this project, and thus this module remains at the conceptual stage.

The transition detector alternates between the analysis of the frontal and lateral range data in order to notify and then confirm to the other modules that the robot is changing topological regions. In order that the recognition strategy be reliable, we exploit temporal redundancy in the available sensory information. To analyze the

frontal data, we detect the *branches* (meaning occlusions and portions beyond the visible range) in the instantaneous range profiles, and then accumulate the resulting data over time so as to create a *binary image*. This image is composed of white regions separated by a black *trace* indicating the branches detected in the profiles. As the robot advances, significant events arise in the evolution of the trace in the binary image. Namely, the module recognizes that the robot is approaching an intersection by an increase in the number of white regions in the image. Conversely, this number decreases to a stable value as the robot leaves the intersection.

To analyze the lateral data, we re-use the concept of the LRF in a dynamic manner, with the purpose of confirming that the currently measured wall shape indicates the topological change previously observed in the "look-ahead" data from the frontal sensor. Specifically, LRF detection is performed periodically on the (gradually lengthening) junction walls, until this feature has reached a stable location behind the vehicle. Our testing of the transition detector module (again in several intersection traversals with a variable trajectory) demonstrate the validity of the perceptual events which have been defined.

This work contributes several innovations to the fields of mining automation and mobile robotics. One one hand, our system is distinguishable by its use of an explicit topological model of the mine. The underground mine model and the *a priori* graph allow us to formulate the problems of global localization, metrical relocalization and mine surveying in an original manner. On the other hand, this project introduces original ideas by its development of the "surveyor-robot's" sensory analysis capabilities. The concept of the LRFs, which characterize important, naturally occurring changes in the mine's local structure, is certainly innovative compared to the existing literature. The binary image of the frontal range profiles is also an original means of representing the locally measured environment. This rich and compact representation of the perceived environment could conceivably have other applications, for example in the context of local vehicle control.

Table des matières

Dédicace iv

Remerciements v

Résumé vi

Abstract x

Table des matières xiv

Liste des tableaux xix

Liste des figures xxi

Liste des annexes xxv

1 Introduction 1

2 Revue bibliographique 10

2.1 Exploration d'environnement 10

2.1.1 Quelques définitions 10

2.1.2 Représentation géométrique du milieu 12

2.1.3 Représentation du milieu par grille d'occupation 15

2.1.4 Représentation topologique du milieu 18

2.2 Navigation autonome en mines souterraines 24

2.2.1 Automatisation d'une chargeuse-navette 24
2.2.2 Automatisation d'un mineur en continu 27
2.2.3 Autres plate-formes mobiles pour le milieu souterrain 28
2.3 Techniques d'arpentage de mines 29
2.3.1 Appareils de mesure 29
2.3.2 La mesure des positions horizontales 30
2.3.3 La mesure des hauteurs 32
2.3.4 Mesures latérales le long du parcours 34
2.3.5 Sources d'erreurs et leur traitement 34
2.4 Situation de notre projet 36

3 Définition du problème 38
3.1 Objectifs du système 38
3.1.1 Arpentage 38
3.1.2 Navigation autonome 39
3.2 Modèle géométrique de la mine 41
3.2.1 Modèle de mine pour la navigation 41
3.2.2 Milieu minier simulé 42
3.3 Modalités sensorielles du robot 43
3.3.1 Comparaison des principes sensoriels de télémétrie 43
3.3.2 Contraintes de l'arpentage des parois 48
3.3.3 Contraintes de la navigation globale 50
3.3.4 Contraintes de la navigation locale 56
3.3.5 Configuration sensorielle proposée 59
3.4 Présentation du système 60
3.4.1 Activité de navigation locale 60
3.4.2 Activité de navigation globale 62
3.4.3 Activité d'arpentage 63
3.4.4 Erreurs de positionnement métrique 66

3.4.5 Suite du mémoire 70

4 Activité d'arpentage 72

4.1 Introduction 72

4.2 Conception du collecteur de mesures 72

4.2.1 Décomposition du module en fonctions 72

4.2.2 Limite du développement 78

4.3 Conception du relocalisateur métrique 79

4.3.1 Méthodes de localisation métrique dans la littérature 80

4.3.2 Considérations dans notre cas 84

4.3.3 Décomposition du module en fonctions 85

4.3.4 Limite du développement 88

4.4 Détecteur de référentiels locaux : conception 89

4.4.1 Conventions typographiques dans la notation 89

4.4.2 Critères importants 90

4.4.3 Notion importante : le segment minimal 94

4.4.4 Décomposition du module en fonctions 99

4.5 Détecteur de référentiels locaux : implantation 104

4.5.1 Arpentage de la jonction 104

4.5.2 Segmentation des parois 106

4.5.3 Estimation de largeur de galerie 109

4.5.4 Calcul des repères propres 110

4.5.5 Modélisation des parois 112

4.5.6 Recherche du segment minimal 114

4.5.7 Positionnement du référentiel local 120

4.6 Détection des référentiels locaux : résultats 120

4.6.1 Cas expérimentaux 120

4.6.2 Résultats et analyse 122

4.6.3 Conclusion 123

5 Activité de navigation globale 130
5.1 Introduction 130
5.2 Le graphe topologique *a priori* 130
5.2.1 Définition 130
5.2.2 Décomposition topologique d'une mine 132
5.3 Conception du navigateur topologique 134
5.3.1 Considérations pour ce module 134
5.3.2 Décomposition du module en fonctions 137
5.3.3 Limite du développement 140
5.4 Conception du détecteur de transitions 140
5.4.1 Conventions typographiques dans la notation 140
5.4.2 Exploiter les senseurs frontal et latéral 141
5.4.3 Exploiter la redondance temporelle 142
5.4.4 Signal de notification d'entrée d'intersection 146
5.4.5 Signal de notification de sortie d'intersection 149
5.4.6 Signaux de confirmation de transition 152
5.4.7 Deuxième phase de l'analyse frontale 152
5.4.8 Décomposition du module en fonctions 155
5.5 Implantation du détecteur de transitions 160
5.5.1 Détection des branches 160
5.5.2 Détection des régions blanches dans l'image 168
5.5.3 Analyse pour la notification d'entrée 170
5.5.4 Analyse pour la jonction de sortie 171
5.5.5 Analyse pour les confirmations de transitions 172
5.5.6 Déroulement normal dans l'environnement 173
5.6 Résultats de la détection des transitions 175
5.6.1 Cas expérimentaux 175
5.6.2 Résultats et analyse 176

5.6.3 Conclusion 179

6 Conclusion 185

Bibliographie 217

Liste des tableaux

3.1 Comparaison des 2 principes sensoriels 47

4.1 Fonction de gestion des mesures d'arpentage brutes 74
4.2 Fonction de segmentation en parois 75
4.3 Fonction d'ordonnancement des points de parois 76
4.4 Fonction d'élimination des points extérieurs 77
4.5 Fonction de gestion des cartes métriques locales 78
4.6 Fonction d'estimation initiale de la pose relative 87
4.7 Fonction d'appariement entre jonction et carte locale 87
4.8 Fonction de correction de la pose relative 88
4.9 Conventions typographiques dans la notation 90
4.10 Fonction d'arpentage de la jonction 101
4.11 Fonction de segmentation des parois 101
4.12 Fonction d'estimation de largeur de galerie 102
4.13 Fonction de calcul des repères propres 102
4.14 Fonction de modélisation des parois 102
4.15 Fonction de recherche du segment minimal 103
4.16 Fonction de positionnement du référentiel local 103
4.17 Les 12 chemins de traversée d'intersections 121
4.18 Amplitude et inclinaison maximale des trajectoires du point M_f versus l'amplitude et la période des déviations appliquées à M_r 121

5.1 Fonction d'interface au chemin global 139
5.2 Fonction d'interface au graphe topologique 139
5.3 Fonction d'interprétation de branchement 140
5.4 Conventions typographiques dans la notation 141
5.5 Fonction de détection des branches 155
5.6 Fonction de gestion de l'image binaire 157
5.7 Fonction de détection de régions blanches dans l'image 157
5.8 Fonction d'analyse pour la notification d'entrée 158
5.9 Fonction d'analyse pour la jonction de sortie 158
5.10 Fonction d'analyse pour les confirmations de transition 159
5.11 Fonction de détection périodique des RLs 160
5.12 Constantes géométriques pour la détection des branches 161
5.13 Paramètres utilisés dans l'analyse pour les confirmations 172
5.14 Les 14 chemins de traversée d'intersections 175
5.15 Valeurs numériques des paramètres de base 176
5.16 Valeurs des paramètres pour les analyses frontale et latérale 177
5.17 Statistiques de la longueur des jonctions d'entrée et de sortie 180

Liste des figures

2.1 Méthodes pour mesurer les angles en horizontale 31
2.2 Techniques de mesure des hauteurs 33
2.3 Méthodes pour faire des mesures latérales 35

3.1 réflexion du signal transmis 45
3.2 Sous-échantillonnage du profil 47
3.3 Configuration sensorielle pour l'arpentage 50
3.4 Détection de la fin d'une galerie 52
3.5 Détection d'un évasement 53
3.6 Détection d'un coin de mur. 54
3.7 Influence de α_{front} 55
3.8 Influence de d_{max} 56
3.9 Navigation dans un tronçon droit 58
3.10 Configuration sensorielle du robot 59
3.11 Activités principales, modules et échanges de données et signaux . . . 61
3.12 Jonction et référentiel local 66
3.13 Relocalisation métrique 67
3.14 Erreurs de mesure de trajectoire par odométrie 68
3.15 Déformation non-rigide de la carte d'une galerie 69

4.1 Collecteur de mesures : échanges de données et signaux 73
4.2 Étiquetage topologique des parois 75

4.3 Notion de points intérieurs et extérieurs à la carte 77
4.4 Relocalisateur métrique : échanges de données et signaux 86
4.5 Position du référentiel local dans la jonction 90
4.6 Stabilité du RL détecté 92
4.7 Observations partielles de jonctions 92
4.8 Algorithme itératif de recherche du segment minimal 95
4.9 Difficultés pour la recherche du segment minimal 96
4.10 Influence de la trajectoire sur la répartition des mesures 98
4.11 Trous dans les parois dans des virages accentués 99
4.12 Détecteur de référentiels locaux : échanges de données et signaux . . 100
4.13 Conversion du profil en distance latérale 104
4.14 Sélection du mur commun d'intersection 106
4.15 Nécessité de segmenter les parois mesurées 107
4.16 Traitement des mesures pour estimer Glarg 109
4.17 Axe médian et repère propre d'une paroi de jonction 111
4.18 repère propre d'un mur commun 112
4.19 Suppression des points très éloignés de $\mathcal{P}_5$ 114
4.20 Exemple de modélisation de parois (d_{arp} = 50 cm) 115
4.21 Abscisses initiales, directions de recherche et bornes en z_1, z_2 116
4.22 Contraintes sur la minimisation 119
4.23 Les cinq intersections d'essai 125
4.24 Résultats pour les 12 traversées d'intersections 126
4.25 Résultats pour les 12 traversées d'intersections (suite) 127
4.26 Résultats pour les 12 traversées d'intersections (fin) 128
4.27 Statistiques de la position des RLs détectés 129

5.1 Représentation topologique d'une mine 131
5.2 Modélisation topologique des intersections 133
5.3 Rôle du navigateur topologique 135

5.4 Navigateur topologique : échanges de données et signaux 138
5.5 NB comme signature pour la notification d'entrée 143
5.6 Occlusions transitoires dans le champ de vue frontal 144
5.7 Fausses LVs dans le champ de vue frontal 145
5.8 Fluctuations de Dlat en fonction de la trajectoire 145
5.9 Évolution des profils frontaux lors d'une traversée d'intersection . . . 147
5.10 Deux représentations différentes pour le traitement par lots 148
5.11 Notification de sortie basée sur (NR = 2) vient trop tard 150
5.12 Limiter le champ frontal en angle pour l'analyse de NB, NR 151
5.13 Détection périodique de RLs 153
5.14 Convergence des RLs au milieu d'une intersection 154
5.15 Détecteur de transitions : échanges de données et signaux 156
5.16 Longueur d'un segment d'occlusion et angle soutendu par une LV . . 161
5.17 Définition d'une LV et d'une occlusion pour la détection 163
5.18 Validation des candidats d'occlusion ouverte 164
5.19 Validation des candidats d'occlusion fermée 165
5.20 Exemple de détection de branches 166
5.21 Sensibilité aux points erronés dans les plages sans mesure 167
5.22 Erreurs dans la détection des occlusions fermées 167
5.23 Changements de pixel blanc à noir dans le contour de l'image binaire 169
5.24 Influence de la détection des branches et de la trajectoire sur NR_i . . 170
5.25 Processus complet de détection des transitions 174
5.26 Les cinq intersections d'essai 181
5.27 Résultats pour les 14 traversées : poses trouvées 182
5.28 Résultats pour les 14 traversées : poses trouvées (suite) 183
5.29 Résultats pour les 14 traversées : poses trouvées (fin) 184

A.1 Création des objets géométriques de la mine 196
A.2 Déplacement du robot dans la mine virtuelle 198

A.3 Projection des rayons laser 199
A.4 Logiciel de gestion de la navigation en mine virtuelle 201
A.5 Visualisation de la navigation en mine virtuelle 202

B.1 Système tel que réalisé par le simulateur 204
B.2 Topologie de la mine virtuelle utilisée par le simulateur 206
B.3 Sélection interactive du chemin global 207
B.4 Schéma de l'interface de navigation avec ses 4 quadrants 208
B.5 Navigation dans la mine virtuelle, affichage des profils et des marqueurs d'événements 209
B.6 Interface usager graphique du programme `MineArpent` 211
B.7 Détection des transitions à l'état *AnalyseFrontEn* 213
B.8 Détection des transitions à l'état *AnalyseLatSo* 214
B.9 Vue 2D de la segmentation des parois en galeries et intersections . . . 215
B.10 Vue 3D de la segmentation des parois en galeries et intersections . . . 216

Liste des annexes

A Banc expérimental de navigation en mine virtuelle 195

A.1 Génération de la mine virtuelle 195

A.2 Déplacement du véhicule 198

A.3 Calcul des profils télémétriques 199

A.4 Interface de visualisation 200

B Simulateur de l'arpentage d'une mine souterraine 203

B.1 Présentation générale 203

B.2 Phase 1 : sélection du chemin global 205

B.3 Phase 2 : navigation dans la mine virtuelle 206

B.4 Phase 3 : détection des transitions de structure 210

B.5 Phase 4 : segmentation des parois mesurées 212

Chapitre 1

Introduction

L'opération d'une mine souterraine englobe de nombreuses activités, chacune contribuant à la fonction principale de l'exploitation qui est d'extraire et de traiter le minerai provenant d'une formation géologique contenant des matières (métaux ou autres) intéressantes pour leur valeur économique. Une fois la formation minérale identifiée par l'exploration géologique, un système d'excavations est planifié qui permettra d'atteindre le ou les gisements souterrains. On prévoit à cette fin des voies d'accès aux différents niveaux prévus dans la mine, des voies de transport du minerai vers la surface et des zones d'extraction du minerai à proprement parler. Lorsque les premières voies d'accès et de transport sont creusées, la mine peut entamer la phase active de son cycle de vie.

Une activité en particulier qui est nécessaire à la gestion de l'exploitation est *l'arpentage* de la mine. En effet, il est important de connaître de manière précise l'étendue et la condition des diverses excavations réalisées. Celles-ci pourront diverger à des degrés plus ou moins importants de leur géométrie planifiée *a priori*. D'autre part, la forme des excavations évolue continuellement pendant le cycle de vie de la mine. L'extraction du minerai, l'ouverture de nouvelles régions, le remblayage de régions inactives, les contraintes de pression exercées sur les ouvertures qui y sont creusées, sont autant de processus qui modifient la condition et l'étendue des excava-

tions. Ainsi, à plusieurs moments dans l'évolution de la mine, des équipes d'arpenteurs descendent dans celle-ci pour y mesurer la forme des ouvertures[1] ; ces mesures sont ensuite utilisées pour dresser des cartes montrant l'état actuel des excavations.

Une tendance notable aujourd'hui dans l'industrie minière est vers l'emploi de technologies d'automatisation, dans le but d'augmenter le rendement des différents processus et aussi de réduire le personnel à envoyer dans les zones souterraines. Des exemples de cette évolution technologique sont : la téléopération appliquée à plusieurs types de machines faisant l'extraction du minerai ; la navigation guidée (sans chauffeur à bord) de véhicules de transport dans des voies souterraines ; l'automatisation de plusieurs étapes dans le traitement du minerai extrait ; et l'informatisation de la planification et la gestion de l'exploitation minière. Cependant, nous remarquons que cette tendance ne s'est pas réellement étendue jusqu'au processus de l'arpentage de la mine. Certes, on a fait plusieurs pas vers l'introduction d'outils informatiques dans la prise des mesures d'arpentage[2]. Mais même avec ces outils, la tâche d'aller prendre les mesures géométriques sur la forme des excavations repose toujours sur des équipes d'arpenteurs. Il serait pourtant souhaitable d'automatiser cette tâche, et ce pour deux raisons. D'abord, une mine souterraine est un milieu difficile d'accès et souvent dangereux et il y a donc intérêt du point de vue de la sécurité à y réduire autant que possible la présence humaine. De plus, il y a dans l'automatisation de l'arpentage le potentiel de réduire le coût économique de cette tâche, mais aussi et surtout de pouvoir effectuer cette tâche plus souvent et donc de mieux suivre la progression des diverses opérations souterraines[3].

Dans cette optique, nous nous sommes proposés comme sujet de recherche de développer un système d'arpentage automatisé pour les mines souterraines. Plusieurs

---

[1] La section 2.3 de la revue bibliographique décrit les techniques usuelles d'arpentage dans les mines.

[2] Voir par exemple Labrosse (1998, section 4.2) pour une revue des développements récents dans ce domaine.

[3] Ce dernier avantage devient d'autant plus intéressant lorsque l'arpentage et la modélisation géométrique de la mine s'inscrivent dans le cadre d'une base de données globale informatisée des opérations minières ; *cf.* Labrosse (1998, chapitre 3) et Labrosse, Debanné, Hervé et Cohen (1998).

critères s'imposent quant à la conception de notre système. Premièrement, nous voulons que le processus de collecte de mesures géométriques soit *autonome*, c'est-à-dire n'exigeant pas d'intervention humaine à l'intérieur de la mine. Deuxièmement, nous voulons que le système soit applicable non seulement à la tâche de l'arpentage, mais aussi à des tâches de *navigation autonome* dans le milieu souterrain, par exemple le transport automatisé du minerai entre des points de chargement et déchargement. Troisièmement, nous voulons éviter que la fonction de navigation nécessite l'installation de structures de guidage dans la mine, car celles-ci sont coûteuses, facilement endommagées et doivent être remplacées régulièrement. Tous ces critères de performance nous indiquent comme point de départ pour notre système l'emploi d'une *plate-forme mobile* (*robot*) capable de se déplacer de manière autonome dans un réseau de voies souterraines. Physiquement, il s'agit d'un véhicule équipé d'un capteur interne mesurant les déplacements de la plate-forme et de plusieurs capteurs externes qui renseignent le robot sur la forme des parois d'excavations dans son environnement local.

Le développement *complet* d'un système d'arpentage automatisé implique de nombreux problèmes à résoudre, autant sur le plan matériel que conceptuel. L'implantation physique d'un "robot-arpenteur" entraîne toute une panoplie de défis techniques, chacun d'eux demandant un travail non-négligeable. Nous pensons entre autres aux problèmes mécaniques associés à l'implantation d'un contrôleur pour la navigation sur un véhicule réel ; à l'étalonnage des différents senseurs afin d'en obtenir des mesures correctes ; et aux phénomènes de bruit sensoriel, inévitables dans la situation réelle, qui rendent plus difficile l'exploitation des données sensorielles. Toutefois, l'objectif de notre recherche est de *démontrer le concept* d'un système d'arpentage automatisé sans forcément réaliser un tel système physiquement. Par conséquent, nous réalisons notre système *en simulation* : le déplacement du robot et la capture des données sensorielles se déroulent dans un milieu minier simulé sur ordinateur. Le banc expérimental en simulation nous permet de traiter les problèmes d'ordre matériel comme ceux que

nous venons de mentionner soit en en faisant abstraction, soit en contrôlant leur niveau de difficulté par le biais de paramètres relatifs au robot ou à l'environnement. Par ailleurs, nous évitons les difficultés et les coûts considérables qui s'imposeraient dans le cas d'expérimentations sur un site souterrain réel.

Sur le plan conceptuel, les problèmes à résoudre s'inscrivent presque tous dans le domaine général de la robotique mobile. Il y a d'une part la question du déplacement autonome du robot dans la mine; c'est un problème de contrôle local du véhicule pour la navigation. Il ne s'agit pas cependant pour le véhicule de suivre un chemin géométriquement précis préétabli car justement nous ne connaissons pas au préalable la forme précise des excavations à parcourir. Le contrôle pour la navigation doit donc intégrer une fonction perceptuelle afin de guider le véhicule par rapport aux parois observées en continu. D'autre part, la capacité d'arpenter la mine sous-entend celle de *localiser* précisément la plate-forme mobile (à partir de laquelle sont prises les mesures) par rapport à un ou des points de repère dans la mine. Le senseur interne à bord du robot ne suffit pas à cette fonction car, quelle que soit sa réalisation technologique, les mesures qu'il fournit vont progressivement diverger de la trajectoire réellement suivie[4]. Comme nous le verrons dans la revue bibliographique, la localisation métrique fait généralement intervenir une *représentation du milieu* (ou *carte*), soit fournie au préalable, soit construite à mesure que le robot se déplace, et avec laquelle sont comparées les observations de l'environnement afin de corriger l'information provenant du senseur interne. Dans notre cas, nous exploitons à la fois la notion de carte métrique construite à partir des mesures externes et celle de points de repère identifiés dans l'environnement pour accomplir les tâches (reliées) de l'arpentage et de la localisation métrique. En plus de cette première forme de localisation, qui est à une échelle locale, il faut localiser le robot à une échelle globale, non-géométrique dans la mine pour pouvoir lui faire suivre des chemins spécifiés à

[4]Il s'agit là d'une difficulté fondamentale dans le domaine de la robotique mobile. Nous reviendrons à la question de la mesure interne des déplacements du robot dans le corps de l'ouvrage.

ce deuxième niveau. Ces deux niveaux de localisation sont reliés dans notre système par l'emploi de points de repère détectés dans l'environnement. Enfin, nous devons gérer les mesures de parois prises par le robot de manière à en produire des données d'arpentage dans un format adéquat pour la construction d'un modèle de la géométrie des excavations parcourues.

Remarquons que tous les problèmes énumérés ci-dessus font appel à l'interprétation des mesures sensorielles pour accomplir diverses tâches ; de ce fait, notre projet consiste à développer les *capacités perceptuelles* du robot. L'élaboration de méthodes spécifiques d'analyse sensorielle pour les différentes tâches s'appuie sur l'établissement d'une *architecture* de système. Cette architecture est hiérarchique : elle identifie à un premier niveau plusieurs activités principales qu'un système complet d'arpentage automatisé doit mener en parallèle pendant que le robot circule dans la mine. Chacune de ces activités principales est ensuite décomposée jusqu'à un niveau fonctionnel, c'est-à-dire un niveau qui spécifie des opérations précises sur les données sensorielles et/ou les structures de données internes au système. C'est au niveau fonctionnel que se situent les raisonnements algorithmiques, tandis que les niveaux hiérarchiques supérieurs traduisent par leur structure les grandes lignes de notre approche. Par souci de limiter l'envergure de ce projet, nous ne développons qu'un sous-ensemble des composantes de l'architecture jusqu'à l'étape expérimentale. Ainsi, certaines composantes aux niveaux supérieurs ne sont pas développées et d'autres sont élaborées jusqu'au niveau fonctionnel mais sans implanter ces fonctions. Les parties du système que nous choisissons d'implanter et tester sont celles que nous jugeons les plus intéressantes en tenant compte de leur nouveauté d'un point de vue de recherche, de notre objectif de faire une preuve de concepts et de la contrainte que nos essais se déroulent en simulation. Néanmoins, notre architecture se veut complète et procure un cadre de développement pour que d'autres travaux puissent compléter le nôtre. En particulier, nous laissons de côté tout l'aspect de contrôle local du véhicule pour la navigation ; mais il serait envisageable de rajouter cette capacité au système sur

la base de travaux existants[5] et futurs sur la navigation autonome en milieu minier souterrain.

Dans la terminologie de la robotique mobile, ce projet peut être vu comme un problème d'*exploration* d'un environnement initialement inconnu ; mais il faut alors faire une distinction importante. En fait, nous considérons que la région de la mine que le robot va parcourir n'est pas complètement inconnue mais plutôt que, en vertu des connaissances qu'on est susceptible d'avoir sur la planification de la région en question et sur les opérations qui s'y sont effectuées, on est capable de la décrire de manière grossière avant l'étape d'arpentage. Ces connaissances *a priori*, combinées à la nature particulière de l'environnement minier souterrain, sont telles que la région à parcourir peut être représentée comme un ensemble de sous-régions qualitativement distinctes et interconnectées. Nous nous basons sur un projet de recherche connexe pour rendre cette notion plus précise. Le travail de Labrosse (1998) porte sur la modélisation générique d'objets et, comme exemple d'application, la modélisation des mines souterraines. Dans ce travail, la mine souterraine est décomposée *topologiquement* en *galeries*, *intersections de galeries* et *chambres*, ces catégories de sous-régions étant différenciées par leurs fonctions dans la mine et leurs types de géométrie. Nous appelons ces sous-régions les *éléments topologiques* de la mine. Sur cette base, nous supposons que la région de la mine à parcourir est décrite au robot par un *graphe topologique* dans lequel les arcs symbolisent les galeries et les nœuds symbolisent les intersections et les chambres[6]. Dans ce contexte, l'activité d'arpentage consiste à découvrir la structure à une échelle fine des éléments topologiques indiquées par le "squelette" de la mine qui est le graphe.

L'existence du graphe topologique *a priori* a des répercussions importantes sur la conception de notre système. La conséquence fondamentale est que la localisation du robot au niveau global dans la mine se fait par rapport au graphe topologique.

---

[5] Voir la section 2.2 de la revue bibliographique.

[6] Nous reviendrons à la question de la décomposition topologique de la mine dans le corps de l'ouvrage.

D'une part, cela signifie que, pour la tâche d'arpenter la mine comme pour celle de se déplacer entre deux points, le parcours à suivre par le robot lui est spécifié comme un *chemin global* prenant la forme d'une liste d'éléments (nœuds et arcs) adjacents dans le graphe topologique. Il s'agit alors pour le robot d'effectuer une trajectoire dans l'environnement réel qui correspond au chemin global. Pour savoir où il se trouve à mesure qu'il se déplace, le robot doit détecter les *transitions* entre les éléments topologiques qu'il traverse. De cette manière, l'existence du graphe simplifie le problème de la localisation globale comparé au cas d'un environnement totalement inconnu au départ. De fait, nous n'avons pas besoin de distinguer globalement tous les endroits rencontrés[7], mais seulement de reconnaître les transitions entre éléments topologiques adjacents. Un autre avantage de l'existence *a priori* du graphe est que le processus de localisation globale ne dépend pas de la mesure interne des déplacements qui, comme nous l'avons vu, n'est fiable qu'à court terme.

Mis à part le graphe topologique, notre approche pour résoudre les problèmes perceptuels identifiés ci-haut se distingue par des choix délibérés quant à la nature des informations sensorielles et notre façon de les traiter. Les senseurs externes sur la plate-forme mobile sont du type *télémétrique* : ils mesurent la distance vers les parois autour du robot dans plusieurs plans orientés latéralement par rapport au véhicule et dans un plan frontal devant le véhicule. Nous attribuons des rôles différents aux senseurs selon leur orientation par rapport au robot. Les mesures provenant des senseurs latéraux servent à établir des cartes[8] métriques des éléments topologiques traversés, pour l'arpentage. En revanche, le télémètre frontal n'est pas employé dans l'arpentage, mais plutôt dans la localisation globale. Il permet d'observer des changements qualitatifs dans la structure de l'environnement devant le robot ; ces événements perceptuels caractérisent l'arrivée de transitions entre éléments topologiques. Les senseurs latéraux sont aussi employés pour reconnaître les transitions signalées

[7] comme le font les travaux vus dans la section 2.1.4 de la revue bibliographique.

[8] Les cartes à cette étape-ci n'incorporent pas de modélisation géométrique des mesures brutes comme le ferait la carte de la mine construite *a posteriori*.

par l'analyse frontale, lors du passage du robot par ces endroits. C'est un processus qui détecte des changements importants dans la structure locale autour du robot tels que des évasements ou rétrécissements de la voie navigable.

Les analyses latérale et frontale pour reconnaître les transitions sont des processus *dynamiques* et les événements perceptuels détectés sont intrinsèquement liés au mouvement du robot dans l'environnement. C'est donc du point de vue de la navigation que ce sont des phénomènes intéressants. Par ailleurs, nous pouvons qualifier notre approche comme étant purement *géométrique*, par opposition à des approches relevant du domaine de la vision artificielle, où l'information sensorielle prend la forme d'images photométriques prises dans l'environnement. Par exemple, le travail de Béranger (1996) porte aussi sur la navigation autonome dans des milieux tels que les mines souterraines ; mais ici, on exploite une caméra frontale montée sur le robot pour reconnaître des points de repère (qui sont les intersections entre galeries) plutôt que des senseurs télémétriques.

Le corps du présent ouvrage est organisé comme suit. Le chapitre 2 présente une revue bibliographique qui comporte trois volets : l'exploration d'environnement[9], la navigation autonome en milieu minier souterrain et les techniques usuelles d'arpentage dans les mines souterraines. Dans le chapitre 3, nous définissons de façon plus précise les objectifs du système, le modèle de l'environnement et les modalités sensorielles du robot, puis présentons l'architecture du système d'arpentage automatisé. Dans les chapitres 4 et 5, nous rentrons dans le développement à proprement parler du système. Le chapitre 4 traite de la partie du système qui réalise l'activité principale de l'arpentage ; au sein de cette activité, nous implantons une méthode d'analyse des mesures latérales pour déterminer des points de repère qui caractérisent la forme des transitions entre éléments topologiques. Les points de repère ainsi calculés servent de référentiels pour les mesures de parois formant les cartes métriques produites par

[9]En fait, nous ne faisons pas vraiment de l'exploration d'environnement car la topologie de la mine nous est fournie *a priori* ; cependant, notre projet partage plusieurs problèmes conceptuels avec les travaux en exploration.

l'arpentage. Le chapitre 5 traite de l'activité principale de la navigation globale, c'est-à-dire la localisation du robot par rapport au graphe topologique. À l'intérieur de cette activité, nous implantons le processus de reconnaissance des transitions entre éléments topologiques à partir des mesures frontales et latérales. La méthode présentée au chapitre 4 pour identifier des points de repère est réutilisée ici, cette fois pour confirmer la présence de transitions détectées auparavant par l'analyse des mesures frontales. Enfin, nous concluons cet ouvrage en évaluant notre travail selon l'objectif initial d'une preuve de concepts et selon ses contributions innovatrices. Aussi, nous indiquons à quels niveaux doit être complété le développement conceptuel de notre système afin d'ouvrir la voie vers son implantation éventuelle sur une plate-forme robotique et sur un site minier réel.

Chapitre 2

Revue bibliographique

2.1 Exploration d'environnement

2.1.1 Quelques définitions

Dans cette revue bibliographique, nous nous intéressons tout d'abord aux systèmes d'exploration d'environnements. Il convient de fournir la définition suivante :

> *définition* : l'exploration d'environnement est l'activité qui consiste en l'acquisition de données décrivant un milieu physique, et dont l'un des buts est de construire une représentation, ou *carte*, de ce milieu.

Cette définition sous-entend un certain nombre de choses :

- que l'environnement physique est initialement inconnu ou seulement partiellement connu ;
- qu'un agent intelligent, capable de se déplacer, effectue l'exploration. Par exemple, l'exploration peut être faite par un humain ou par un robot autonome ;
- que l'agent acquiert des connaissances sur le milieu au moyen d'appareils de mesure. Deux exemples sont les organes sensoriels de l'humain et les appareils sensoriels installés à bord d'un robot ;

- qu'il existe un modèle d'environnement selon lequel on interprète les données acquises pour construire la représentation ;
- que l'on veut par la suite utiliser cette représentation. Par exemple, une carte géologique d'un territoire permet de gérer l'exploitation de ressources. Ou encore, une carte routière permet de planifier de trajets.

Nous nous intéressons aux systèmes qui ont pour but d'utiliser la représentation de l'environnement pour la *navigation*. Plus précisément, la carte de l'environnement sert à la planification de chemins et à la localisation dans l'environnement. Nous parlons aussi, dans certains cas, de systèmes de navigation (plutôt que d'exploration) qui utilisent un modèle de l'environnement connu *a priori* pour aider un agent mobile à se localiser. De plus, nous nous intéressons particulièrement à des systèmes où l'agent explorateur est mécanisé et autonome (*i.e.* robot mobile autonome), et où le milieu que l'on veut représenter est celui dans lequel évolue l'agent (un contre-exemple serait la cartographie aérienne).

Nous divisons notre recherche selon le type de modèle qui est utilisé pour représenter l'environnement. Nous distinguons les types de modèle d'environnement suivants :

1. modèle géométrique ;
2. modèle de grille d'occupation ;
3. modèle topologique ;

Ces termes sont définis dans les sections qui suivent. Pour chacun de ces modèles, notre étude est en plus divisée selon le type d'environnement qui est exploré : environnement non-structuré, partiellement structuré ou structuré. Nous définissons ces trois termes de la manière suivante :

> *environnement non-structuré* : un environnement où les espaces rencontrés sont de natures trop diverses et où les surfaces sont de formes trop

irrégulières pour pouvoir les exprimer de manière systématique. Par exemple, un terrain extérieur naturel peut contenir des zones de forêt, de montagne, de pré, de route, etc. Dans un tel milieu, il est clairement très difficile d'établir des règles génériques décrivant la forme du sol et des objets rencontrés.

environnement partiellement structuré : un environnement où, à une échelle globale, il est possible d'établir des règles décrivant la nature des espaces rencontrés, mais où, à une échelle plus fine, les surfaces qui constituent le milieu sont de complexité élevée. Par exemple, plusieurs milieux peuvent être qualifiés de *milieux de corridors* ; l'environnement routier et l'environnement minier souterrain en sont des exemples. Dans ces milieux, l'agent évolue dans un espace qui est décomposable en corridors (*e.g.* les routes ou les tunnels miniers), en intersections et (s'il y a lieu) en zones ouvertes. Cependant, les surfaces qui composent le milieu (*e.g.* les parois rocheuses de la mine, la chaussée avec ses irrégularités, le bord de la route) ne sont pas de forme géométrique simple.

environnement structuré : un environnement où, que se soit à une échelle globale ou à une échelle plus fine, il est possible de décrire la nature des espaces et la forme des surfaces rencontrées par des règles relativement simples. Par exemple, le milieu de salles et de couloirs à l'intérieur d'un édifice à bureaux est un milieu structuré.

2.1.2 Représentation géométrique du milieu

La carte que l'on constitue peut décrire les espaces et objets rencontrés par des agglomérations de formes géométriques relativement simples, situées de manière précise dans un référentiel global cartésien. C'est ce que nous appelons carte *géométrique* du milieu. Le plus souvent, l'environnement est doté d'un sol sensiblement plat, et les

structures les plus importantes à représenter ne varient pas en hauteur. On peut donc réduire l'environnement à sa projection 2D parallèle au sol à la hauteur perçue par l'agent. Les parois délimitant l'espace navigable sont représentées par des segments de droite, tandis que des polygones (regroupements de segments) indiquent dans la carte les contours des objets repérés. À cause, d'une part, des contraintes de modélisation imposées par l'emploi de formes géométriques simples, et d'autre part, de l'incertitude introduite dans la carte par le déplacement du robot et par ses capteurs (phénomène discuté dans la suite), ce type de représentation s'applique surtout aux milieux structurés d'échelle restreinte.

Tous les systèmes d'exploration par robot autonome cherchant à construire une carte géométrique du milieu inconnu doivent accomplir simultanément deux tâches essentielles : localiser le robot par rapport à un ou des référentiels externes, et intégrer les données sensorielles acquises dans une carte globale cohérente. Dans un système réel, les deux tâches sont rendues plus difficiles par le fait que les modalités sensorielles internes (odométrie au moyen d'encodeurs de roues, ou senseurs inertiels) et externes (télémètre sonar, laser à balayage ou lumière structurée) sont entachées d'erreurs de mesures. De ce fait, un processus d'exploration qui ne tient pas compte de ces incertitudes divergera inévitablement d'une modélisation correcte du milieu. Face à ce problème, les projets que nous allons voir proposent diverses stratégies pour corriger à la fois les erreurs de positionnement et de modélisation qui s'introduisent dans l'exploration.

Chatila et Laumond (1985) sont parmi les premiers à proposer l'idée d'exploiter les données télémétriques acquises non seulement pour construire la carte mais aussi pour corriger les erreurs d'intégration de la trajectoire du véhicule commises par l'odométrie. Pour traiter des incertitudes de mesure, on emploie un modèle de bruit aléatoire gaussien, de variance variable, ce qui permet de *quantifier* les erreurs associées à chaque nouvelle donnée (position du robot, sommets d'un nouveau segment). On présente des règles pour fusionner des données géométriques anciennes et nouvelles

en les pondérant par leurs erreurs (variances) respectives. Lorsque le robot redécouvre des régions de l'environnement déjà représentées dans la carte, une isométrie est calculée qui superpose les deux structures dans le référentiel global. Si la structure dans la carte est de certitude insuffisante, l'isométrie, ainsi que les différences dans la nouvelle structure, servent à modifier la carte. Dans le cas contraire, l'isométrie sert à corriger la pose du robot. On applique un principe introduit par Brooks (1985) pour rétro-propager cette correction vers les positions précédentes dans la trajectoire.

Le principe télémétrique des ondes ultrasoniques (sonar), bien que fournissant une résolution angulaire faible et susceptible au problème des réflexions multiples d'ondes, est cependant relativement robuste et peu coûteux. De ce fait nous le retrouvons souvent dans les applications de robotique mobile. Leonard et Durrant-Whyte (1992) proposent un modèle rigoureux d'interprétation des données sonar basé sur les Régions de Profondeur Constante (RPC), qui sont des arcs de cercles centrés sur le véhicule et de rayons équivalents aux distances vers des surfaces réfléchissantes. On combine des RPCs détectées au cours du temps pour en former des objets (ou "cibles") différentes : segments de droites, coins concaves et convexes. Les auteurs développent alors des approches complémentaires pour la localisation du robot et la construction de carte, basées sur le filtre Kalman étendu. Pour la localisation, des équations d'états gèrent la prédiction de la nouvelle pose du véhicule et du vecteur de cibles à observer relativement au robot. Les observations réelles associées aux cibles prédites permettent d'estimer la pose corrigée et la modification des matrices de covariances. Pour la construction de carte, c'est l'inverse qui se produit : la même boucle de prédiction - observation - mise en correspondance va maintenant modifier les cibles dans la carte. Comparée aux approches de Chatila et Laumond (1985) et Crowley (1985), on applique ici un formalisme mathématique plus rigoureux.

Par rapport au système précédent, Leonard, Durrant-Whyte et Cox (1992) rajoutent aux matrices de covariances une deuxième mesure d'incertitude : une fonction de "crédibilité" associée à chaque cible détectée. Cette fonction augmente lorsqu'une

cible prédite est confirmée par l'observation, et diminue dans le cas contraire. Ceci permet d'éliminer de la carte créée les cibles provoquées par des structures transitoires (*e.g.* porte qui s'ouvre brièvement) et par le phénomène des réflexions multiples. (Notons que les résultats ici sont en opposition avec MacKenzie et Dudek (1994), qui supposent que les artefacts du principe sonar constituent des points de repère stables pour la navigation.)

Enfin, dans les travaux de Borthwick, Stevens et Durrant-Whyte (1993) et Borthwick et Durrant-Whyte (1994) ainsi que Rencken (1993) et Rencken (1994), on poursuit les techniques du filtrage Kalman de données, en mettant l'accent sur l'importance d'isoler, dans l'erreur entre les observations prédite et mesurée de structure dans le milieu, les contributions respectives de l'intégration de trajectoire et de la perception. À chaque nouvelle perception, on tente d'abord de corriger l'odométrie en combinant l'équation d'état du véhicule et les éléments de la carte de certitude élevée trouvant leurs contreparties dans l'observation. Dans un deuxième temps, les nouvelles structures observées, connues dans le repère rectifié du robot, sont intégrées dans la carte. Cette stratégie pour mener simultanément la localisation et la perception sans aucune connaissance initiale du milieu était déjà proposée sous le nom de "relocalisation-fusion" par Moutarlier et Chatila (1989).

2.1.3 Représentation du milieu par grille d'occupation

Dans la deuxième approche de modélisation d'environnement, on veut construire une carte qui montre comment est délimité l'espace navigable par le robot. L'espace cartésien (en général 2D) est discrétisé pour donner une grille de cellules qu'on appelle *grille d'occupation.* L'exploration assigne des probabilités d'occupation aux cellules, et dégage progressivement les régions libres des régions occupées de la carte. Le plus souvent, la valeur d'une cellule varie entre 0 (vide) et 1 (pleine), la valeur 0,5 indiquant l'ambiguïté. Les principes de base de la représentation d'environnements par grille d'occupation furent posés dans les travaux de Moravec et Elfes (1985), Elfes

(1987) et Matthies et Elfes (1988), puis furent adaptés dans de nombreux projets à des applications particulières en environnements structurés et non-structurés.

2.1.3.1 Environnements structurés

La grille d'occupation est utilisée d'abord en exploration robotique par Moravec et Elfes (1985) et Elfes (1987). On y développe une méthodologie pour construire la carte à partir de mesures provenant d'un anneau de senseurs sonar, prises à des instants différents sur la trajectoire d'un robot mobile. L'incertitude spatiale sur la distance mesurée par un senseur sonar est modélisée par une fonction de probabilité d'occupation spatiale du cône balayé par le faisceau acoustique (projeté dans le plan du sol). La mise à jour de la carte à chaque nouvelle série de mesures est basée sur des règles bayesiennes de composition de probabilités conditionnelles.

L'intégration d'informations provenant de senseurs de types différents fait l'objet des travaux de Matthies et Elfes (1988) et Moravec (1988). L'intérêt d'utiliser plusieurs types de capteurs est de compenser les limites de chaque principe sensoriel en portée, en champs de vue ou en résolution angulaire. Ici, ce sont le sonar et la vision stéréo que l'on veut fusionner. Le principe acoustique permet de dégager des zones importantes d'espace libre, mais, à cause de sa basse résolution angulaire, distingue difficilement la forme précise des objets. La vision stéréo, en revanche, a un champ visible plus restreint, mais en général détecte les frontières des objets plus précisément que le sonar. Les résultats expérimentaux démontrent cette complémentarité des deux senseurs.

Nous citons ici quelques projets plus récents qui exploitent le principe de la grille d'occupation en exploration d'environnements structurés. Stuck, Manz, Green et Elgazzar (1994) se proposent de concevoir un robot capable de contourner en temps réel des obstacles mobiles et/ou de formes complexes, en rajoutant à un module de base d'évitement de collision les capacités de représenter l'environnement et de planifier des trajectoires locales d'évitement. La plate-forme mobile est dotée de

deux systèmes sensoriels : un télémètre laser BIRIS (*c.f.* Blais, Rioux, Domey et Beraldin, 1988) monté sur un plateau rotatif et un anneau de capteurs sonar. Les données BIRIS servent à constituer une carte globale du milieu, qui prend la forme d'une *grille d'histogrammes*, version modifiée de la grille d'occupation introduite par Borenstein et Koren (1991*b*) et Borenstein et Koren (1991*a*). Simultanément, le robot exploite les données sonar (de taux plus élevé) pour calculer en continu des grilles d'histogrammes locales, dont il extrait les directions de mouvement libre.

Le système d'exploration décrit par Oriolo, Vendittelli et Ulivi (1995) est semblable à celui de Moravec et Elfes (1985), mais avec cette différence que la carte globale est mise à jour selon des règles de logique floue plutôt que bayesiennes. La planification de chemin consiste à trouver une chaîne de cellules libres connectées dans la grille d'occupation, reliant la position courante à un but prédéfini. Le robot se déplace alors, s'arrêtant avant d'atteindre sa destination si un obstacle devient trop proche ou s'il quitte la région connue dans la carte.

Enfin, Min, Cho, Lee et Park (1996) proposent deux innovations par rapport au modèle de grille d'occupation de Moravec et Elfes (1985). D'une part, ils mesurent la qualité d'une région de la carte établie jusqu'à lors en évaluant son entropie moyenne (qui diminue lorsque les cellules de cette région ont des valeurs qui s'éloignent de 0.5, l'ambiguïté). D'autre part, ils superposent à la carte globale une structure d'arbre quaternaire pour permettre au robot de choisir des destinations temporaires pendant l'exploration. Chaque sous-but est le centre d'une région du arbre quaternaire d'entropie supérieure à un seuil.

2.1.3.2 Environnements partiellement ou non-structurés

L'article de Stentz et Hebert (1995) présente une architecture de planification de chemin et de navigation dans un milieu non-structuré inconnu. La planification de chemin comporte deux modules : le navigateur "SMARTY", qui se concentre sur l'évitement local d'obstacles, et le navigateur global, qui doit assurer que le robot

atteigne sa destination. Au niveau local, l'environnement est représenté par une grille de haute résolution (40 cm), que le module SMARTY remplit par segmentation et analyse de cartes de profondeurs procurées par un télémètre laser. Simultanément, SMARTY évalue un ensemble de trajectoires de courbures différentes selon leurs proximités aux cellules-obstacles. Au niveau global, le milieu est représenté par une grille de dimensions plus grandes et de résolution plus faible (1 mètre), où sont intégrées les descriptions d'obstacles. Cette deuxième carte est exploitée par un algorithme de planification dynamique de chemin. Le suivi de pose du robot se fait grâce à la combinaison d'encodeurs de roues avec un senseur inertiel. Avec ce système, le véhicule autonome NAVLAB II de C.M.U. parvient à traverser un parcours de 1.4 kilomètres dans un terrain de nature variée (route, prés, bois) et à constituer un carte des obstacles rencontrées.

2.1.4 Représentation topologique du milieu

Dans la troisième approche au problème de modéliser l'environnement, le robot cherche à construire une représentation de la *topologie* du monde réel. Ce qui est intéressant dans l'exploration est d'identifier les endroits importants dans l'environnement (les *landmarks*), et de savoir comment naviguer entre ces endroits. L'agent autonome doit faire le tri de toutes les informations qu'il reçoit sur son entourage, pour ne garder que ce qui le renseigne sur les lieux importants. La carte prend la forme d'un graphe, où les nœuds symbolisent les lieux identifiés et les arcs symbolisent les chemins de parcours. Cette forme d'exploration s'applique à plusieurs catégories d'environnements.

2.1.4.1 Environnements réduits à des graphes

Si l'on fait abstraction de la structure physique du monde réel et de l'interaction sensorielle entre celui-ci et l'agent, alors l'environnement est directement réduit à sa représentation symbolique sous forme de graphe. Explorer l'environnement signifie alors construire un graphe topologique interne (*i.e.* en mémoire) qui soit isomorphe

au graphe externe (*i.e.* l'environnement).

Ceci est le point de départ dans les travaux de Dudek, Jenkins, Milios et Wilkes (1991) Dudek, Jenkin, Milios et Wilkes (1993) et Dudek, Freedman et Hadjres (1996). On suppose que le robot a la capacité perceptuelle de détecter toutes les branches lorsqu'il arrive à un endroit et de les ordonner localement par rapport à la branche d'arrivée. Lorsque le robot arrive au même nœud par une branche différente, le deuxième ordonnancement des branches est une permutation du premier (par rotation dans un graphe plan).

Il est facile de montrer qu'en l'absence d'autres informations sensorielles, le robot ne peut pas résoudre le problème d'exploration en général à cause du problème de distinguer entre nouveaux nœuds et nœuds du même degré déjà rencontrés. Par exemple, dans un milieu où tous les nœuds ont le même degré (se rattachent à un même nombre de branches), le robot ne peut pas les distinguer et par conséquent il ne trouve pas une représentation unique du milieu. Cette question fait l'objet de la recherche dans Dudek et al. (1996).

Dans Dudek et al. (1991), on introduit une autre source de renseignements perceptuels : le robot dispose de plusieurs marqueurs, tous distinguables entre eux, qu'il peut déposer et ramasser aux nœuds du graphe réel. Pendant l'exploration, il maintient un sous-graphe exploré S et une liste d'arcs non-explorés U. On présente un algorithme d'exploration dans lequel les marqueurs sont utilisés pour distinguer les nœuds inconnus de ceux déjà placés dans le sous-graphe S. L'exploration est terminée quand la liste U est réduite à zéro. Le graphe S est alors équivalent au graphe réel. L'algorithme réussit même avec un seul marqueur, car il suffit qu'un seul nœud réel à la fois soit globalement distinguable pour que le problème d'exploration puisse être résolu.

Dans Dudek et al. (1993), on fait les mêmes suppositions que Dudek et al. (1991) au niveau du milieu et du robot (qui possède toujours des marqueurs), et en plus on suppose que le robot possède initialement le graphe topologique du milieu. On

peut maintenant résoudre un problème complémentaire à celui de l'exploration : localiser le robot dans le milieu en l'ignorance de sa position initiale. D'abord, on postule toutes les hypothèses possibles sur l'identité du nœud initial et d'une branche de référence quittant ce nœud ($O(n^2)$ hypothèses, n étant le nombre de nœuds du graphe). Ensuite, on procède à l'exploration du milieu comme dans Dudek et al. (1991), en éliminant progressivement les hypothèses en conflit avec le graphe exploré. Bien que cette méthode de localisation soit démontrable rigoureusement, en terme d'efficacité elle représente le scénario du pire cas, qui sera de loin améliorée dans d'autres contextes de navigation.

2.1.4.2 Environnements structurés

L'article de Kuipers et Byun (1987) est sans doute l'un des premiers à amener le principe de la modélisation qualitative (plutôt que métrique) vers le domaine de l'exploration robotique de milieux structurés. On suppose que le robot dispose d'un senseur télémétrique omnidirectionnel de faible résolution angulaire. On raisonne que pour identifier les lieux intéressants, le robot doit rechercher des extremums de plusieurs "signatures sensorielles" telles que le degré de symétrie autour de lui, le nombre de directions de déplacement possible ou la distance vers l'objet le plus proche. Pour distinguer un lieu nouveau d'un lieu connu de signature semblable, le robot génère un parcours dans le graphe qui l'amène dans des nœuds voisins, pour ensuite vérifier si oui ou non ses observations réelles correspondent aux endroits connus.

En se servant du contexte posé par Kuipers et Byun (1987), Engelson et McDermott (1992) mettent l'emphase sur la question de comment corriger les *erreurs* qui peuvent s'introduire dans le graphe topologique, en exploitant l'information de déplacement relatif provenant de l'odométrie. Si la position géométrique d'un lieu reconnu est erronée, on augmente son intervalle d'incertitude pour inclure la position actuelle du robot. Si au contraire la position du robot dans le graphe s'avère

non-plausible face aux mesures sensorielles actuelles, cette position topologique est déplacée vers un des nœuds voisins. Le robot peut décider que deux nœuds voisins représentent le même lieu, pour alors les fusionner, ou inversement scinder en deux un nœud qui peut avoir englobé deux lieux distincts à cause d'erreurs odométriques importantes. En outre, on prévoit une stratégie pour supprimer de la carte des nœuds ou arcs engendrés par des structures qui s'avèrent transitoires.

Borghi et Brugali (1995) reprennent la structure de carte de Engelson et McDermott (1992) (appelée "diktiométrique") ; on définit la "vue" caractérisant un endroit comme étant une carte géométrique 2D composée de coins et segments décrivant les contours d'objets visibles à cet endroit. Tant que le robot est capable de mettre en correspondance des structures autour de lui avec la carte locale du dernier endroit connu, les nouvelles données télémétriques sont fusionnées dans cette carte et la pose du robot mise à jour dans le référentiel de l'endroit. Lorsqu'il n'est plus possible de reconnaître des objets environnants (par exemple dans un couloir où l'on ne perçoit que les deux murs), le robot se trouve alors dans un nouveau lieu topologique.

Yamauchi et Beer (1996) appliquent le principe de la modélisation topologique à des environnements *dynamiques*, où l'on classifie les changements de structure comme étant soit transitifs (à court terme), topologiques (de plus long terme) ou encore perceptuels (causés par des erreurs sensorielles, notamment par l'odométrie). On dénote la structure de carte ici comme Réseau d'Endroits Adaptatif (REA). Dans le REA, des niveaux de confiance sont associés à tous les nœuds et arcs du graphe. Un nouveau nœud est rajouté au graphe dès que la distance du robot vers l'endroit connu le plus proche dépasse un seuil. Un lien est créé quand le robot passe la première fois entre deux nœuds, et son niveau de confiance est modifié chaque fois que le robot réussit ou échoue à réemprunter ce lien. Ce type de modèle est en effet bien adapté à représenter un milieu dynamique. Cependant, nous pouvons dire que le robot n'apprend pas vraiment la topologie de l'espace navigable, car le REA n'est en réalité qu'une discrétisation de l'espace métrique en secteurs, les liens entre secteurs

correspondant à segmenter la trajectoire suivie en morceaux de lignes droites.

De nombreux projets de recherche récents ont exploité un carte topologique *a priori* d'un milieu structuré pour aider un robot à naviguer. Un premier exemple est le travail de Sgouros, Papakonstantinou et Tsanakas (1996). À partir d'un graphe topologique d'un ensemble de salles de bureau, leur système génère une deuxième carte qui prédit les changements qualitatifs à observer dans les mesures sensorielles (*e.g.* distance qui augmente ou diminue) entre cases voisines dans l'espace cartésien discrétisé. La carte topologique initiale sert à la planification de chemin, tandis que la deuxième carte sert à localiser le véhicule en conjonction avec l'odométrie. Simmons et Koenig (1995) adoptent une approche probabiliste markovienne pour la modélisation d'environnement (de corridors) et des processus décisionnels dans la navigation orientée-but. La carte initiale est un graphe schématisant les corridors et intersections à traverser. Par dessus ce squelette on dresse un modèle en chaînes d'états markoviens ; l'espace navigable est discrétisé en unités de translation et les orientations possibles du robot en quatre valeurs orthogonales. Les actions reliant les états entre eux sont des translations et rotations du véhicule. Pendant la navigation, la localisation du robot prend la forme d'une distribution de probabilités sur plusieurs états. Aussi, des incertitudes dans le modèle *a priori* du milieu peuvent être traitées ; par exemple, un corridor de longueur incertaine est représenté par plusieurs chaînes de Markov parallèles de longueurs différentes.

2.1.4.3 Environnements partiellement ou non-structurés

C'est surtout dans les environnements de grande échelle, à savoir les milieux naturels extérieurs, que la modélisation topologique paraît plus avantageuse que de créer une carte géométrique. En effet, son degré d'abstraction plus élevé par rapport aux détails physiques du milieu permet de limiter la quantité d'information à représenter et de mieux gérer l'incertitude sur la localisation métrique.

L'article de Kuipers et Levitt (1988) présente la hiérarchie suivante de niveaux de

descriptions d'un environnement à grande échelle :

Sensori-moteur : interactions sensorielles immédiates avec l'environnement ;

Procédural : combinaisons de renseignements sensori-moteurs (c'est-à-dire de "vues") permettant de retrouver des trajets et des endroits particuliers ;

Topologique : inférences sur la structure du milieu : construction d'un réseau d'endroits et de chemins (graphe) ; notion de régions délimitées par ces chemins ;

Métrique : valeurs quantitatives (position, orientation absolue) ou relatives (distance, direction) rattachées aux entités topologiques (endroits, chemins, régions).

On présente ensuite quelques modèles pour l'exploration de milieux et la navigation en les situant par rapport à la hiérarchie ci-dessus. L'un de ces travaux est le modèle d'exploration Qualnav de Levitt et Lawton (1990). Dans Qualnav, on suppose un terrain ouvert, naturel, où sont repérables des événements perceptuels, appelés *landmarks*. Le robot possède un senseur visuel capable de fournir des vues panoramiques omnidirectionnelles. La notion centrale de *viewframe* dénote un lieu défini par une série de landmarks détectés avec leurs distances et orientations relatives à l'agent (niveau de description sensori-moteur). Au niveau métrique, la navigation repose sur les viewframes. Chaque paire de landmarks dans le viewframe induit une région dans le plan cartésien du sol où le robot doit se trouver. L'intersection des régions pour toutes les paires de landmarks constitue la localisation du viewframe. De plus, lorsque deux viewframes partagent un même landmark, le robot peut calculer une *direction* reliant les localisations des deux viewframes. La planification et le suivi de chemin à ce niveau de description consistent alors à relier des points de départ et d'arrivée par une succession de directions reliant des viewframes partageant deux à deux des landmarks.

Au niveau topologique, on supprime l'information de distances aux landmarks dans les viewframes, pour ne conserver que l'ordonnancement des landmarks perçus à un endroit. On procède alors à la division du sol en régions dont les frontières sont les droites perceptuelles passant par toutes les paires de landmarks. En reliant en réseau toutes ces régions, on obtient un graphe par rapport auquel l'agent peut se situer en ayant des renseignements purement qualitatifs sur les landmarks autour de lui. Les algorithmes développés dans le modèle Qualnav sont démontés expérimentalement à l'aide d'un simulateur qui incorpore un terrain de relief accentué et la vision artificielle.

2.2 Navigation autonome en mines souterraines

2.2.1 Automatisation d'une chargeuse-navette

L'automatisation d'une chargeuse-navette (LHD) dans des galeries souterraines a constitué le sujet de plusieurs projets de recherche et développement. Nous pouvons distinguer deux approches parmi les solutions trouvées à l'aspect navigation du LHD : la navigation *guidée* et la navigation *non-guidée*. Dans le premier cas, on introduit dans l'environnement une ou des structures qui définissent les chemins à suivre par le véhicule. Dans le deuxième cas, on ne modifie pas l'environnement au profit du véhicule qui doit s'y déplacer ; il doit localiser le chemin à suivre à l'aide de structures existant naturellement dans l'environnement.

Dans Saint-Amant, Piché et Hurteau (1988), nous trouvons une étude comparative de plusieurs alternatives technologiques pour réaliser la navigation automatique (c'est-à-dire sans opérateur à bord) d'une chargeuse-navette. On divise le trajet typique du véhicule en trois zones : la zone de navette (qui comprend le point de déchargement), les points d'accès et la zone de chargement (où se trouve le matériau à ramasser). Dans la catégorie des systèmes *guidés*, les auteurs énumèrent plusieurs implantations possibles, qui se différencient entre elles par la nature physique du guide et par le mécanisme permettant au véhicule de le détecter et de le suivre. On trouve des

systèmes à guide "trolley", à guide par induction, à guide optique et à guide-faisceaux. Dans la catégorie des systèmes *non-guidés*, les auteurs identifient deux approches potentielles :

1. **Intégration de la trajectoire par senseurs internes :** ceux-ci sont basés soit sur le principe odométrique, soit sur le principe inertiel. Avec l'odométrie, l'incertitude de la position mesurée du véhicule croît en fonction de la distance parcourue. Avec des senseurs inertiels, l'erreur de mesure augmente avec le temps. Dans les deux cas, il faut donc recalibrer régulièrement le(s) senseurs. Ceci peut se faire à l'aide de points de repère de positions connues sur la trajectoire.

2. **Suivi de murs :** le véhicule longe une paroi verticale à distance fixe de celle-ci. Il se positionne par rapport au mur à l'aide d'un capteur télémétrique. Cette stratégie est limitée aux corridors de la mine. Dans les espaces ouverts (*e.g.* chambres) la navigation doit se baser plutôt sur des senseurs internes.

L'étude de Saint-Amant et al. (1988) débouche sur la proposition suivante d'un système hybride de navigation d'une chargeuse-navette :

1. Dans la zone de navette : système à guide par induction ou à guide optique ;

2. Aux points d'accès : suivi de mur ou système à guide-faisceaux ;

3. Dans la zone de chargement : téléopération (pas de fonctionnement autonome).

Dans la perspective de l'étude ci-dessus, un système à guide optique pour le déplacement d'une chargeuse-navette a été développé conjointement par l'École Polytechnique de Montréal et le Centre de Technologie Noranda (Hurteau, Saint-Amant, Laperrière et Chevrette, 1992; Laperrière, 1991). Le guide réfléchissant est fixé au toit de la galerie entre les points de chargement et de déchargement. Les points critiques du chemin (par exemple, là où le véhicule doit changer son angle d'articulation) sont

encodés par des rubans réfléchissants plus minces que le guide. L'appareil sensoriel consiste en une caméra CCD et une source lumineuse qui éclaire le guide. Le module de contrôle exploite les erreurs de position et d'orientation du LHD procurées par le senseur optique, pour déterminer les commandes de vitesse et d'angle d'articulation qui ramèneront le véhicule vers le guide. Le fonctionnement de ce système a été démontré dans des mines de la compagnie Noranda.

Un autre système de navigation guidée pour véhicule minier souterrain est présenté dans Kitchener (1986). La compagnie minière Inco a développé un camion à propulsion électro-hydraulique pour le transport de minerai. La navigation du camion se fait à l'aide de marqueurs placés le long des rails d'alimentation électrique posés sur le toit des galeries (système à guide "trolley"). Un détecteur est rattaché au joint reliant le camion au rail. Lorsque le détecteur est à proximité d'un marqueur, un signal radio est envoyé à un ordinateur de contrôle, qui lui renvoie les commandes appropriées pour ajuster la trajectoire du camion.

Juneau, Hurteau, Freedman et Chevrette (1993) ont étudié la navigation autonome *non-guidée* d'un véhicule minier du type LHD dans des galeries de mine. La galerie se compose de tronçons d'axe clothoidal. L'objectif pour le véhicule est de rester au milieu du tunnel en estimant la géométrie de celui-ci (largeur et courbure instantanée de l'axe), ainsi que ses erreurs de position et d'orientation par rapport à l'axe. Il se renseigne sur la distance aux parois grâce à des senseurs laser. On introduit un filtre Kalman dans la boucle de contrôle pour augmenter la stabilité de la trajectoire.

On a aussi étudié le contrôle automatique d'un LHD au Colorado School of Mines (Steele, King et Strickland, 1991; Steele, Ganesh et Kleve, 1993). Un contrôleur PID doit maintenir le véhicule à une distance fixe d'une des parois de la galerie en ajustant l'angle d'articulation au centre du LHD. La mesure de distance au mur provient d'un senseur sonar latéral. Avec ce système, le LHD parvient à suivre des parois non-rectilignes.

2.2.2 Automatisation d'un mineur en continu

Le U.S. Bureau of Mines, en collaboration avec le Field Robotics Center de l'université Carnegie Mellon et le National Bureau of Standards, a mené un projet de recherche sur l'opération automatisée d'un mineur en continu (Schnakenberg, 1988). Le système proposé est semi-automatique, c'est-à-dire que la machine doit d'elle-même avancer dans la galerie, se positionner par rapport à la paroi d'extraction, et creuser le minerai, mais son fonctionnement se déroule sous la surveillance d'un opérateur qui peut communiquer avec les ordinateurs de bord de la machine.

Le déplacement du mineur en continu dans la galerie constitue un problème de navigation autonome dans une région restreinte de la mine. Dans le mode classique, l'opérateur se base sur des marques d'arpentage observées sur les parois latérales pour positionner la machine. Cependant, ces marques ne sont pas facilement détectables par des senseurs télémétriques. Le système de guidage utilise donc comme référentiel pour la position de la machine, une structure de contrôle mobile placée proche de la paroi d'extraction (Schnakenberg, 1988). La détermination de la pose du mineur dans ce référentiel comporte deux parties : une triangulation externe de la pose utilisant deux télémètres laser placés sur la structure de contrôle qui détectent trois rétro-réflecteurs cylindriques alignés sur le dessus de la machine (Anderson, 1990), et un jeu de senseurs inertiels à bord de la machine pour le suivi à court terme de son orientation (à trois degrés de liberté) (Sammarco, 1990; Sammarco, 1988).

Shaffer et Stentz (1992) prennent une approche de navigation plus globale pour le déplacement du mineur en continu. Ici, on suppose que la machine devra traverser plusieurs couloirs (et donc franchir des intersections) avant d'arriver à la paroi d'extraction. Elle dispose d'un télémètre laser à balayage et de senseurs odométriques. Pour exécuter la trajectoire (qui est prédéterminée), le mineur corrige l'erreur de position accumulée par l'odométrie grâce à une carte *a priori* de l'environnement et aux mesures provenant du capteur laser. La mise en correspondance des balayages omnidirectionnels télémétriques avec la carte (employant une technique développée

par Gonzalez, Stentz et Ollero (1992)), permet de corriger la pose initiale provenant de l'odométrie.

2.2.3 Autres plate-formes mobiles pour le milieu souterrain

On a développé conjointement au U.S. Bureau of Mines et au Field Robotics Center de C.M.U., une plate-forme mobile appelé le "Locomotion Emulator", destiné notamment pour les recherches en navigation autonome en milieu minier (Schnakenberg, 1988; Shaffer, Stentz, Whittaker et Fitzpatrick, 1992). La partie locomotrice de cette plate-forme est capable de bouger de manière généralisée sur une surface plane. Une interface de logiciel permet de commander le mouvement de la plate-forme selon plusieurs configurations de navigation, de manière à imiter les caractéristiques de différents véhicules.

L'article de Shaffer et al. (1992) décrit une méthode de localisation d'une plate-forme mobile (testée sur le Locomotion Emulator) dans un milieu de galeries de mines souterraines. Dans ce problème, le robot est muni d'odométrie et d'un capteur laser à balayage. L'environnement est supposé connu initialement sous forme d'une carte 2D qui correspond à la projection de la mine (dont on néglige les variations en hauteur) sur le plan cartésien du sol. Le télémètre saisit des profils omnidirectionnels denses, que l'on traite pour identifier des structures saillantes dans l'environnement. On distingue deux types de structures : les segments et les coins. Le processus de localisation exploite la carte pour prédire les structures susceptibles d'être visibles et mettre celles-ci en correspondance avec les structures détectées dans le profil laser. Une faiblesse de cette approche est que la notion de "coins" n'est pas toujours applicable aux mines réelles, où les parois aux intersections sont de forme arrondie.

Un autre prototype de véhicule autonome destiné à naviguer dans des environnements miniers souterrains est le "Autonomous Navigation Testbed" de l'université de West Virginia (Banta et Rawson, 1994; Banta, Nutter et Xia, 1992). On combine les principes du contrôle hiérarchique avec ceux du contrôle réactif pour développer

un système de navigation qu'on peut qualifier de "multi-mode". Un module *pilote* est capable d'opérer en plusieurs modes : suivi de mur, virage, suivi de convoi, etc. Un module *navigateur* calcule la position du robot par rapport à une carte routière en utilisant l'odométrie le long des couloirs et les points de repère (intersections ou autres) lorsqu'ils se présentent. Un troisième module, le *chercheur de landmarks*, analyse les données provenant d'une caméra vidéo et de plusieurs télémètres pour détecter et caractériser les points de repères. À un niveau de raisonnement plus élevé, un module de gestion de mission possède la liste de commandes que doit exécuter le robot, et coordonne les actions des modules sous-jacents. Expérimentalement, on ne présente des résultats que pour certains des processus de bas niveau.

2.3 Techniques d'arpentage de mines

L'arpentage dans les mines souterraines consiste en la mesure systématique et rigoureuse des positions de points saillants décrivant les excavations. Les données produites par l'arpentage sont utilisées pour construire une carte de ces excavations. Nous nous intéressons ici aux aspects principaux de l'arpentage traditionnel faite par des équipes d'arpenteurs dans les mines (Williams, 1983). Notre étude reste générale et ne couvre pas toutes les techniques qui peuvent exister.

2.3.1 Appareils de mesure

L'appareil de mesure de distances le plus élémentaire pour les arpenteurs est le ruban métallique gradué; celui-ci est limité en général à une trentaine de mètres. Il existe cependant aujourd'hui des appareils électroniques de mesure de distance (télémètres) basés sur le principe des ondes électromagnétiques. Ce genre d'appareil requiert une calibration initiale, mais fournit des mesures plus précises et sur des distances bien plus importantes. Le théodolite est l'appareil par excellence pour mesurer des angles horizontaux et verticaux. Il est muni d'un télescope et de plateaux de rotation en

horizontale et verticale, et peut être mis à niveau une fois positionné. Le bâton de pointage est une grande règle graduée qui, employée avec le théodolite, sert à mesurer des hauteurs relatives (*cf.* section 2.3.3).

2.3.2 La mesure des positions horizontales

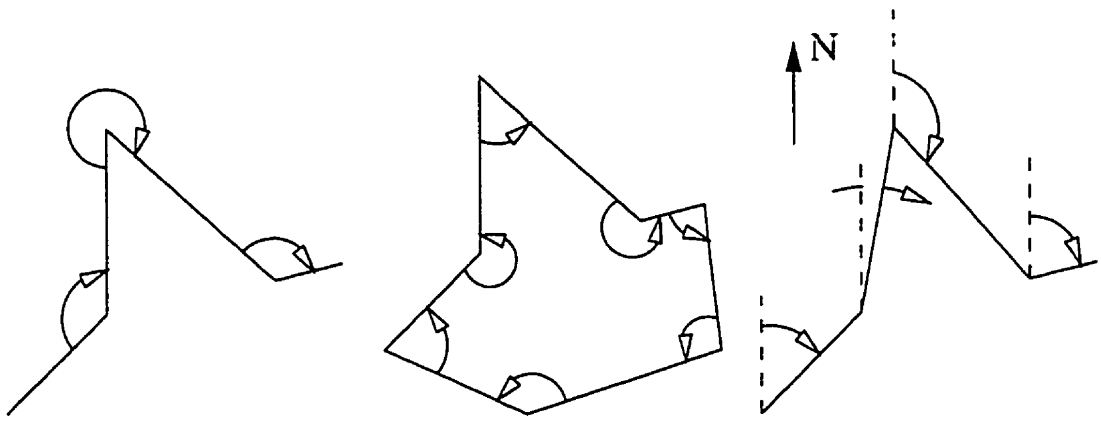
Le plus souvent, la région où se situe la mine est cartographiée et de ce fait dotée d'un réseau de références de positions géographiques connues avec précision. La première étape dans l'arpentage est d'établir un point de référence de position absolue connue sur le terrain de la mine, par triangulation à partir de plusieurs points géodésiques.

Dans le cas où la mine souterraine est reliée à la surface par un puits, il est nécessaire d'effectuer la "descente" d'un méridien (une droite d'orientation nord-sud) dans la mine afin d'avoir l'orientation absolue correcte des mesures souterraines. Nous reviendrons à cette étape à la section 2.3.5.

Le principe pour l'arpentage des excavations est de suivre un parcours formé de segments de droites consécutifs et de mesurer leurs longueurs et leurs orientations relatives. On appelle *station* un point de position (absolue ou relative) connue: une station est matérialisée par une pièce métallique fixée au lieu en question (généralement au plafond pour ne pas être détruite par le passage de véhicules). Les stations sont les sommets des segments formant le parcours. Ainsi, on avance incrémentalement sur le parcours, en calculant la position de chaque nouvelle station en se basant sur la position (connue) de la précédente.

Pour mesurer les *angles* entre segments on utilise un théodolite, que l'on place à un point intermédiaire appelé *point tournant.* Trois méthodes communément employées sont celles des *angles à droite*, des *angles intérieurs* et des *azimuts.* L'angle à droite (figure 2.1(a)) est mesuré en tournant le théodolite de la direction du point précédent vers celle du point suivant sur le parcours. On peut répéter cette procédure plusieurs fois et prendre la valeur angulaire moyenne pour augmenter la précision. La méthode des angles intérieurs (entre segments consécutifs) est employée lorsque le parcours est

fermé (figure 2.1(b)). La méthode des azimuts utilise un méridien connu pour obtenir des orientations absolues plutôt que relatives. L'azimut d'un segment est mesuré en tournant l'instrument de la direction nord vers celle du point suivant (figure 2.1(c)).



(a) L'angle à droite. (b) Les angles intérieurs. (c) L'azimut.

Figure 2.1 : Méthodes pour mesurer les angles en horizontale

On emploie un télémètre, placé lui aussi au point tournant, pour mesurer les *distances* entre stations. Les distances entre le point précédent et le point tournant, d'une part, et entre le point tournant et le point suivant, d'autre part, sont mesurées. Une simple relation trigonométrique donne la position du nouveau point par rapport au précédent. Lorsque l'excavation est en pente (légère), on peut mesurer aussi cette inclinaison, pour ensuite reporter la distance en pente et la distance horizontale entre deux points.

Le résultat de toutes ces opérations est un tableau résumant toutes les positions et orientations (absolues ou relatives) des stations et segments. On reporte ensuite tous les points dans un référentiel cartésien global pour pouvoir dresser la carte des excavations.

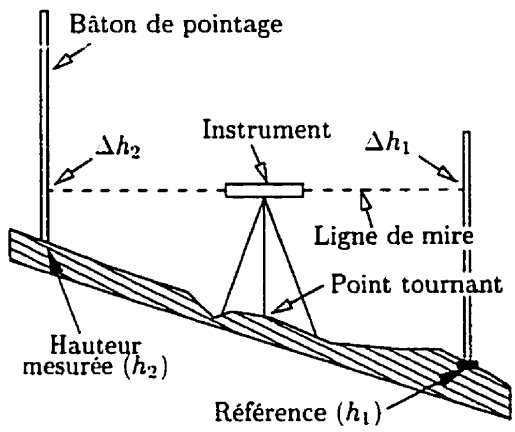
2.3.3 La mesure des hauteurs

Lorsque la section de mine à arpenter comporte des parties en pente (par exemple, les rampes d'accès à partir de la surface ou entre les garages d'équipements et la zone d'exploitation à un niveau souterrain donné), ou veut mesurer ces changements de hauteur. Comme pour l'arpentage des positions horizontales, le principe ici est de déterminer la hauteur relative d'un nouveau point par rapport au point précédent de hauteur connue (*référence*).

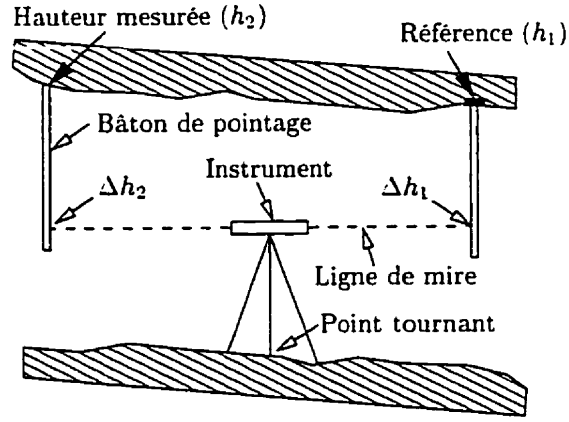
La figure 2.2(a) illustre la technique habituelle pour arpenter des hauteurs en surface. On choisit d'abord une référence; la première référence est un point géodésique d'altitude connue précisément. Ensuite, on place un théodolite à un point tournant et des bâtons de pointage sur la référence et sur un nouveau point à mesurer. Une première lecture sur le bâton de la référence permet de déterminer la hauteur de la ligne de mire de l'instrument ($h_1 + \Delta h_1$). Une deuxième lecture sur l'autre bâton donne la hauteur du nouveau point par simple différence : $h_2 = h_1 + \Delta h_1 - \Delta h_2$. Cette technique est répétée plusieurs fois pour "amener" l'altitude d'un point géodésique vers une référence sur le terrain de la mine.

Une fois dans la mine, la même technique est utilisée, mais avec cette différence que maintenant les points à mesurer sont au plafond de l'excavation (figure 2.2(b)). La différence de hauteurs à calculer est alors : $h_2 = h_1 - \Delta h_1 + \Delta h_2$. Pour connaître la hauteur du sol, on mesure aussi la hauteur de l'excavation en ces mêmes endroits.

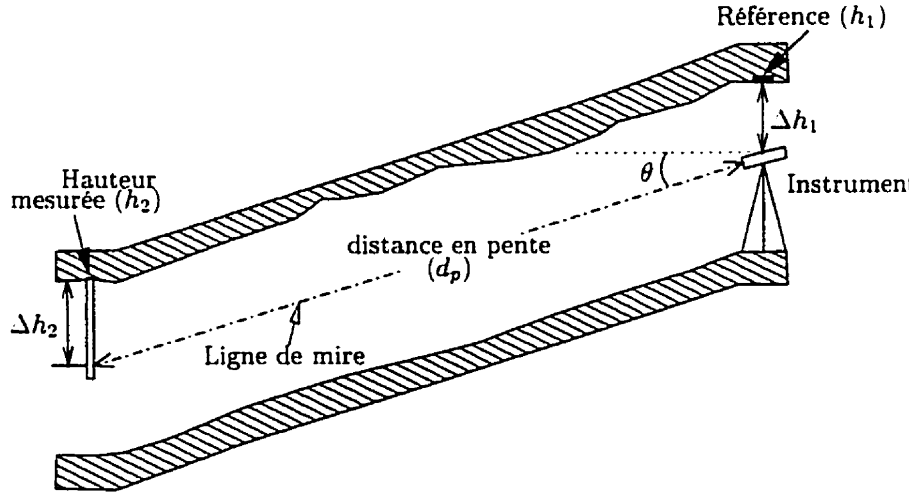
Une autre méthode pour arpenter les hauteurs, utilisée par exemple dans les rampes d'accès, est illustrée dans la figure 2.2(c). L'instrument est un théodolite surmonté d'un télémètre; on le place sous la référence. Un bâton de pointage est placé en le nouveau point. On mesure la distance le long de la pente (d_p), l'inclinaison (θ) et les dénivellations Δh_1 et Δh_2. Une relation trigonométrique donne la hauteur du nouveau point : $h_2 = h_1 - \Delta h_1 + \Delta h_2 - d_p \sin(\theta)$.



(a) À la surface ($h_2 = h_1 + \Delta h_1 - \Delta h_2$).



(b) Dans une excavation ($h_2 = h_1 - \Delta h_1 + \Delta h_2$).



(c) Dans une rampe d'accès ($h_2 = h_1 - \Delta h_1 + \Delta h_2 - d_p \sin(\theta)$).

Figure 2.2 : Techniques de mesure des hauteurs

2.3.4 Mesures latérales le long du parcours

Dans un parcours d'arpentage des positions horizontales, chaque segment de droite reliant deux stations représente l'axe central (idéal) de l'excavation en question. Le tableau des positions et orientations de la série de stations et segments ne représente que le "squelette" de la section de mine qu'il faut représenter. Les arpenteurs doivent donc compléter cette information par des mesures *latérales* décrivant les structures importantes observées le long du parcours.

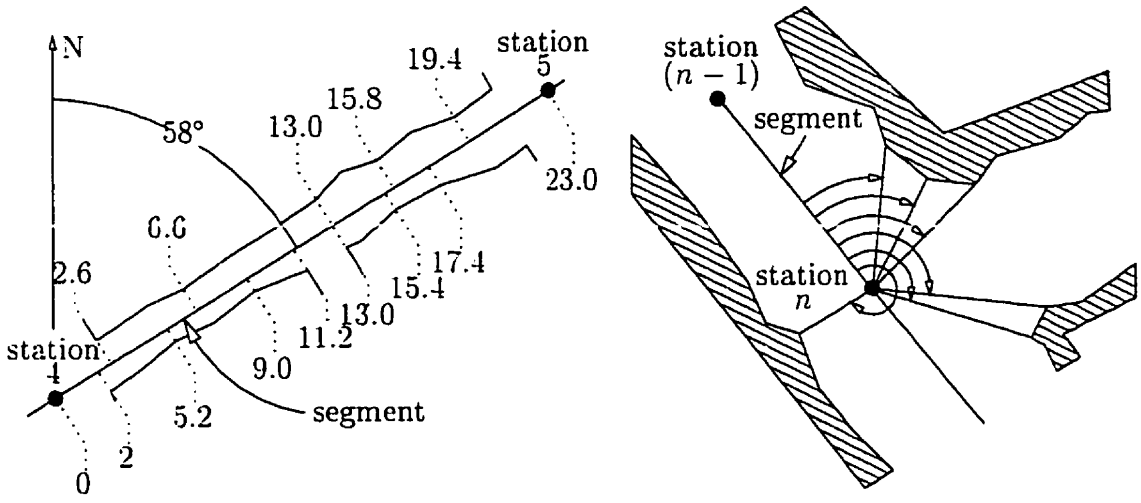
Une première manière d'effectuer ces mesures latérales est illustrée dans la figure 2.3(a). En avançant sur le segment joignant deux stations consécutives, on note toute structure importante par son type, sa distance longitudinale sur le segment et sa distance latérale (à gauche ou à droite) par rapport au segment. Les structures que l'on veut indiquer sont le plus souvent des piliers, les coins formés par l'intersection de deux galeries, ou la largeur locale de la galerie que l'on suit. Le résultat est un tableau qui identifie le segment (les deux stations) en question, puis énumère toutes les structures observées (type et positions longitudinale et latérale).

Une autre méthode consiste à mesurer les distances vers les parois dans une série de directions autour d'un point tournant ; on peut utiliser un théodolite surmonté d'un télémètre. Dans l'exemple de la figure 2.3(b), on indique une série de points saillants dans une intersection par leurs angles à droite et leurs distances du point tournant. Cette deuxième méthode est appropriée pour décrire des intersections de forme plus complexe ou bien des galeries d'axe curviligne.

2.3.5 Sources d'erreurs et leur traitement

Quel que soit le soin apporté au travail des arpenteurs, on commet inévitablement des erreurs de mesure, qui sont manifestées par des imprécisions dans la représentation de la mine lorsqu'on en construit la carte.

Une première source d'imprécision est à l'étape de la "descente" d'un méridien dans la mine. La méthode traditionnelle pour ce faire consiste à suspendre deux fils à



(a) Structures le long d'un segment du parcours. (b) Points saillants dans une intersection.

Figure 2.3 : Méthodes pour faire des mesures latérales

plomb dans le puits, allant de la surface jusqu'au niveau souterrain voulu. L'azimut (l'angle par rapport au nord) de la droite passant par les deux fils est mesuré à la surface; on suppose que cette même orientation est conservée jusqu'au fond du puits. Cependant, à cause d'oscillations possibles des deux fils, cette méthode peut introduire une erreur d'orientation absolue dans l'arpentage (qui est reflétée dans les positions absolues des points). Cependant, on a accès maintenant à des méthodes gyroscopiques pour mesurer les orientations. Grâce au gyro-théodolite, qui maintient l'orientation nord-sud avec une très bonne précision, toute orientation sous la terre peut être connue en absolue.

Un deuxième type d'erreur survient lorsque le parcours d'arpentage est fermé. Dans un tel cas, la première et la dernière stations du parcours doivent en principe être à la même position. Si une distance non-nulle entre ces deux points est indiquée lorsqu'on reporte toutes les mesures dans un référentiel commun, il s'agit d'une erreur de non-fermeture de boucle. Cette distance est comparée à la longueur du parcours.

Si l'erreur de non-fermeture est trop grande, on doit refaire l'arpentage. Sinon, on distribue l'erreur totale sur toutes les stations dans le parcours afin de "refermer" la boucle. Cette distribution est généralement uniforme, à moins que les arpenteurs aient noté des problèmes à des endroits particuliers, auquel cas une partie plus grande de l'erreur totale est reportée à ces endroits-là.

2.4 Situation de notre projet

L'idée de développer un système d'arpentage automatisé pour les mines souterraines, lequel système repose sur le déplacement d'un robot autonome, est certainement nouvelle par rapport à tout ce qui existe dans la littérature. Nous venons de voir qu'il existe, d'une part, de nombreuses approches pour l'exploration par un robot mobile d'un environnement inconnu en vue d'en construire une représentation. Mais aucun de ces systèmes n'est appliqué à l'environnement minier souterrain, que nous qualifions de milieu de corridors partiellement structuré.

D'autre part, dans le domaine de l'automatisation minière, nous avons décrit plusieurs systèmes pour l'automatisation du déplacement d'un véhicule. Mais ces systèmes ont tous des limitations majeures par rapport aux objectifs que nous nous fixons. Certains d'entre eux dépendent de l'installation de guides, chose que nous voulons éviter car cette opération est longue, coûteuse et doit être répétée régulièrement. D'autres de ces systèmes sont restreints à des trajets simples dans la mine, par exemple le suivi d'une seule galerie, et ne s'adressent pas au problème de la navigation à une échelle plus globale. Quelques auteurs (Banta et al., 1992; Shaffer et al., 1992; Steele et al., 1991) proposent des méthodes pour la navigation selon des trajets plus complexes composés de plusieurs galeries, intersections ou chambres. Mais nous n'y trouvons point de solution vraiment complète ou robuste aux problèmes concourants du contrôle de véhicule, de la navigation globale et de la représentation du milieu. Par ailleurs, aucun des systèmes de navigation en mines souterraines que

nous avons vus n'est adapté à la tâche de l'arpentage.

Dans le contexte de l'arpentage de mines, nous remarquons d'abord que notre système n'utilise ni les mêmes appareils de mesure, ni les mêmes techniques que dans l'arpentage traditionnel. Notre "robot-arpenteur" ne produit pas non-plus ses mesures dans le même format que les arpenteurs humains. Mais l'objectif est bien sûr identique : fournir un ensemble de mesures décrivant les excavations parcourues, mesures qui servent ensuite à dresser une représentation géométrique de ces mêmes excavations. Par rapport à l'arpentage traditionnel, notre robot récolte des mesures beaucoup plus *denses* sur la forme des parois afin de pallier à son "manque d'expérience" comparé à l'arpenteur humain. En effet, ce dernier sait quelles sont les mesures importantes à faire et peut donc fournir une description plus éparse de la forme des excavations.

Il faut noter que dans l'élaboration de notre approche, nous ne tenons pas compte des dénivellations ou parties en pente de la mine (*e.g.* les rampes entre niveaux différents). Ceci veut dire que les informations récoltées décrivent la projection de la mine dans un plan horizontal. Il faudrait alors intégrer à notre système un senseur tel qu'un inclinomètre afin de tenir compte des changements de hauteur. Par ailleurs, nous avons développé notre approche dans l'optique d'arpenter des zones de transport dans la mine plutôt que des zones d'extraction du minerai où les excavations sont beaucoup plus denses (*e.g.* des zones de chambres et piliers). Cependant, la géométrie des zones d'extraction est forcément hautement variable au cours du temps et peut donc être représentée avec une précision moindre que les galeries de transport. Nous pouvons donc envisager d'arpenter ces zones de la mine au moyen d'un ensemble de télémètres à balayage placés à des endroits fixes, plutôt qu'avec un robot mobile.

Chapitre 3

Définition du problème

3.1 Objectifs du système

3.1.1 Arpentage

L'objet de notre projet est le développement d'une plate-forme mobile (robot) capable de faire l'arpentage de mines souterraines. L'objectif primaire de notre système est donc de récolter des données télémétriques décrivant la forme des parois d'un réseau minier souterrain. Ce premier aspect de notre projet s'accorde avec un projet de recherche connexe, celui de la modélisation d'objets (*cf.* Labrosse, 1998), et en particulier ici la modélisation de mines souterraines sous tous leurs aspects d'opération. Selon le modèle générique (nommé GNOMine), les excavations de la mine sont décomposées en galeries, intersections de galeries et chambres (d'excavation, de rangement d'équipement et autres). Nous reviendrons dans ce chapitre à la représentation du milieu selon GNOMine.

Pour démontrer l'utilisation du modèle générique d'objets, on a développé dans le contexte du travail de Labrosse (1998) des algorithmes pour la modélisation de la géométrie des excavations. Ce processus prend comme données d'entrée des mesures télémétriques en largeur et en hauteur, supposées prises à intervalles réguliers par

une plate-forme mobile circulant à l'intérieur du réseau minier souterrain. La reconstruction géométrique suppose de plus que les données brutes sont pré-segmentées en morceaux décrivant séparément chaque galerie, intersection et chambre rencontrée. Chacun de ces *éléments topologiques* de la mine doit être spécifié dans son repère géométrique propre : celui-ci correspond naturellement à un point d'accès par lequel le véhicule est rentré dans cet élément.

Nous ferons ici une supposition qui a des conséquences importantes pour la tâche d'arpentage : le robot se déplacera *en continu* et donc ne pourra pas s'arrêter à des endroits particuliers pour accumuler des données sensorielles. Ce critère est relié au deuxième objectif de notre système, celui de la navigation autonome (voir la section suivante). Les capacités sensorielles que nous développons doivent se prêter aussi bien au contexte de la navigation entre deux points qu'à celui de l'arpentage. Face à la supposition de déplacement en continu du véhicule, notre système devra tenir compte, entre autres, du fait que lors d'un passage par une intersection ou chambre, les mesures accumulées pourront ne décrire que partiellement les parois de l'endroit. Pour avoir une description complète d'une région relativement complexe, il faut donc prévoir la possibilité de fusionner les mesures provenant de plusieurs passages par l'endroit suivant des parcours différents.

3.1.2 Navigation autonome

Le deuxième objectif de notre système, qui est évidemment relié à celui de l'arpentage automatisé, est la navigation autonome en milieu minier souterrain. Outre la fonction d'arpenter les parois de la mine, nous voulons que notre robot mobile puisse se déplacer et se localiser à l'intérieur de la mine sans intervention humaine (*e.g.* sans chauffeur à bord ni téléopérateur). Plus précisément, le robot doit être capable de suivre de manière fiable et répétable un chemin préétabli entre deux endroits de la mine. Par exemple, une application de ceci serait le transport automatisé de minerai entre des points de chargement et déchargement. Ceci dit, nous voulons

éviter que le système dépende de l'installation de structures artificielles pour guider le véhicule, comme c'est le cas pour les chargeuse-navettes automatisées dans notre revue bibliographique à la section 2.2.1. Nous voulons au contraire exploiter les structures inhérentes d'un réseau minier souterrain pour aider le robot à se repérer.

Nous ferons désormais une supposition importante : que le robot possède une graphe topologique *a priori* dans lequel les arcs symbolisent les galeries et les nœuds symbolisent les intersections de galeries et les chambres. Cette connaissance initiale partielle du milieu permet deux choses. D'une part, le chemin à suivre par le robot peut maintenant être spécifié par rapport à cette représentation globale, topologique, de la mine. Nous parlerons de *chemin global* à suivre par le robot. D'autre part, l'apport d'information par le graphe sur la topologie de la mine entraîne une réduction de la difficulté de l'analyse sensorielle à faire aux niveaux des intersections et chambres. Concrètement, si le robot sait d'avance le nombre de branches qu'il doit détecter lorsqu'il débouche dans une intersection ou chambre (équivalent au degré du prochain nœud dans le chemin global), ceci l'aidera à interpréter correctement les données structurales qu'il perçoit. Il aura beaucoup moins de chances de mal identifier les branches et de se perdre par la suite dans la mine.

Il y a une raison supplémentaire justifiant l'introduction du graphe topologique, reliée à la tâche de localiser le véhicule dans la mine. Nous avons vu au chapitre 2 que dans l'exploration d'un environnement complètement inconnu au départ, le robot est continuellement confronté au problème de distinguer les lieux rencontrés ayant des "signatures sensorielles" semblables. C'est pourquoi par exemple le système de Kuipers et Byun (1987) introduit une procédure de confirmation durant laquelle le robot doit examiner le voisinage de l'endroit d'identité ambiguë. Dans notre cas, par contre, nous évitons ce genre de complication : pourvu que le robot reconnaisse systématiquement les passages d'un élément topologique à un autre, il pourra suivre sa position dans le graphe. Le problème de distinguer globalement tous les endroits rencontrés est remplacé, grâce au graphe *a priori*, par la tâche moins complexe de

reconnaître les *transitions* entre régions de la mine.

Il faut remarquer cependant que tout ceci suppose que l'on ait *construit* au préalable le graphe que le robot exploitera et donc que l'on ait établi des règles générales pour décomposer la mine en éléments topologiques. Ces règles feront intervenir d'une part les catégories de régions pouvant exister dans la mine et d'autre part les capacités perceptuelles du robot. Nous reviendrons à cette question de la modélisation topologique d'une mine à la section 3.4.2 ainsi qu'au chapitre 5.

3.2 Modèle géométrique de la mine

3.2.1 Modèle de mine pour la navigation

Nous nous basons sur le modèle générique de mines souterraines, GNOMine, pour l'interprétation de l'environnement et pour la formulation du graphe topologique *a priori*. Le formalisme complet de GNOMine est présenté dans Labrosse, Hervé et Cohen (1996). Cependant, il est nécessaire d'en faire ici un résumé pour la compréhension de la suite de notre démarche.

Le modèle GNOMine fait la distinction entre des régions telles que galeries, chambres et intersections, car la tâche de navigation n'est pas la même dans ces différentes régions. La mine est représentée par plusieurs *attributs*, dont celui qui nous intéresse en premier lieu est la géométrie des parois. Dans la définition de l'attribut géométrie, la mine se décompose en plusieurs types d'objets géométriques que nous définissons comme suit :

GALERIE : une série de tronçons raccordés l'un à la suite de l'autre. La galerie peut varier dans ses dimensions latérales (hauteur, largeur), ou dans l'évolution de sa courbure (par exemple, passage d'axe droit à circulaire). Une galerie relie entre elles des intersections et/ou des chambres. Un référentiel est rattaché à chaque extrémité de la galerie.

TRONÇON DE GALERIE : un tube de section approximativement rectangulaire, de largeur constante et dont l'axe peut être caractérisé par une équation de clothoïde. Le tube est caractérisé aussi par sa largeur, hauteur et longueur.

INTERSECTION : un ensemble de murs, plus un toit et un sol, qui relient n galeries à leurs extrémités (avec $n \geq 2$). Un référentiel d'intersection est rattaché à chaque extrémité de galerie incidente. Des transformations géométriques existent entre chaque référentiel et les $n-1$ autres. Les dimensions d'une intersection sont à l'échelle d'un véhicule minier.

CHAMBRE : une chambre se différencie d'une intersection par sa dimension (échelle de plusieurs véhicules) et par la forme des murs (qui sont concaves plutôt que convexes, à partir d'un point d'observation à l'intérieur de la chambre).

3.2.2 Milieu minier simulé

Afin d'évaluer nos stratégies de navigation et d'analyse sensorielle, il faut réaliser l'interaction entre trois choses :

1. l'environnement (en l'occurrence, le milieu minier souterrain) ;
2. le véhicule ;
3. les senseurs.

Cette interaction peut se dérouler soit dans le monde réel, soit dans un milieu de simulation. Étant donné que notre objectif expérimental est de démontrer le concept d'un système d'arpentage automatisé et non pas de réaliser un tel système physiquement, nous optons pour le deuxième choix : la navigation se déroule dans un environnement minier *simulé*. Ainsi, les essais peuvent se réaliser de manière sécuritaire, répétable et beaucoup moins coûteuse que si l'on opérait avec un vrai robot dans une vraie mine. Aussi, nous pourrons contrôler tous les paramètres de l'environnement qui influent sur la complexité du problème de navigation.

Nous avons donc développé un banc d'essai en simulation qui incorpore le modèle du milieu de la section 3.2.1 ainsi que des modèles du véhicule et de ses senseurs. Ce travail est documenté en détail dans un rapport technique (Debanné, 1997). Les principales étapes dans la réalisation du banc d'essai sont passées en revue dans l'annexe A.

3.3 Modalités sensorielles du robot

3.3.1 Comparaison des principes sensoriels de télémétrie

Afin de remplir les deux tâches de navigation autonome et d'arpentage des parois identifiées à la section 3.1, notre robot doit porter des senseurs qui le renseignent sur la structure environnante. En général, les systèmes de robotique mobile autonome utilisent comme senseurs de distances l'un des trois principes sensoriels suivants :

1. télémétrie sonar ;
2. télémétrie à lumière structurée ;
3. télémétrie laser à balayage.

Dans le premier cas, un émetteur-récepteur acoustique mesure la distance vers un objet comme fonction linéaire du temps de propagation (transmission et réflexion) d'un faisceau de forme conique. Le télémètre à lumière structurée est réalisé par le couplage d'un projecteur de plan lumineux (par diffraction d'un faisceau laser) et d'une caméra CCD qui capte les points d'intersection du plan avec la scène ; la position des points détectés dans l'image permet de calculer leur profondeur. Le troisième principe consiste en un faisceau laser balayé dans la scène par un miroir rotatif. La distance aux points interceptés est fonction du décalage de phase entre émission et réception.

Nous devons choisir un des principes ci-dessus comme modèle de départ des senseurs externes du robot. Un critère de sélection primordial est celui de robustesse : un senseur monté sur une plate-forme mobile circulant dans une mine subit des chocs et vibrations importants. Pour cette raison, nous excluons la télémétrie laser à balayage car un dispositif de cette sorte comporte des éléments trop fragiles pour l'environnement minier. Il nous intéresse donc ici de comparer les deux autres principes, à savoir le sonar et la lumière structurée, selon plusieurs critères de performance.

3.3.1.1 Résolution angulaire

La résolution angulaire de mesure d'un système sonar est limitée par la largeur du faisceau acoustique, dont le lobe principal a la forme d'un cône. La résolution est de l'ordre de 15 à 20 degrés pour un émetteur-récepteur standard (Forsberg, Larsson et Wernersson, 1995; Kuc et Viard, 1991; Matthies et Elfes, 1988). À l'intérieur de ce faisceau on n'obtient qu'une seule mesure.

Le principe de la lumière structurée, *e.g.* le système BIRIS (Lang, Korba, Blais et Lecavalier, 1993; Blais, Rioux et Domey, 1991; Blais et Rioux, 1986), procure une résolution angulaire beaucoup plus élevée que le sonar. L'intersection du plan lumineux avec la scène, appelé *profil*, est discrétisée en un nombre de points égal à la résolution-image perpendiculaire au profil. Par exemple, le senseur BIRIS décrit dans Lang et al. (1993) calcule un profil de 120 points ayant un écartement angulaire de 20 degrés. D'autres systèmes du même type peuvent produire des profils d'écartement angulaire atteignant 90 degrés et de résolution dépassant 400 points.

3.3.1.2 Influence de la rugosité des parois

À cause de la rugosité des parois, la section de paroi interceptée par le faisceau acoustique ou le plan laser présente des morceaux d'orientation variable par rapport à la direction du senseur. Sur chaque morceau, la surface agit comme un miroir pour la réflexion du signal (figure 3.1).

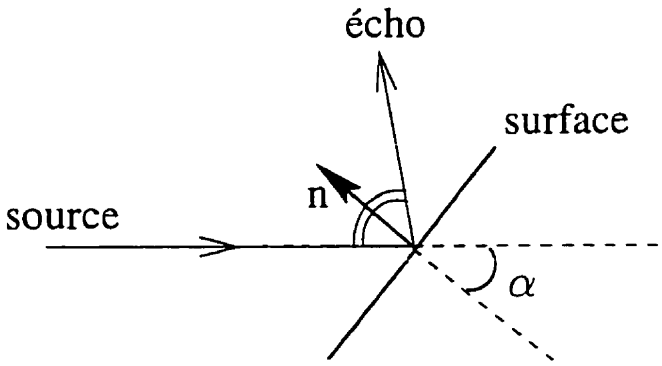


Figure 3.1 : réflexion du signal transmis

Pour le senseur à lumière structurée, ceci se manifeste par l'affaiblissement du signal de retour dans certaines régions de l'image. Les points de profil correspondants deviennent erronés ou non-détectables. Des tests sur le système BIRIS (Lang et al., 1993) montrent que pour l'angle α allant jusqu'à 45 degrés, le taux de détection demeure supérieur à 90%. Si $\alpha > 45$ degrés le taux de détection décroît rapidement.

A l'intérieur du cône émis par le sonar, la surface interceptée va renvoyer l'énergie acoustique dans plusieurs directions. Au total, il y aura un écho détectable par le senseur, sauf en cas dégénéré.

3.3.1.3 Portée maximale

Les deux types de senseurs que nous considérons sont physiquement limités en portée, la cause principale étant la dispersion de l'énergie émise. Dans le cas du sonar, Forsberg et al. (1995) cite la limite d'utilisation du sonar comme étant de 5 mètres. Si l'émetteur-récepteur comporte un amplificateur à gain variable pour compenser l'effet de dispersion du signal, la distance détectable peut dépasser 10 mètres (Kuc et Viard, 1991).

La projection d'un plan de lumière est susceptible non seulement au facteur géométrique de dispersion, mais à d'autres facteurs tels que l'absorption atmosphé-

rique et les propriétés de réflexion des surfaces dans la scène. Dans le cas de BIRIS (Lang et al., 1993), la limite d'utilisation est typiquement d'environ 5 mètres. Cependant, nous raisonnons qu'en augmentant la puissance du faisceau laser et en réduisant le champ visuel de la caméra dans la direction d'échantillonnage du profil, il est possible de construire et d'étalonner un télémètre fonctionnant jusqu'à 10 voire même 20 mètres.

3.3.1.4 Précision de mesure

La précision de mesure d'un télémètre sonar est relativement bonne. Elle dépend surtout des circuits de chronométrie qui évaluent le laps de temps écoulé avant de recevoir l'écho. Pour un dispositif standard, on évalue l'incertitude de distance à entre 1 et 3 centimètres, ou de l'ordre de 1% de la limite supérieure mesurable (Elfes, 1987; Matthies et Elfes, 1988; Forsberg et al., 1995). Cependant, les mesures procurées par le sonar sont souvent erronées par l'effet des réflexions multiples du signal, surtout dans le cas des surfaces spéculaires. Ceci constitue un inconvénient majeur du sonar comparé au senseur à lumière structurée.

Dans un télémètre à lumière structurée, la profondeur calculée d'un point résulte d'une triangulation de la position de ce point dans l'image. Le modèle du couplage projecteur-caméra est tel que l'incertitude de mesure Δz augmente proportionnellement au carré de la distance z (Blais et al., 1988; Blais et Rioux, 1986). Les résultats statistiques pour BIRIS (Lang et al., 1993) montrent que dans la gamme $0-3$ mètres, l'incertitude ne dépasse pas 2,5% de z. En ayant un système adapté et étalonné pour une gamme de profondeur plus élevée (z_{max} = 10 ou 20 mètres), il devrait toujours être possible de maintenir une erreur relative de 2,5% de z.

3.3.1.5 Conclusion

Le tableau 3.1 résume notre comparaison des deux principes sensoriels selon les quatre critères considérés : résolution angulaire, portée maximale, précision de mesure et

influence de la rugosité.

Tableau 3.1 : Comparaison des 2 principes sensoriels

	Résolution angulaire	Influence de la rugosité	Portée maximale	Précision de mesure	Taux d'erreur
Sonar	faible	robuste	$z_{max} = 10\text{m}$	$\Delta z/z_{max} = 1\%$	élevé
Lumière structurée	élevée	sensible	$z_{max} = 20\text{m}$	$\Delta z/z_{max} = 2{,}5\%$	bas

Il est clair que le télémètre à lumière structurée répond mieux aux besoins d'exploration et d'arpentage que le sonar. C'est donc ce principe sensoriel que nous retenons pour le robot. L'inconvénient majeur de ce choix est la susceptibilité du profil laser aux aspérités des parois. Pour pallier à ce problème, nous allons sous-échantillonner la résolution totale du profil en regroupant plusieurs mesures contiguës dans une "plage" (figure 3.2). A l'intérieur de chaque plage, nous appliquerons un filtrage des mesures individuelles. Ainsi, la probabilité d'avoir une mesure correcte à l'intérieur d'une plage sera supérieure à celle d'un seul point.

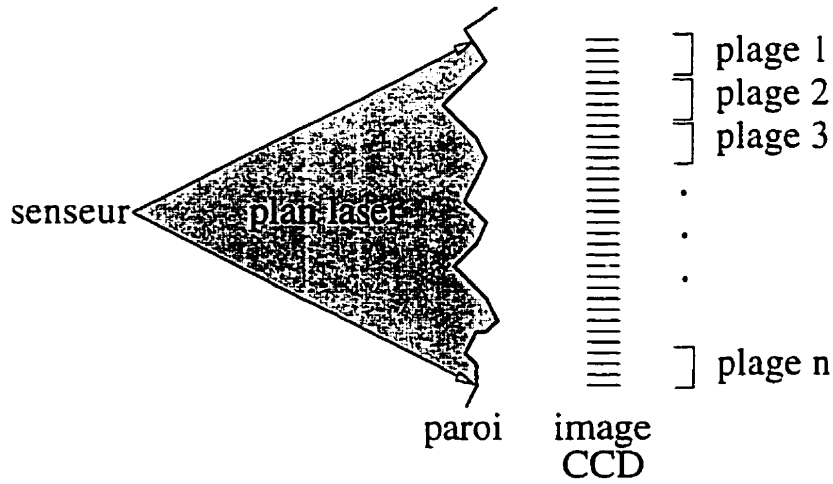


Figure 3.2 : Sous-échantillonnage du profil

3.3.2 Contraintes de l'arpentage des parois

Comme nous l'avons vu à la section 3.1.1, l'arpentage consiste à produire des mesures télémétriques des parois de la mine dans un format exploitable pour la reconstruction d'un modèle géométrique de la mine (Labrosse, 1998). Ceci implique qu'il faut associer les mesures de distance aux parois, qui sont dans un repère lié à la plate-forme mobile, à des informations sur le *déplacement* de cette plate-forme.

3.3.2.1 Senseur interne de position

Le robot doit maintenir une estimation de sa *pose* (position et orientation) par rapport à un référentiel externe associé à un point fixe dans la mine.

Deux principes sensoriels communément employés en robotique mobile s'apparentent à ce genre de mesure :

1. l'odométrie, où des encodeurs mesurent les rotations des roues ou d'un essieu ;
2. le senseur inertiel, qui mesure les accélérations angulaires instantanées du véhicule.

Dans les deux cas, l'estimation de la pose du véhicule résulte de l'intégration au cours du temps de la mesure instantanée de son déplacement.

Du point de vue de son coût, la technologie odométrique est plus accessible que celle des senseurs inertiels, mais procure moins de précision. Les effets de glissement des roues font rapidement décroître l'exactitude de l'estimation de pose. Par conséquent, tout système de robotique mobile utilisant l'odométrie est obligé de corriger régulièrement l'erreur de pose accumulée au moyen de points de repère externes connus que le robot détecte (Chatila et Laumond, 1985; Cox, 1991; Greiner et Isukapalli, 1996). Un senseur inertiel pourra fournir une estimée beaucoup plus précise, (surtout au niveau de la mesure d'orientation du véhicule) mais néanmoins entachée d'erreurs. Selon une étude expérimentale d'un système inertiel destiné à la robotique mobile (Barshan et Durrant-Whyte, 1995), le senseur inertiel en question (composé

de gyroscopes, d'accéléromètres et d'inclinomètres) ne fournit des données de position précises que pendant de courtes durées. C'est surtout la mesure de position du robot ((x, y) dans le plan du sol) qui se dégrade, car pour l'obtenir il faut intégrer deux fois les mesures instantanées des accéléromètres. Malgré une modélisation soigneuse des biais de mesures de tous les senseurs et l'introduction d'un filtre Kalman, les estimations de x et y divergent de leurs valeurs réelles à un taux variant de 5 à 30 cm/seconde. Dans le contexte de la localisation à plus longue durée, on propose alors le senseur inertiel comme système d'appoint pour une méthode de positionnement absolu utilisant des points de référence externes.

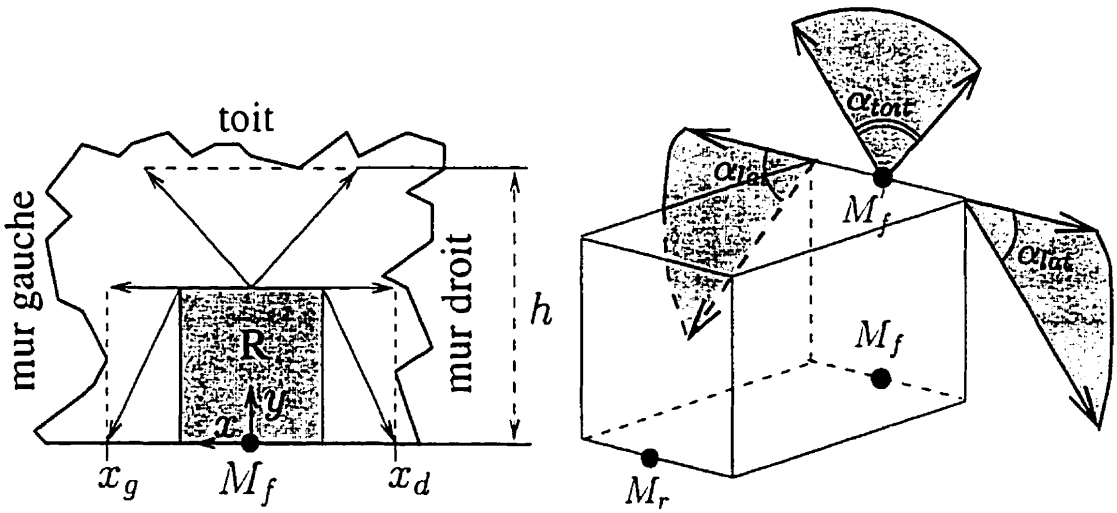
Les erreurs dans l'intégration de la trajectoire faite par le senseur interne ont une influence importante sur la complexité de la tâche d'arpentage. D'une part, la validité des modèles générés (*e.g.* longueur et forme des galeries, topologie des intersections) dégénère lorsque les données d'arpentage sont associées à un parcours mesuré qui dévie grandement du parcours véritable. D'autre part, la fusion de données d'arpentage provenant de multiples passages du robot par un même endroit devient très difficile en l'ignorance du parcours exact. Par conséquent, nous nous prévaudrons de certaines suppositions simplificatrices au niveau du modèle d'erreur en simulation expérimentale. Ces suppositions seront présentées à la section 3.4.4.

3.3.2.2 Senseurs externes

Pour mesurer la forme des parois, le robot doit à intervalles réguliers sur son parcours capter des mesures de distance sur trois parois : le mur à gauche, le mur à droite et le toit (figure 3.3(a)). Dans le but d'obtenir une représentation stable de la géométrie de la mine, nous introduisons la notion de redondance de mesures dans l'appareil sensoriel. En effet, la rugosité des parois et le bruit de mesure d'un senseur réel introduisent de l'incertitude dans les mesures latérales. Plutôt que de prendre des mesures ponctuelles, nous projetons des plans laser vers les trois parois. Les mesures retenues pour la reconstruction du modèle de mine seront pondérées sur les profils bruts, par

exemple en évaluant les distances les plus contraignantes autour du véhicule. Ceci correspond à l'idée des plages de mesures de la figure 3.2.

Nous plaçons donc trois plans laser coplanaires à l'avant du véhicule (figure 3.3(b)). Les profils des murs gauche et droite sont produits par les plans latéraux-avant, et pour obtenir le profil du toit, nous rajoutons un plan laser coplanaire des deux autres et orienté vers le toit. Les trois profils sont calculés dans un référentiel local au véhicule, celui du point M_f. Les écartements angulaires α_{lat} et α_{front} des télémètres sont établis en fonction des dimensions du véhicule et de la largeur et hauteur moyennes prévues des galeries.



(a) Distances mesurées aux 3 parois.

(b) Plans laser latéraux et du toit.

Figure 3.3 : Configuration sensorielle pour l'arpentage

3.3.3 Contraintes de la navigation globale

La navigation globale consiste en la localisation du robot par rapport au graphe topologique de la mine. Le robot doit à tout moment reconnaître l'élément topologique dans lequel il se trouve, qui peut être une galerie, une intersection ou une chambre. Ceci implique les capacités suivantes :

1. Reconnaître qu'il débouche dans une intersection/chambre :

2. Reconnaître qu'il s'engage dans une nouvelle galerie :

3. Identifier le branchement qu'il effectue dans une intersection/chambre.

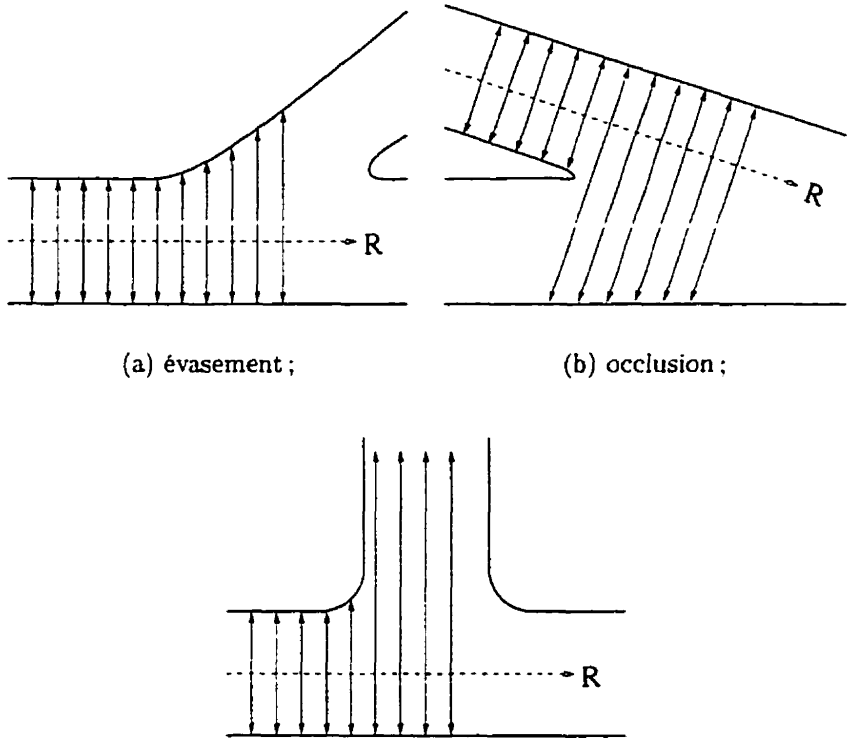
C'est par l'analyse des données sensorielles reçues que le robot identifie les *transitions de structure* énumérées ci-dessus et donc qu'il se situe par rapport au graphe topologique. Nous verrons de manière plus formelle au chapitre 5 les stratégies d'analyse sensorielle utilisées. Ici, nous proposons une configuration de télémètres pour la navigation globale.

3.3.3.1 Rajout d'un plan laser frontal

Les mesures télémétriques provenant des plans laser latéraux à l'avant du robot dans la figure 3.3 permettent de détecter certaines transitions de structure. Dans les figures 3.4(a), 3.4(b) et 3.4(c), l'arrivée de la fin d'une galerie se manifeste de trois façons : évasement, occlusion et coin de mur.

Pour détecter ces trois phénomènes, le robot peut surveiller l'évolution temporelle des mesures latérales à gauche f_g et à droite f_d (en gardant en mémoire ces mesures aux instants $t, t-1, t-2, ...$). Si f_g et/ou f_d accuse une discontinuité de t à $t+1$, il y a une occlusion ou disparition de paroi. Si f_g et/ou f_d se met à augmenter, il y a évasement des murs. Dans les deux cas, R pose l'hypothèse qu'il est sorti de la galerie.

Cependant, une fois que le robot est sorti de la galerie, ses senseurs latéraux ne le renseignent guère sur la structure de l'intersection ou chambre dans laquelle il débouche. Par exemple, il ne peut différencier l'intersection en Y et l'élargissement de galerie des figures 3.5(a) et 3.5(b) uniquement à l'aide des mesures latérales. De même, après avoir reconnu un coin de mur dans les cas de figure 3.6(a) et 3.6(b), la poursuite du chemin droit devant en l'absence d'autre information peut provoquer une collision.

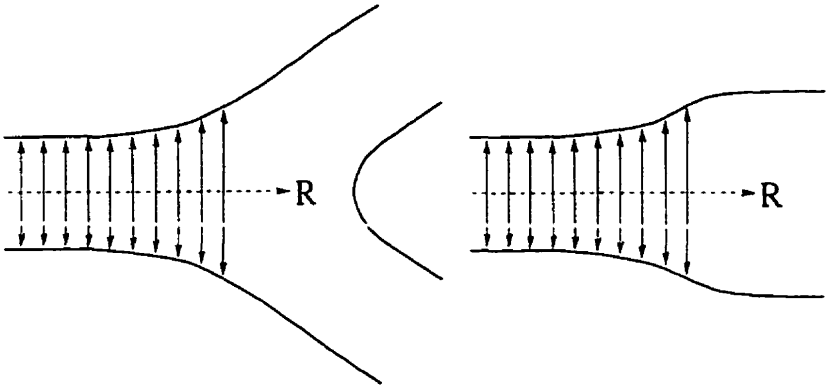


(a) évasement ; (b) occlusion ;

(c) coin de mur.

Figure 3.4 : Détection de la fin d'une galerie

Ces derniers cas de figure démontrent que le robot a besoin de percevoir l'environnement *devant* lui afin de naviguer correctement. Par conséquent, nous lui rajoutons un télémètre *frontal*, qui projette un plan laser horizontal devant le véhicule (voir la figure 3.10 plus loin). Voyons maintenant l'influence de deux caractéristiques principales du plan frontal : son écartement angulaire et sa portée maximale. Dans la discussion qui suit, les figures représentent à l'échelle un camion minier de dimensions (R_{long}, R_{larg}) = (5 mètres, 2,5 mètres) circulant dans des galeries de largeur 5 mètres.



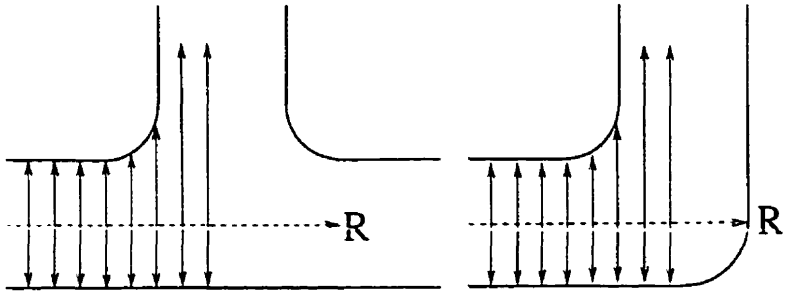
(a) intersection en Y. (b) élargissement de galerie.

Figure 3.5 : Détection d'un évasement

3.3.3.2 Influence de l'écartement angulaire

L'avantage de maximiser l'écartement angulaire du plan frontal est évident : plus ce senseur se rapproche d'être omnidirectionnel, plus le robot voit de son entourage à tout moment donné. La figure 3.7 illustre cette réalité pour trois valeurs de l'écartement angulaire frontal α_{front} : 90°, 135° et 180°. Dans cette figure, le robot débouche dans une intersection et donc il vient de détecter un élargissement (figure 3.7(a)) ou une discontinuité (figure 3.7(b)) de la largeur de galerie mesurée ($f_g + f_d$). Il doit réaliser un embranchement de la branche G_1 vers une des autres branches de l'intersection. Les deux dessins montrent ce que "voit" le robot devant lui pour les trois valeurs de α_{front}.

Dans l'intersection en Y, la valeur $\alpha_{front} = 180°$ suffit pour que le profil frontal englobe toute la structure des parois gauche et droite des branches de sortie. Mais dans la figure 3.7(b), même avec un demi-plan frontal (180°) le robot, une fois rentré dans l'intersection, ne peut percevoir qu'une des parois de G_2 et de G_4. Si nous voulions garantir que, quelle que soit la géométrie de l'intersection, le champs sensoriel du robot engloberait toutes les branches de sortie, il faudrait que le plan frontal soit



(a) intersection en T. (b) coude

Figure 3.6 : Détection d'un coin de mur.

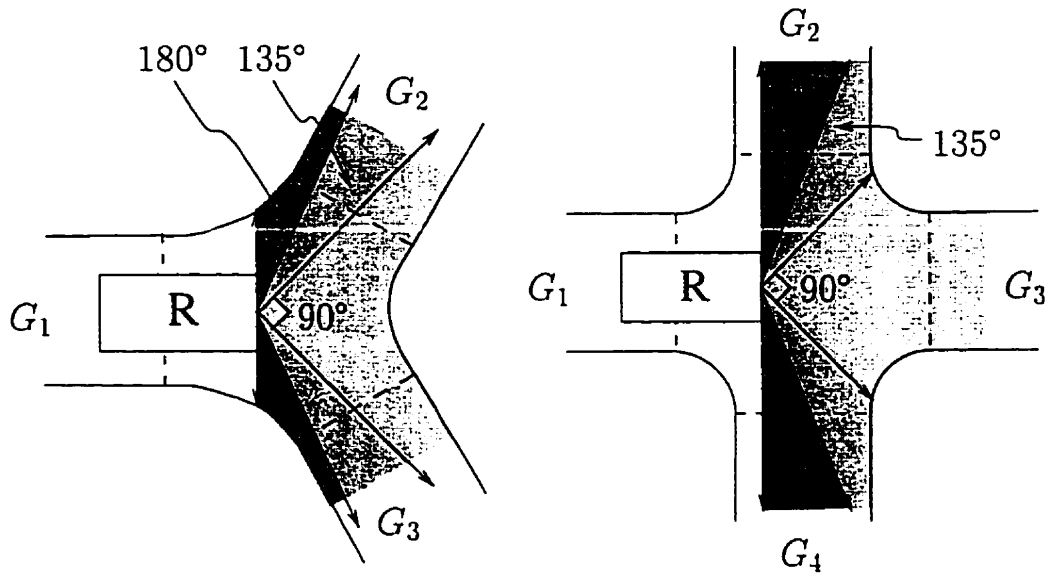
omnidirectionnel, c'est-à-dire $\alpha_{front} = 360°$.

Cependant, une question d'ordre technologique se pose : comment réaliserait-on physiquement ce senseur omnidirectionnel sous forme d'un télémètre à lumière structurée ? En effet, les systèmes de projection de lumière structurée sont généralement limités à une largeur angulaire de 90° (LASIRIS, n.d.). Il faudrait donc envisager d'employer quatre couplages caméra-projecteur, disposés à $\pi/2$ les uns des autres et balayant tout le plan horizontal. Mais la mise en commun de plusieurs télémètres de cette manière entraînerait des complications importantes aux niveaux de la réalisation technique et du traitement des données.

Nous devons par conséquent viser un compromis entre les difficultés de la tâche de navigation d'une part, et de la réalisation physique du senseur frontal d'autre part. Compte tenu de ces deux contraintes, nous prévoyons que le robot soit doté d'un *demi-plan frontal*, donc $\alpha_{front} = 180°$.

3.3.3.3 Influence de la portée maximale

Les capacités d'analyse perceptuelle et la stratégie de contrôle du robot dans une intersection ou chambre vont dépendre de la distance maximale d_{max} détectable par le senseur frontal par rapport aux dimensions de l'endroit. La figure 3.8 illustre deux



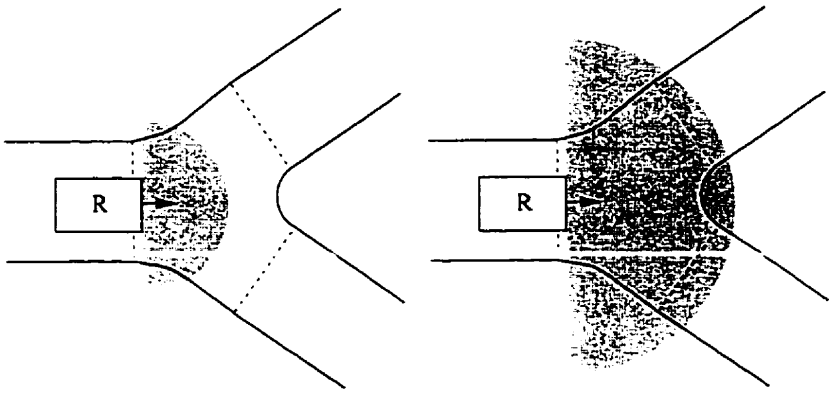
(a) intersection en Y. (b) Intersection en X.

Figure 3.7 : Influence de α_{front}

situations différentes.

Dans 3.8(a), le robot ne "voit" pas les deux branches à l'autre bout de l'intersection en Y. Il est obligé de faire du suivi de mur, jusqu'à ce qu'il arrive dans une des autres branches. Par contre, dans 3.8(b) le robot voit toute l'intersection et peut analyser le profil sensoriel frontal pour détecter les autres branches. Il peut alors choisir une des branches et planifier un virage.

Nous avons vu à la section 3.3.1 qu'un télémètre à lumière structurée est assujetti à une limite physique de sa portée, limite que nous évaluons à environ 20 mètres. En revanche, le robot devra explorer des intersections dont la plus grande dimension atteint 15 à 20 mètres dans des mines réelles. Le robot se trouvera donc généralement dans la situation de la figure 3.8(b).



(a) R est "myope". (b) R voit plus loin.

Figure 3.8 : Influence de d_{max}

3.3.4 Contraintes de la navigation locale

Par "navigation locale" nous désignons le déplacement autonome du robot dans la mine en l'absence de trajectoire précise ou d'information métrique préalable. Notre raisonnement ici a pour but non pas d'élaborer un module de navigation locale, mais plutôt de déterminer si la configuration de senseurs élaborée jusqu'ici est apte à fournir l'information structurale requise pour un tel module.

3.3.4.1 Contrôle pour le suivi de galerie

Plusieurs projets de recherche se sont penché sur la navigation autonome en milieu minier souterrain. Pour ce qui est du problème plus précis de la navigation dans une galerie de mine, nous dégageons d'une étude des méthodes existantes deux approches possibles :

- le suivi d'un des murs ;
- le suivi de l'axe de la galerie.

Juneau et al. (1993) contrôlent un véhicule de type chargeuse-navette pour le maintenir dans l'axe d'un couloir composé de tronçons clothoïdaux. On modélise complète-

ment la cinématique et la dynamique de premier ordre du véhicule. On modélise aussi le couloir, par sa largeur et par une équation linéaire régissant la courbure instantanée de l'axe. Un filtre de Kalman combine, à chaque instant de contrôle, deux mesures télémétriques relevées sur une des parois et la mesure de l'angle d'articulation pour estimer les paramètres géométriques du couloir ainsi que les erreurs de position latérale et d'orientation du véhicule par rapport à l'axe. Il s'agit alors pour le contrôleur de générer des commandes de navigation pour réduire ces erreurs.

Steele et al. (1993) ont aussi étudié le contrôle automatique d'une chargeuse-navette dans des galeries de mine. Dans ce cas-ci, le véhicule doit rester à une distance fixe d'une des parois verticales tandis qu'il avance dans des couloirs non-rectilignes. Dans le modèle, des équations relient le moment appliqué à l'articulation, l'angle d'articulation, la direction-avant de la chargeuse-navette par rapport à l'axe du couloir (supposé localement droit) et la distance latérale au mur mesurée par un senseur sonar. Un contrôleur PID doit maintenir le véhicule (qui avance à vitesse constante) à la distance voulue du mur en ajustant l'angle d'articulation.

Ces deux approches parviennent à contrôler un véhicule de manière stable dans des galeries d'axe curviligne, en se servant d'un nombre très réduit de mesures télémétriques. Par contre, on atteint cette performance au coût de la modélisation d'un type particulier de véhicule (en l'occurrence la chargeuse-navette).

3.3.4.2 Suivi de galerie dans notre cas

Des deux approches vues précédemment, c'est celle de "suivi de l'axe" qui serait la mieux adaptée pour contrôler le véhicule dans notre cas. En effet, il est souhaitable de maintenir le robot le plus proche possible du milieu des deux parois avec sa direction tangente à l'axe, et ce de manière stable, pour trois raisons :

- le robot pourra mieux détecter les événements structuraux en début ou fin de galerie ;
- s'il est mal orienté dans la galerie, le robot peut être mené à détecter incorrecte-

ment un événement structural par l'augmentation de la largeur apparente de la galerie ;

- un senseur interne d'odométrie ou d'inertie fournira une mesure plus fiable de la pose du véhicule.

L'aspect perceptuel du suivi de galerie s'exprime comme l'estimation des erreurs instantanées de position et d'orientation (e_{lat}, e_θ) du centre de l'essieu-avant du véhicule (point M_f) par rapport à l'axe. L'aspect contrôle consiste à envoyer des commandes d'accélération et de braquage au véhicule pour réduire (e_{lat}, e_θ). La figure 3.9 illustre la situation d'un tronçon de galerie droit.

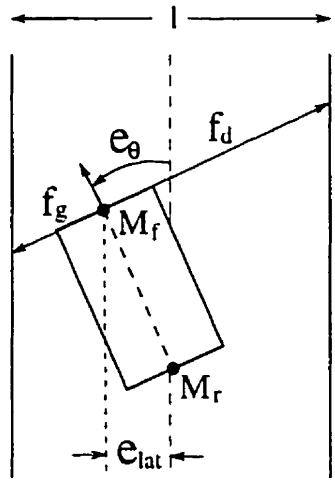


Figure 3.9 : Navigation dans un tronçon droit

Les plans latéraux-avant (prévues pour l'arpentage) fournissent deux mesures latérales, (f_g, f_d), qui sont relativement fiables grâce au facteur de redondance verticale. De plus, le plan frontal (prévu pour la navigation globale) procure de l'information sur la forme des parois plus loin devant le véhicule. Il faut noter que les mesures frontales sont moins précises que f_g et f_d, pour deux raisons :

- le télémètre frontal détecte généralement des parois plus éloignées, et donc avec une précision de mesure moindre, que les télémètres latéraux ;

- il n'y a pas de redondance verticale des points du profil frontal pour confirmer leurs positions dans le plan du sol.

Mais, avec des modèles d'incertitude adéquats pour les deux sources d'information, nous raisonnons que l'information frontale pourrait être intégrée aux mesures latérales de manière à améliorer la performance d'un système de contrôle. Ainsi, il semble justifié de dire que la configuration sensorielle déjà identifiée (senseur interne de position, plans laser latéraux et frontal) constitue un appareil perceptuel adéquat pour l'implantation d'un algorithme de contrôle local tel que dans Juneau et al. (1993).

3.3.5 Configuration sensorielle proposée

En récapitulation, le robot dispose d'un jeu de senseurs télémétriques illustré à la figure 3.10 pour remplir la fonction perceptuelle des tâches d'arpentage, de navigation globale et de navigation locale. De plus, il possède un senseur interne qui lui procure une mesure de sa pose dans un référentiel associé à un point fixe dans la mine.

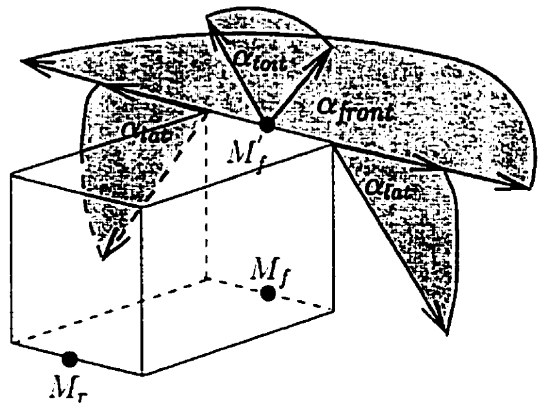


Figure 3.10 : Configuration sensorielle du robot

3.4 Présentation du système

Dans cette section, nous présentons l'architecture générale de notre système d'arpentage automatisé. Le robot doit gérer trois activités en parallèle : la **navigation locale**, la **navigation globale** et l'**arpentage**. La figure 3.11 montre l'interaction entre ces trois activités.

Le système se décompose selon trois niveaux de hiérarchie :

1. **Activités principales** : navigation globale, navigation locale, arpentage ;
2. **Modules** : chaque *activité principale* est formée de un ou plusieurs *modules* ;
3. **Fonctions** : pour implanter le système, nous divisons aussi chaque module en *fonctions* ; une fonction effectue une opération précise sur les données.

Nous traitons dans ce qui suit des rôles de chaque activité principale et de leur décomposition en modules. Nous ne décrivons ici que les niveaux hiérarchiques 1 et 2 du système. Le troisième niveau, celui des fonctions, sera traité dans les deux prochains chapitres.

3.4.1 Activité de navigation locale

La navigation locale s'occupe de contrôler le déplacement du robot dans la mine afin d'effectuer le chemin global en l'absence de trajectoire pré-établie ou d'information métrique préalable. Cette activité se décompose en deux modules.

3.4.1.1 Module de contrôle de suivi de galerie

Le **contrôleur de suivi de galerie** est actif pendant que le robot se trouve dans une galerie. Ce module prend en entrée les données télémétriques des plans latéraux et frontal et doit maintenir le robot le plus proche possible du milieu des deux parois de la galerie avec sa direction tangente à l'axe.

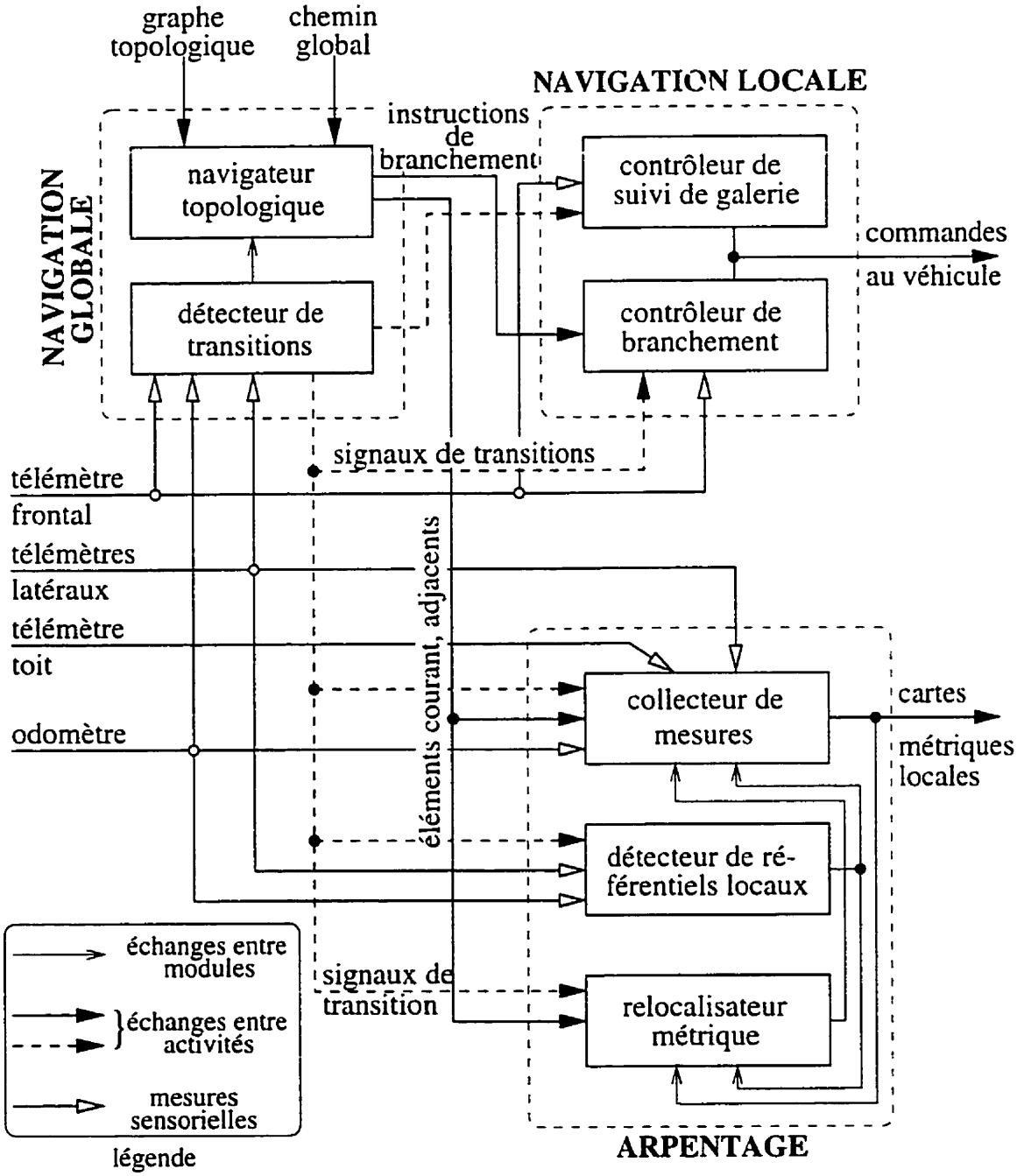


Figure 3.11 : Activités principales, modules et échanges de données et signaux

3.4.1.2 Module de contrôle de branchement

Le **contrôleur de branchement** est actif lorsque le robot traverse une intersection ou une chambre. Dans cette situation, le module de contrôle doit assurer que le véhicule effectue correctement (et sans collision) une trajectoire reliant les extrémités de deux galeries, selon ce qu'indique le chemin global.

Cette activité principale reçoit de la navigation globale des signaux de changements d'élément topologique et, pour effectuer un branchement, l'identité de la *branche* à suivre parmi celles disponibles dans l'intersection ou chambre.

3.4.2 Activité de navigation globale

La navigation globale est responsable de localiser le robot par rapport au graphe topologique de la mine. Cette activité se décompose en deux modules.

3.4.2.1 Module de détection des transitions

Le **détecteur de transitions** doit envoyer un signal, appelé *signal de transition*, aux autres modules chaque fois que le robot change d'*élément topologique* dans la mine. Ce dernier terme a déjà été employé dans ce chapitre, mais nous en donnons maintenant une signification plus précise :

> *définition* : un élément topologique est une région de la mine qui est uniforme du point de vue de la tâche de navigation pour un véhicule. Les catégories d'éléments topologiques qui constituent la mine sont : galerie, intersection et chambre. Dans une galerie, l'espace navigable définit une tâche à une dimension, tandis que dans une intersection ou chambre, l'espace navigable définit une tâche à plusieurs dimensions. La division de toute la mine en de telles régions correspond à construire un graphe de sa topologie. Les galeries de la mine correspondent aux arcs du graphe et les intersections et chambres correspondent aux nœuds du graphe.

Le détecteur de transitions analyse en continu la structure environnante captée par les plans laser latéraux et frontal, afin de reconnaître lorsque le robot débouche dans une intersection ou chambre ou qu'il s'engage dans une nouvelle galerie.

3.4.2.2 Module de navigation topologique

Le **navigateur topologique** maintient un pointeur vers la position courante du robot dans le graphe topologique *a priori*. Alors que le module précédent joue un rôle perceptuel, celui-ci doit plutôt fonctionner sur les structures de données du graphe et du chemin global. Avec chaque transition signalée, il met à jour les étiquettes (provenant du graphe) de l'élément topologique actuel et des éléments adjacents dans le graphe. En combinant cette information avec celle du prochain élément dans le chemin global, ce module indique aussi la bonne branche à suivre au contrôleur en branchement.

Dans un système complètement autonome[1], il faut aussi que ce module s'occupe de la replanification de chemin dans le graphe lorsque survient un événement perceptuel inattendu. Le robot peut se retrouver dans un telle situation s'il a fait auparavant une fausse hypothèse sur l'environnement (*e.g.* mauvais branchement ou fausse identification d'une intersection) ou alors si le graphe *a priori* contient des erreurs par rapport à la mine réelle et aux capacités sensorielles du robot.

3.4.3 Activité d'arpentage

Cette activité principale doit récolter des mesures de la forme géométrique des parois de la mine, dans un format exploitable pour la reconstruction d'un modèle géométrique de la mine. L'activité d'arpentage comporte trois modules.

[1]par opposition au cas d'un véhicule d'arpentage conduit par un humain, qui remplacerait donc les modules autonomes de contrôle local du véhicule et qui superviserait la détection des transitions et les branchements afin de corriger des erreurs éventuelles de navigation globale immédiatement lorsqu'elles se produisent.

3.4.3.1 Module de collecte de mesures

Le **collecteur de mesures** produit et maintient à jour un ensemble de *cartes métriques locales* des différents éléments topologiques de la mine.

> *définition* : une carte métrique locale est une structure de données qui décrit la forme des parois délimitant l'espace d'un élément topologique. La carte contient les informations suivantes :
>
> - étiquette topologique de l'élément et des galeries (respectivement intersections[2]) rattachées à cette intersection (respectivement galerie) ;
> - Transformations géométriques (translation, rotation) entre le *référentiel principal* de la carte (qui est associé à la *jonction* par laquelle le robot est rentré dans l'élément la première fois) et les référentiels associés aux autres jonctions de l'élément ;
> - Listes ordonnées de points spécifiés dans le référentiel principal, chaque liste décrivant une des parois de l'élément. Chaque point de paroi est un triplet (x, z, h) : x et z sont les coordonnées cartésiennes du point dans le plan du sol[3] et h est une valeur de hauteur associée à cette position sur la paroi. Ces mesures proviennent des profils télémétriques latéraux et du toit.

Il faut fournir une définition supplémentaire :

> *définition* : une jonction est formée de deux morceaux de parois qui décrivent l'endroit où une galerie et une intersection, adjacentes dans

[2]Étant donné que les intersections et les chambres sont de même nature dans la topologie de la mine, nous ne parlerons dorénavant que d'intersections. Toute discussion sur les intersections s'appliquera donc aussi aux chambres.

[3]Nous orientons le repère cartésien de l'espace tridimensionnel tel que les axes X et Z définissent le plan du sol et l'axe Y positif pointe vers le haut. Cette convention s'applique aussi bien au repère rattaché au véhicule qu'aux référentiels placés dans la mine.

la topologie de la mine, se rattachent physiquement dans la mine. La jonction chevauche les deux éléments topologiques.

Le collecteur de mesures exploite les résultats de l'analyse faite par les deux autres modules de l'activité d'arpentage.

3.4.3.2 Module de détection des référentiels locaux

Le **détecteur de référentiels locaux** s'occupe de localiser précisément les référentiels locaux à l'intérieur des jonctions. Il communique cette information aux modules de collecte de mesures et de relocalisation métrique.

définition : un référentiel local est un repère cartésien 3D, dans le plan du sol, associé à une jonction. Il sert d'origine pour l'arpentage d'un élément topologique et indique l'endroit précis du raccordement entre cet élément et le précédent dans le parcours suivi par le robot.

La figure 3.12 illustre les notions de "jonction" et de "référentiel local". Notons que le terme "jonction" sera redéfini d'un point de vue perceptuel et le terme "référentiel local" d'un point de vue algorithmique, au chapitre suivant.

3.4.3.3 Module de relocalisation métrique

Le **relocalisateur métrique** intervient lorsque le robot repasse par un élément topologique déjà arpenté (*cf.* la remarque à la fin de la section 3.1.1). Afin de fusionner correctement les nouvelles données d'arpentage avec la carte locale existante, il faut parvenir à localiser le robot par rapport à cette carte.

La figure 3.13 illustre le problème. À son premier passage par une intersection de degré trois, le robot suit une trajectoire de la galerie G_1 vers la galerie G_2. Il crée une première carte métrique *partielle* dont l'origine est le référentiel local $\{R_1\}$. Il calcule en même temps les relations géométriques ${}^2_1T$ et ${}^3_1T$ entre $\{R_1\}$ et les référentiels locaux $\{R_2\}$ et $\{R_3\}$ associés aux deux autres jonctions.

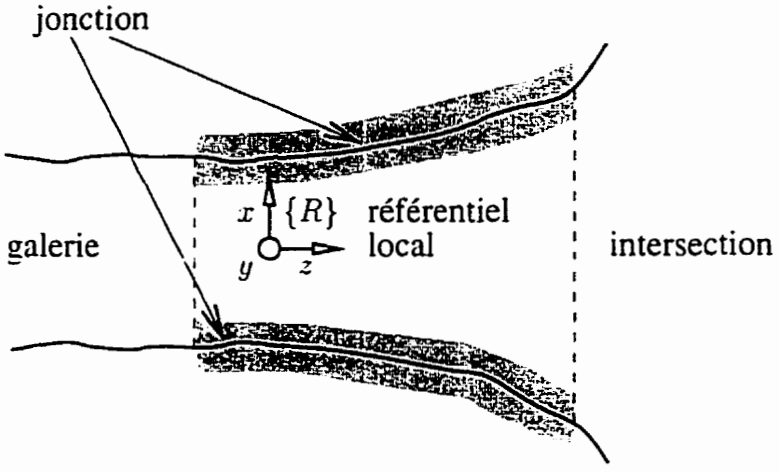


Figure 3.12 : Jonction et référentiel local

Au deuxième passage, le robot rentre dans l'intersection depuis G_3, pour suivre une trajectoire vers G_2. L'origine de la deuxième carte métrique partielle sera $\{R'_3\}$. Le relocalisateur métrique se servira de la pose relative ${}^{3}_{1}T$ (provenant de la carte existante) comme valeur initiale pour déterminer la relation géométrique ${}^{3'}_{1}T$ entre les origines des deux cartes partielles.

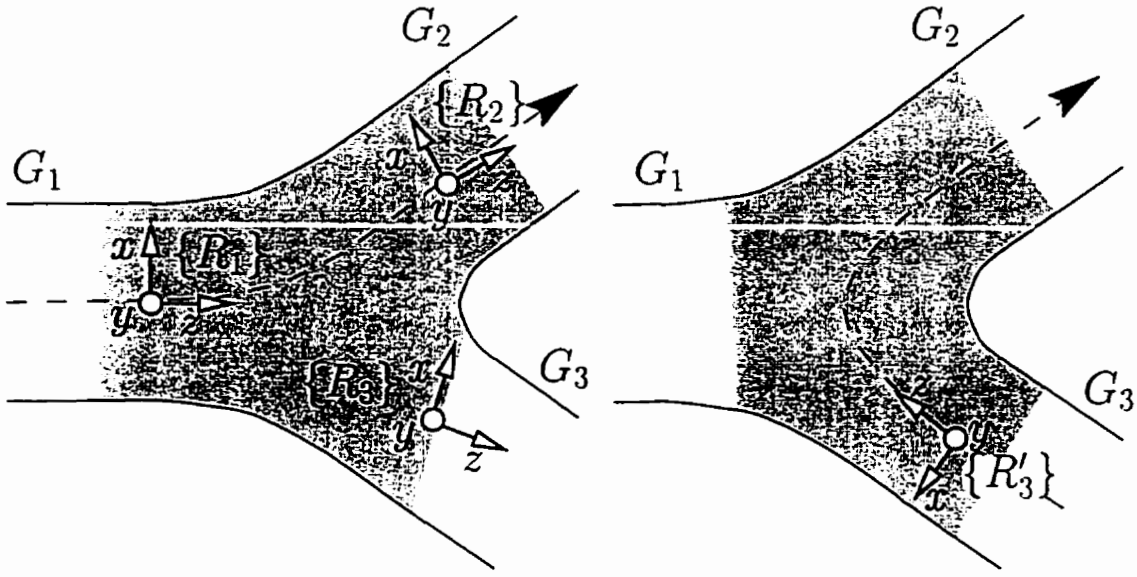
3.4.4 Erreurs de positionnement métrique

Que l'on emploie des encodeurs de roues ou des senseurs inertiels pour mesurer le déplacement du véhicule, l'intégration de la trajectoire sera dans les deux cas entachée d'erreurs, comme il a été montré à la section 3.3.2. Ce phénomène est représenté dans la figure 3.14 et par les équations qui suivent.

$$X_2 = X_1 + \Delta z \, sin(\theta_1) + \Delta x \, cos(\theta_1)$$

$$Z_2 = Z_1 + \Delta z \, cos(\theta_1) - \Delta x \, sin(\theta_1)$$

$$\theta_2 = \theta_1 + \Delta\theta$$



(a) 1^e passage. (b) 2^e passage.

Figure 3.13 : Relocalisation métrique

$$(\Delta z, \Delta x, \Delta\theta)_{odom} = (\Delta z, \Delta x, \Delta\theta)_{réel} + (\delta z, \delta x, \delta\theta)_{err}$$

Alors que l'erreur $(\delta x, \delta z, \delta\theta)_{err}$ de mesure du *déplacement* instantanée $(\Delta x, \Delta z, \Delta\theta)_{réel}$ sera généralement d'amplitude constante, l'erreur sur la *pose* instantanée du véhicule dans un référentiel externe va accumuler les erreurs $(\delta x, \delta z, \delta\theta)_{err}$ au cours du temps et pourra donc croître sans bornes si l'on ne lui applique pas régulièrement une correction.

3.4.4.1 Conséquences pour le système

La conception de notre système tient compte de l'incertitude sur la pose mesurée du véhicule de deux manières. D'une part, les données d'arpentage (cartes métriques locales) des différentes régions de la mine sont spécifiées dans des référentiels propres aux éléments topologiques, plutôt que dans un seul repère global de toute la mine. La navigation globale, qui intrinsèquement n'exploite pas la localisation métrique

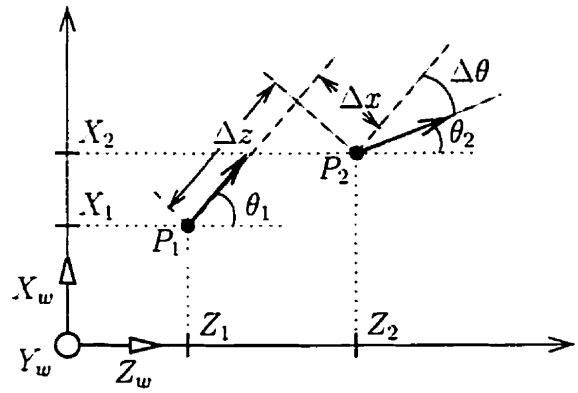


Figure 3.14 : Erreurs de mesure de trajectoire par odométrie

du robot, permet en quelque sorte de faire régulièrement des "remises à zéro" de l'intégration de trajectoire par le senseur interne.

D'autre part, la stratégie de relocalisation métrique n'utilise pas l'odométrie, mais plutôt la détection de structures externes caractéristiques des zones de transition entre éléments topologiques. Ce sont les référentiels locaux associés aux jonctions qui permettront au robot de se situer à l'intérieur d'une région qu'il a déjà explorée.

3.4.4.2 Déformations non-rigides des cartes métriques

La stratégie de relocalisation métrique élucidée ci-dessus a supposé que le fusionnement de deux cartes métriques partielles ne faisait intervenir qu'une transformation *rigide* entre les deux cartes locales. Cependant, les erreurs d'intégration de trajectoire par le senseur interne induisent en réalité *une déformation non-rigide* de la forme réelle de l'élément topologique dans la carte mesurée. La figure 3.15 illustre ceci dans le cas d'une galerie curviligne que le robot traverse deux fois en sens inverses. La déformation de la galerie lors de chaque passage a deux composantes : l'erreur latérale d'amplitude croissante le long de la trajectoire et l'erreur longitudinale liée au fait que les encodeurs de roues ont tendance à surestimer (dans cet exemple) la distance parcourue, à cause du glissement des roues.

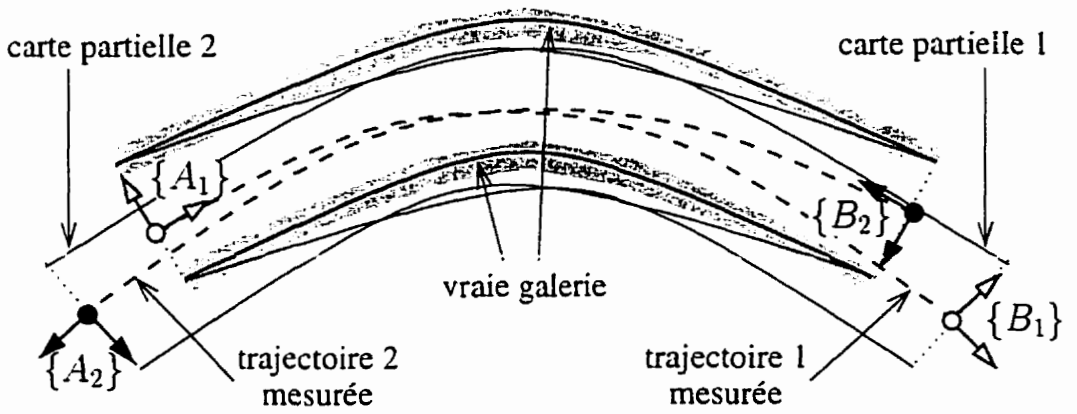


Figure 3.15 : Déformation non-rigide de la carte d'une galerie

Si l'on sait que les référentiels locaux $\{A\}$ et $\{B\}$ associés aux deux extrémités de la galerie sont détectés (approximativement) aux mêmes endroits dans la mine entre les deux passages, il devient possible de corriger les déformations dans chaque carte partielle. L'équation suivante représente le point de départ d'une telle approche :

$$\hat{{}^B_A T} = f({}^B_A T_1, \epsilon_1, {}^B_A T_2, \epsilon_2) \qquad \epsilon_1, \epsilon_2 : \text{ incertitudes}$$

La première étape serait donc de calculer la transformation géométrique corrigée $\hat{{}^B_A T}$ entre $\{A\}$ et $\{B\}$, en pondérant les deux mesures ${}^B_A T_1$ et ${}^B_A T_2$ de cette entité par les incertitudes ϵ_1 et ϵ_2 (qu'il faut modéliser) sur les deux trajectoires intégrées. Ensuite, il faudrait propager la correction de ${}^B_A T$ sur les points de mesure de chaque carte partielle de la galerie. Nous pourrions nous inspirer ici du principe de rétro-propagation de correction d'odométrie adopté par Brooks (1985) et Chatila et Laumond (1985), ou bien de l'approche de Lu et Milios (1995), qui traitent ce problème comme l'optimisation d'un réseau de contraintes odométriques et structurales entre vues partielles de l'environnement.

Néanmoins, la tâche de simultanément corriger les déformations et fusionner plusieurs cartes métriques demeure assez complexe. En effet, à un niveau plus global

dans la mine, il existe aussi le problème de non-fermeture (géométrique) des boucles d'éléments topologiques. Lorsque que le robot repasse par un endroit déjà arpenté, il y aura forcément une erreur $(\Delta x, \Delta z, \Delta\theta)$ de non-fermeture, liée à l'odométrie. Ce problème est d'autant plus difficile que le graphe topologique de la mine peut éventuellement contenir des boucles imbriquées, chacune donc imposant des contraintes de correction d'odométrie qui affectent les autres.

Cependant, nous remarquons que les arpenteurs humains font face à ce même problème à cause des erreurs de mesure le long de parcours fermés dans la mine, comme nous avons vu à la section 2.3. Mais en répartissant l'erreur totale de non-fermeture sur l'ensemble des positions dans le parcours, les données d'arpentage corrigées permettent de cartographier adéquatement la mine. D'autre part, nous venons de voir que plusieurs approches existent dans la littérature, *e.g.* Brooks (1985), Chatila et Laumond (1985) et Lu et Milios (1995), pour fusionner des données métriques partielles d'un milieu en présence d'erreurs d'intégration de trajectoire. Par conséquent, nous ne poursuivons pas la résolution de ce problème dans ce projet.

3.4.5 Suite du mémoire

Le problème du contrôle local de véhicule basé sur l'information perceptuelle de l'environnement (en l'absence de trajectoire pré-établie) a été étudié en recherche, y compris dans le contexte du milieu minier souterrain (*cf.* Juneau et al., 1993; Steele et al., 1993; Banta et al., 1992). Pour cette raison, nous avons jugé préférable de laisser de côté cet aspect du système au profit des aspects perceptuels reliés à la localisation globale dans la mine et à la construction d'une représentation géométrique cohérente de la mine, problèmes qui ont été comparativement peu étudiés jusqu'à présent.

Dans le reste de ce projet, nous nous consacrons donc au développement des activités principales d'arpentage et de navigation globale. Le chapitre 4 traitera de l'arpentage : sa décomposition au niveau hiérarchique des fonctions, les notions

importantes dans la conception de celles-ci, puis l'implantation algorithmique et expérimentale du module de détection des référentiels locaux. Au chapitre 5, nous présenterons de la même manière le développement de l'activité de navigation globale.

Chapitre 4

Activité d'arpentage

4.1 Introduction

Dans ce chapitre, nous traitons en un premier temps de la conception des trois modules de l'activité principale d'arpentage et de leur décomposition au niveau fonctionnel. En un deuxième temps, nous présentons le développement du module de détection des référentiels locaux. En un troisième temps, nous présentons des résultats expérimentaux pour la détection des référentiels locaux.

4.2 Conception du collecteur de mesures

4.2.1 Décomposition du module en fonctions

Le schéma d'interaction des différentes fonctions du module collecteur de mesures se trouve à la figure 4.1. Les paragraphes qui suivent décrivent les fonctions (les boîtes dans l'organigramme) en termes d'entrées, sorties et rôle.

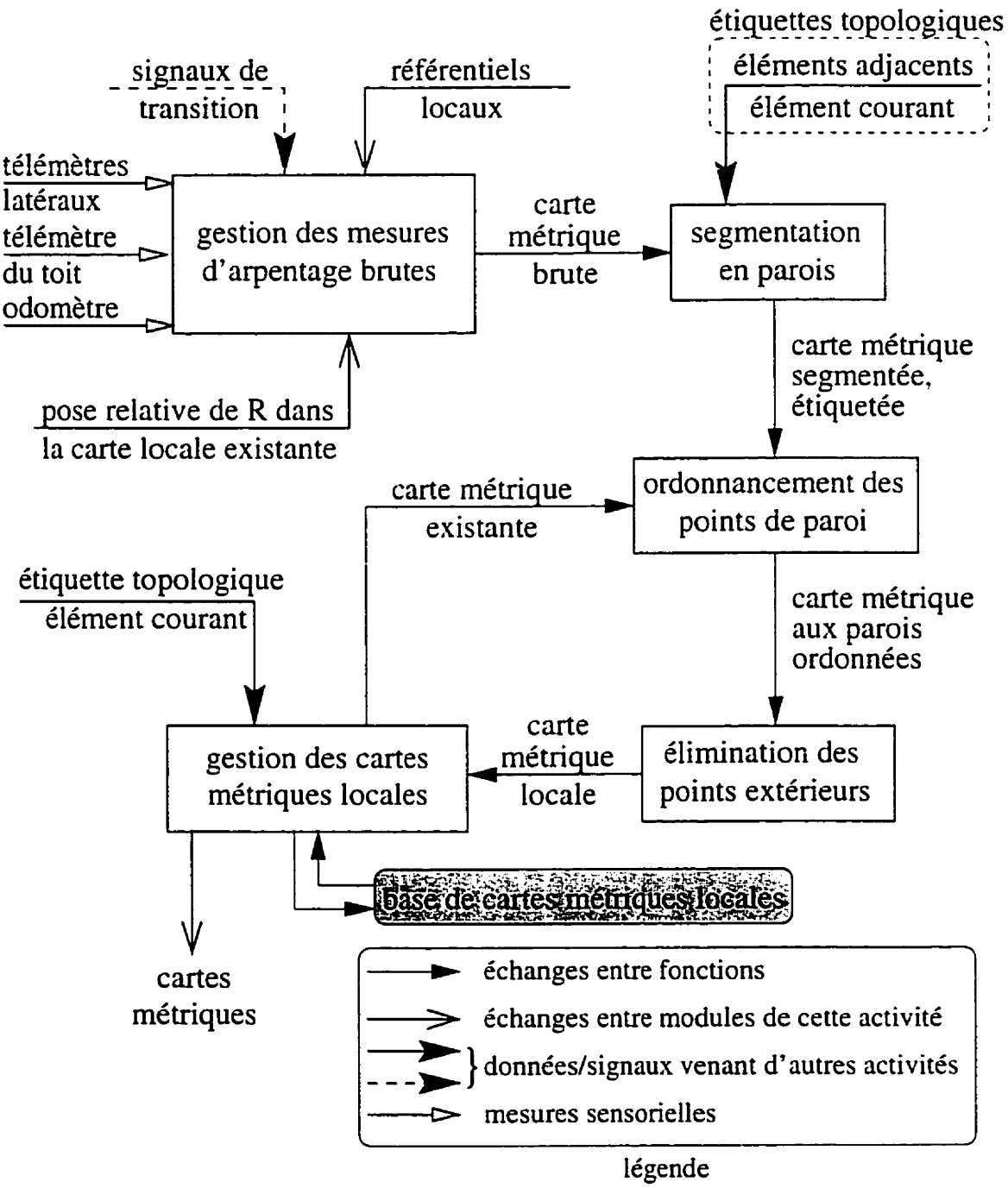


Figure 4.1 : Collecteur de mesures : échanges de données et signaux

4.2.1.1 Gestion des mesures d'arpentage brutes

Cette fonction crée une *carte métrique brute* de l'élément topologique actuellement traversé, en accumulant les profils télémétriques latéraux et du toit, en convertissant ces profils en mesures de distance (dans le repère du robot) et en associant ces mesures à leurs valeurs odométriques dans le référentiel local détecté à l'entrée de cet élément (*cf.* tableau 4.1). Dans la carte métrique brute, les parois ne sont pas décrites séparément; il s'agit d'un seul "nuage" de points décrivant la structure de l'élément.

Cette fonction doit s'assurer de la gestion correcte des données d'arpentage lors du passage entre deux éléments topologiques successifs. En fait, deux éléments topologiques consécutifs dans le chemin du robot vont partager, dans leurs cartes brutes, des tronçons de parois en commun (ceux formant la jonction). Lorsque le signal de transition est reçu, cette fonction s'occupe de transformer les mesures déjà accumulées pour l'élément actuel vers le nouveau référentiel local.

Tableau 4.1 : Fonction de gestion des mesures d'arpentage brutes

entrée	- profils télémétriques instantanés latéraux et du toit ; - déplacements odométriques entre instants de mesure ; - référentiel *d'entrée* de l'élément topologique (relatif au robot) ; - référentiel(s) de *sortie* de l'élément topologique (relatifs au robot) ; - signaux de transition.
sortie	- ensemble de points (triplets (x, z, h)) décrivant l'élément topologique. - poses relatives entre le référentiel principal et les autres référentiels. $\Longrightarrow$ *Carte métrique brute.*

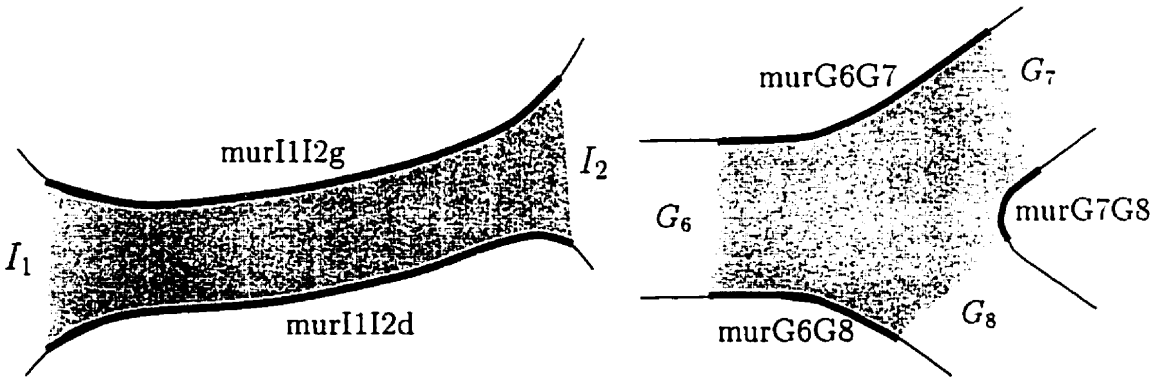
4.2.1.2 Segmentation en parois

Une fois l'élément topologique traversé, cette fonction segmente la carte métrique brute en parois physiquement distinctes et identifie celles-ci par rapport au graphe topologique (étiquetage topologique) (*cf.* tableau 4.2).

Tableau 4.2 : Fonction de segmentation en parois

entrée	- ensemble de points décrivant l'élément topologique ; - étiquettes topologiques des éléments courant et adjacents.
sortie	- une série de points (x, z, h) *par paroi* de l'élément topologique ; - poses relatives entre le référentiel principal et les autres référentiels. - étiquette topologique associée à chaque paroi. $\Longrightarrow$ *Carte métrique segmentée, étiquetée.*

La figure 4.2 sert d'illustration pour l'étiquetage topologique de parois. Dans la figure 4.2(a), l'étiquetage distingue la paroi gauche ("murI1I2g") de la paroi droite ("murI1I2d") lorsque le robot traverse une galerie en allant de l'intersection I_1 vers I_2. Dans la figure 4.2(b), une intersection se rattache à trois galeries, G_6, G_7 et G_8. Les trois parois se nomment alors "murG6G7", "murG7G8" et "murG6G8".



(a) d'une galerie. (b) d'une intersection.

Figure 4.2 : Étiquetage topologique des parois

4.2.1.3 Ordonnancement des points de parois

Cette fonction prend la carte produite par la fonction précédente et doit *ordonner* les points de chaque paroi dans un sens de parcours le long de la paroi (*cf.* tableau 4.3).

En effet, à cause de la rugosité des parois, mais surtout lorsque le robot effectue un virage dans une intersection, des *inversions* pourront se produire dans l'ordre de *mesure* de points par les senseurs latéraux, par rapport à l'ordre *idéal* de ces mêmes points dans un sens de parcours le long de la paroi. Conceptuellement, ce *sens de parcours* correspond à la variation monotone du paramètre d'une courbe qui modéliserait la paroi.

Tableau 4.3 : Fonction d'ordonnancement des points de parois

entrée	- ensemble de parois décrites comme des séries de points non-ordonnées; - ensemble de parois provenant de la carte locale existante (si l'élément topologique est déjà connu).
sortie	- ensemble de parois décrites par des séries ordonnées de points. - poses relatives entre le référentiel principal et les autres référentiels. $\Longrightarrow$ *Carte métrique aux parois segmentées, étiquetées et ordonnées.*

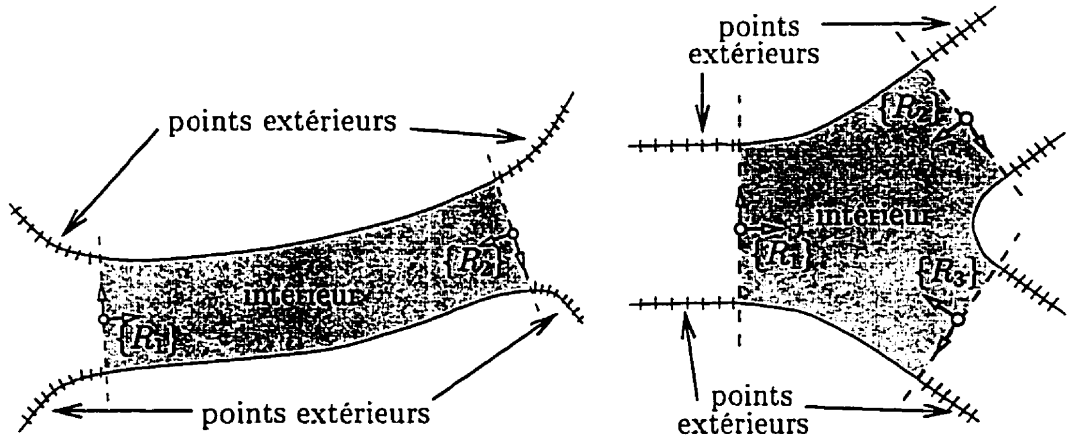
À cette étape intervient aussi le *fusionnement* de la carte métrique actuellement mesurée avec la carte existante de l'élément topologique, si celui-ci a déjà été parcouru. Ici, la fusion des deux cartes partielles consiste à transformer les points de la nouvelle carte partielle vers le *référentiel principal*[1], défini dans la carte existante, puis de réunir (grâce à l'étiquetage) les séries de points appartenant aux mêmes parois. Ensuite, le processus d'ordonnancement leur est appliqué.

4.2.1.4 Élimination des points extérieurs

Afin de modéliser correctement les connections entre éléments topologiques de la mine, il est important de supprimer de la carte locale maintenant formée les points de parois qui tombent à *l'extérieur* de la région qui est à représenter. Plus précisément, nous définissons la région *intérieure* de la carte comme étant délimitée d'une part par

[1]Le *référentiel principal* d'une carte locale est le référentiel local associé à la jonction par où le robot est rentré dans l'élément topologique la première fois.

les parois mesurées et d'autre part par les droites passant par les référentiels locaux associés à toutes les jonctions de l'élément topologique (voir figure 4.3). Il s'agit alors d'éliminer les points au-delà des intersections entre ces droites et les parois (*cf.* tableau 4.4).



(a) Carte d'une galerie.

(b) Carte d'une intersection.

Figure 4.3 : Notion de points intérieurs et extérieurs à la carte

Tableau 4.4 : Fonction d'élimination des points extérieurs

entrée	- ensemble de parois décrites par des séries ordonnées de points ; - poses relatives entre le référentiel principal et les autres référentiels.
sortie	- les mêmes parois, limitées aux points *intérieurs* à la carte. - poses relatives entre le référentiel principal et les autres référentiels. $\Longrightarrow$ *Carte métrique locale.*

4.2.1.5 Gestion des cartes métriques locales

Cette partie du module de collecte de mesures est responsable d'intégrer le résultat de l'arpentage d'un élément topologique à une base de cartes métriques locales décrivant

les régions explorées de la mine. Elle effectue les opérations d'accès (lecture/écriture) à cette base globale de données géométriques et fournit des cartes locales d'éléments individuels à d'autres fonctions (*i.e.* l'ordonnancement de points de parois) et modules (*i.e.* le relocalisateur métrique) (*cf.* tableau 4.5).

Tableau 4.5 : Fonction de gestion des cartes métriques locales

entrée	Carte métrique locale traitée (résultat des fonctions précédentes).
sortie	- base de donnée de cartes métriques locales décrivant tous les éléments topologiques traversés ; - cartes métriques locales individuelles (à d'autres fonctions/modules).

4.2.2 Limite du développement

La gestion des mesures d'arpentage brutes et la gestion des cartes métriques locales ne font intervenir que des opérations simples sur les données : données sensorielles spatio-temporelles dans le premier cas et données géométriques (avec étiquettes topologiques) dans le deuxième. Il n'y a en définitive que des problèmes "classiques" à résoudre dans ces deux fonctions.

L'ordonnancement des points de parois peut paraître à prime abord comme une tâche facile. Mais il s'agit de trier des éléments à deux dimensions (les coordonnées (x, z) des points), chose plus compliquée que le simple tri de nombres. De plus, la complexité éventuelle de la forme des parois, et la formation de "boucles" dans les points de mesures (liées à la trajectoire du véhicule), rendent la tâche plus difficile. Cependant, comme nous verrons à la section 4.5, ce problème est développé algorithmiquement dans le contexte de la détection de référentiels locaux.

L'élimination des points extérieurs fait intervenir une notion intéressante qui est celle de région intérieure à la carte d'un élément topologique. Cependant, une fois les parois de la carte modélisées adéquatement par des courbes (processus que nous

développons à la section 4.5) et leurs intersections calculées avec les droites limitrophes, la détermination des points extérieurs devient triviale.

La seule fonction de ce module qui pose un problème vraiment intéressant est la segmentation de la carte métrique brute en parois, dans le cas d'une intersection[2]. En effet, les facteurs de la trajectoire suivie, de la forme (éventuellement complexe) des parois, et des occlusions et zones non-visibles de l'espace navigable, peuvent se combiner durant la traversée d'une intersection pour rendre la segmentation des points de mesure non-triviale. Cependant, il s'agit encore une fois d'un problème que nous développons (du moins en partie) dans le contexte de la détection de référentiels locaux.

Compte tenu de la discussion précédente et aussi par souci de limiter l'envergure du projet, nous nous arrêtons à ce stade-ci dans la conception du module de collecte de mesures.

4.3 Conception du relocalisateur métrique

Rappelons que l'objectif de ce module-ci est de calculer la position du robot, lorsque celui-ci rentre de nouveau dans un élément topologique connu, par rapport à l'un des référentiels de la carte métrique locale. Ceci est un problème de localisation métrique d'un robot mobile à l'aide de points de repères détectés dans le milieu. C'est une tâche fondamentale dans l'exploration de milieux et la navigation autonome exploitant une carte géométrique (*cf.* sections 2.1.2 et 2.2 de la revue bibliographique). Avant de présenter la conception de ce module, nous fournissons donc une revue bibliographique des méthodes de localisation métrique de robots mobiles.

[2]Dans une galerie, l'appareillage télémétrique segmente naturellement les mesures en deux parois, à gauche et à droite du robot. Étant donné qu'une galerie n'accuse jamais de changements brusques d'orientation (axe généralement continu en courbure), cette "segmentation sensorielle" demeurera correcte.

4.3.1 Méthodes de localisation métrique dans la littérature

Il se dégage des travaux faits dans ce domaine deux approches principales qui se distinguent par le degré d'analyse appliquée aux mesures sensorielles (qui dans tous les cas proviennent d'un télémètre balayant la scène dans un plan parallèle au sol).

4.3.1.1 Méthodes dites "iconiques"

La première approche, dite "iconique", exploite *directement* les *points* formant les profils télémétriques captés, pour localiser le véhicule. Quelques systèmes employant une telle stratégie sont ceux de MacKenzie et Dudek (1994), Cox (1991) et Gonzalez et al. (1992). Le robot possède une carte qui décrit les structures saillantes dans l'environnement sous forme de segments de droites. Le positionnement part d'une valeur de pose du véhicule fournie par l'odométrie. L'algorithme de base de l'approche iconique est alors le suivant :

1. **Classification** des points de profil instantané pour les associer aux segments de la carte selon le critère de distance euclidienne (erreur) e_i minimale entre point et segment (ou droite infinie) ;

2. Calcul d'une **correction** $T_{corr} = (\Delta x, \Delta z, \Delta\theta)_{corr} = (\vec{V}_{corr}, \Delta\theta_{corr})$ de la pose initiale $(x, z, \theta)_{odom}$ qui minimise globalement une somme S pondérée sur les distances e_i entre les segments et leurs points associés ;

3. **Application** de T_{corr} et recalcul de S ; si S et/ou $||\vec{V}_{corr}||$ est toujours trop grand, remonter à l'étape 1.

MacKenzie et Dudek (1994) séparent en deux étapes la correction de la position odométrique, $\vec{V}_{corr}$ et celle de l'orientation, $\Delta\theta_{corr}$. Le vecteur $\vec{V}_{corr}$ est calculé simplement comme la somme des vecteurs de corrections $\vec{V}_i$ ramenant chaque point sur son segment correspondant. On calcule une mesure de qualité E_{cf} de l'estimation de la position. Lorsque E_{cf} est suffisant, on calcule alors $\Delta\theta_{corr}$ en maximisant la fonction $E_{cf}(\theta)$.

Cox (1991) emploie une méthode de régression linéaire pour calculer T_{corr}. On linéarise l'expression de la transformation géométrique d'un point dans le plan cartésien, en supposant que θ_{corr} sera petit ($\ll$ 1 radian).

Enfin, Gonzalez et al. (1992) se servent d'une grille d'occupation superposée à la carte géométrique : cette deuxième carte divise le plan cartésien en cellules et identifie si chaque cellule contient des segments dans la carte géométrique. Ceci permet d'accélérer l'étape de classification des points de profil.

4.3.1.2 Méthodes basées sur l'extraction de formes polygonales

Dans la seconde approche, on applique une analyse préalable aux points de profils télémétriques, pour en *extraire des formes polygonales.* La carte du milieu est elle aussi constituée des mêmes catégories de formes, *e.g.* segments, coins et autres groupements de segments caractérisant les contours d'objets typiques d'un milieu structuré. Ceci requiert un raisonnement et des outils de calcul en général plus sophistiqués que dans la méthode iconique. En revanche, le niveau supplémentaire d'analyse sensorielle apporte une réduction de l'espace de recherche dans la classification des observations et permet au processus de localisation d'exploiter davantage les connaissances sur l'environnement.

Des systèmes employant cette deuxième méthode sont décrits dans Shaffer et al. (1992), Crowley (1985), Borthwick et Durrant-Whyte (1994) et Bulata et Devy (1994). Toutes ces implantations suivent l'algorithme de base suivant :

1. **Prédiction** des structures à observer dans l'environnement, à partir de la carte et du déplacement du robot prédit par le contrôle et/ou mesuré par l'odométrie ;

2. **Extraction** de formes polygonales du profil capté à la nouvelle position du robot;

3. **Appariement** des formes détectées (observations) avec celles de la carte (modèles), selon des critères de proximité et de cohérence géométrique ;

4. **Correction** de la pose du robot prédite par le contrôle et/ou mesurée par l'odométrie.

Shaffer et al. (1992) modélisent l'environnement par des segments et des coins. On applique un "opérateur de coins" au profil pour isoler les changements brusques d'orientation des surfaces ; les segments joignant les coins successifs sont interpolés. Pour la mise en correspondance avec la carte, on considère d'abord la similitude entre les formes prédites et détectées, aux niveaux de **a)** la position des formes relativement au robot (± tolérances) et **b)** les propriétés des formes (longueur, orientation, angle d'ouverture, ± tolérances). On établit ainsi une liste de *paires modèle-observation*, ordonnée dans le champs de vue du robot. On considère ensuite la cohérence entre les relations spatiales reliant les formes observées et les relations reliant leurs contreparties dans la carte, pour déterminer la liste maximale de paires consécutives cohérentes. C'est cette sous-liste qui sert à corriger la pose du robot (fournie par l'odométrie) selon l'algorithme itératif de Gauss-Newton.

Crowley (1985) introduit le Modèle Local Composite (MLC) : c'est une carte de l'environnement immédiat du robot et qui se compose de segments de droites et de leurs niveaux de confiance associés. Un processus itératif de segmentation modélise les profils (sonar) captés par des segments. L'auteur propose une série de tests de proximité, de colinéarité et de chevauchement des angles soutendus dans le champs de vue du robot, pour établir les paires modèle-observation. La correction de pose revient à une simple moyenne des vecteurs d'erreur de toutes les paires.

Borthwick et Durrant-Whyte (1994) se basent sur les segments et les coins (convexes ou concaves) dans leur modélisation. On découpe le profil en segments par un filtrage médian qui isole les déviations importantes de profondeur. Certaines configurations de segments adjacents indiquent la présence de coins. Les formes, segments ou coins, identifiées dans le profil sont les "cibles" observées. Pour chacune d'elles, on recherche dans la carte une cible du même type suffisamment proche ; une distance de Mahalanobis calculée sur *l'innovation* entre les positions prédites et observées de

la cible sert à valider l'appariement. La correction de la pose, effectuée à chaque nouvelle cible observée, emploie le filtrage Kalman. Ceci implique qu'on prédise (à l'instant t) grâce à la commande de contrôle, la pose du robot et la position relative de la cible à $t+1$; qu'on calcule grâce à l'observation réelle (à $t+1$), l'innovation sur la prédiction du vecteur d'état du véhicule ; et enfin qu'on corrige (à $t+1$) la pose prédite du robot.

Enfin, Bulata et Devy (1994) ont deux niveaux d'interprétation des mesures télémétriques. Au premier niveau, on segmente le profil en segments de droites. Pour construire la carte du milieu, on regroupe les segments en plusieurs configurations possibles, appelées *landmarks*, qui représentent des murs, des meubles et des ouvertures ou portes. Pour localiser le robot, l'appariement se fait entre des landmarks de la carte et les segments détectés dans le profil. Ceci se produit en deux étapes :

1. Construction d'hypothèses, chaque landmark étant considéré en isolement, en utilisant des contraintes de proximité et de relations spatiales entre segments. On construit une table d'hypothèses d'appariement pour chaque landmark ;

2. Élimination des fausses hypothèses par une méthode de clique maximale, en exploitant des contraintes entre landmarks, *e.g.* l'exclusivité mutuelle des sous-ensembles de segments appariés. On retient la clique maximale d'hypothèses cohérentes.

La correction de la pose utilise comme les deux systèmes précédents l'approche du filtrage Kalman. On maintient des filtres Kalman pour l'état du véhicule et pour chaque landmark visible. Si la pose actuelle du robot est connue avec une bonne précision (quantifiée par une matrice de covariances), la localisation va "suivre" seulement quelques landmarks de positions précises dans le voisinage. Si par contre le robot est "perdu" (sa pose est de mauvaise précision), la localisation cherchera tous les landmarks connus dans cette région de la carte afin d'initialiser de nouvelles cibles et ré-initialiser l'état du véhicule.

4.3.2 Considérations dans notre cas

Notre milieu est partiellement structuré. À une échelle fine, les parois décrites dans une carte métrique locale seront de formes irrégulières, c'est-à-dire non-rectilignes et rugueuses. Par conséquent, les formes polygonales caractérisant les milieux et extraites des mesures télémétriques dans les systèmes du paragraphe 4.3.1.2 ne s'appliquent pas bien à notre situation.

Nous envisageons donc plutôt d'utiliser une méthode *iconique* pour relocaliser le véhicule. Ceci signifie de comparer directement des points de mesure de parois (correspondant au passage actuel par l'élément topologique) avec la carte métrique locale de l'endroit. En fait, les données sensorielles proviennent de l'arpentage de la jonction et non du balayage de la scène à partir d'une seule position du robot.

Afin d'appliquer une approche de localisation telle qu'au paragraphe 4.3.1.1, nous aurions normalement besoin de traiter la carte métrique locale pour obtenir un niveau de description plus élevé que celui des listes de points de parois. Une première option serait de décrire chaque paroi comme une courbe paramétrique dans le plan. Ceci entraînerait deux difficultés : la sélection d'une classe de courbes qui s'appliquerait adéquatement à toutes les formes de parois rencontrées ; et l'adaptation à l'approche iconique (classification des points aux courbes, calcul de la correction) d'une paramétrisation plus complexe que celle d'un segment de droite. Une deuxième option serait de décomposer chaque paroi en segments de droites. La segmentation isolerait des endroits de coupure par un critère de courbure locale ; des segments seraient alors calculés sur les morceaux segmentés. Nous pourrions ensuite appliquer directement une des approches iconiques vues précédemment.

Mais en réalité, il y a une solution plus simple : mettre en correspondance les points de la jonction actuelle *avec les points* de la carte locale. A priori, ceci pose un problème car les points de mesure de la paroi actuellement arpentée, d'une part, et de cette même paroi dans la carte locale, d'autre part, ne correspondent pas aux mêmes endroits exacts dans l'environnement. Ceci introduit une erreur (de mesure) dans

chaque *vecteur de correction* entre point de jonction et point de la carte. Cependant, si les mesures sont suffisamment denses (les points suffisamment rapprochés), ces erreurs deviennent minimes. D'autre part, l'aspect aléatoire de ces erreurs de mesure fait qu'elles auront tendance à s'annuler mutuellement dans le calcul de la correction totale de la pose du robot.

L'appariement direct de points à points pose un deuxième problème, celui-ci plus grave. Si un point donné mesuré dans la jonction actuelle se situe sur une partie de la paroi qui est absente de la carte locale, le vecteur de correction pour ce point sera erroné et de longueur importante. Il pourra évidemment corrompre le calcul de la nouvelle pose. Mais nous pouvons éliminer ce risque tout simplement en supprimant du calcul de correction les vecteurs individuels de norme trop grande.

4.3.3 Décomposition du module en fonctions

Le schéma d'interaction des différentes fonctions du module relocalisateur métrique se trouve à la figure 4.4. Les paragraphes qui suivent décrivent les fonctions (les boîtes dans l'organigramme) en termes d'entrées, sorties et rôle.

4.3.3.1 Estimation initiale de la pose relative

Cette fonction doit calculer la pose relative du robot par rapport au référentiel principal de la carte locale existante de l'élément topologique courant, ${}^{R}_{c}T_{init}$. Elle prend en entrée la pose relative (par rapport au robot) du référentiel local de la jonction actuellement traversée et la transformation géométrique entre le référentiel correspondant et le référentiel principal dans la carte locale (*cf.* tableau 4.6). La pose relative calculée ici sert en fait de valeur initiale pour le processus de relocalisation.

Cette fonction doit être synchronisée avec les transitions entre éléments topologiques, d'où l'activation par les signaux de transition provenant de la navigation globale.

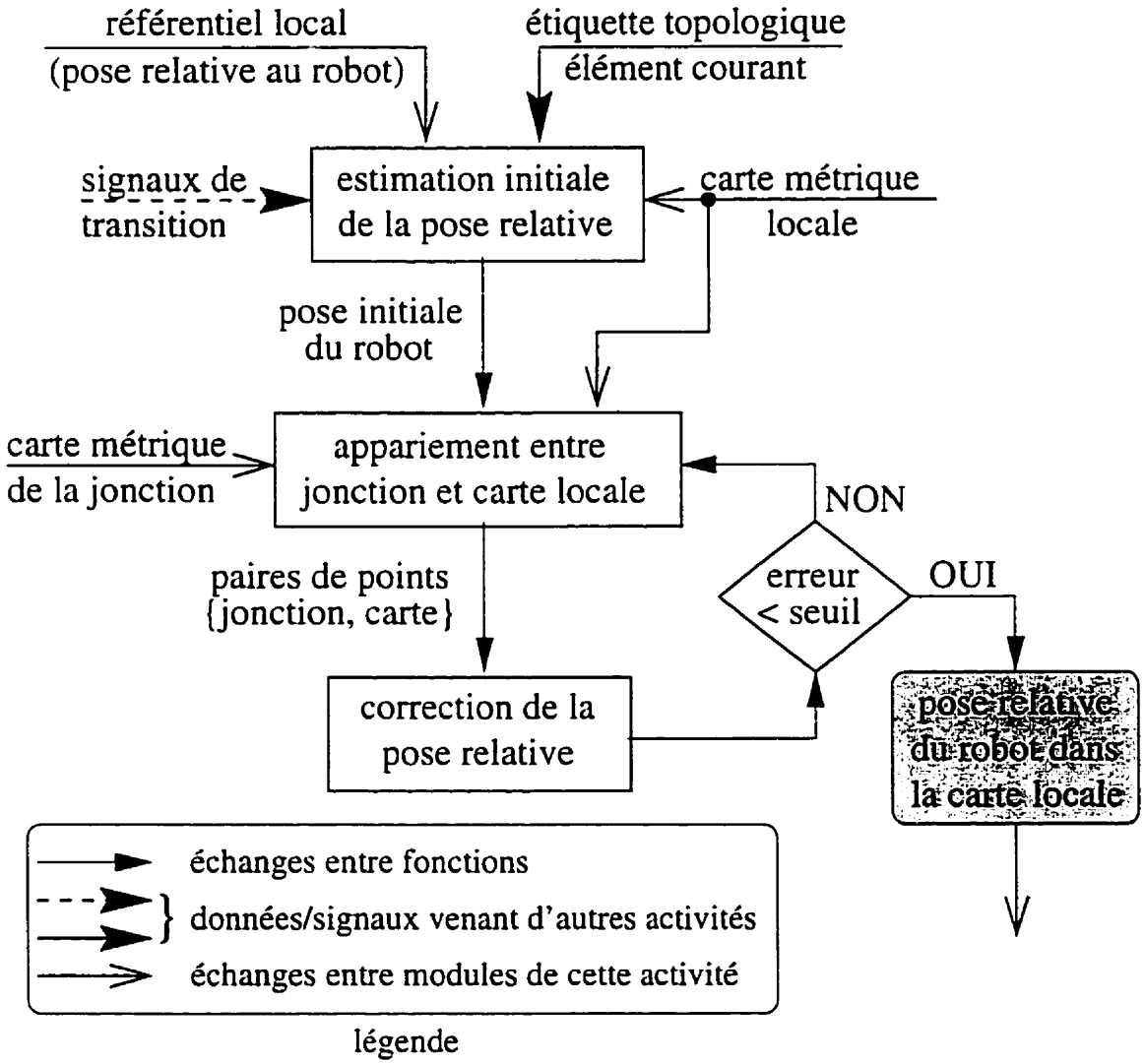


Figure 4.4 : Relocalisateur métrique : échanges de données et signaux

Tableau 4.6 : Fonction d'estimation initiale de la pose relative

entrée	- pose relative entre le robot et le référentiel local de la jonction actuelle; - pose relative entre le référentiel local correspondant (à la même jonction) et le référentiel principal, dans la carte métrique locale; - étiquette topologique de l'élément courant; - signaux de transition.
sortie	${}^{R}_{c}T_{init}$: valeur *initiale* de la pose relative du robot.

4.3.3.2 Appariement entre jonction et carte locale

L'étape suivante est la mise en correspondance de la jonction actuelle (dont la carte métrique provient du détecteur de référentiels locaux) avec la carte métrique locale. Cette fonction utilise un critère de proximité pour apparier les points des parois arpentées dans la jonction avec des points de parois de la carte métrique locale (*cf.* tableau 4.7).

Tableau 4.7 : Fonction d'appariement entre jonction et carte locale

entrée	- pose relative initiale ${}^{R}_{c}T_{init}$; - carte métrique locale de l'élément topologique courant; - parois arpentées de la jonction actuelle;
sortie	Paires de points {jonction,carte} mis en correspondance.

Cet appariement procure l'information nécessaire pour calculer une mesure d'erreur (et donc de corriger) la valeur de ${}^{R}_{c}T_{init}$.

4.3.3.3 Correction de la pose relative

La dernière étape dans la relocalisation est la correction de la pose relative initiale ${}^{R}_{c}T_{init}$. L'objectif est de déterminer un déplacement $(\Delta x, \Delta z, \Delta\theta)_{R,carte}$ qui réduise le plus possible une mesure d'erreur calculée sur les distances entre points appariés par

la fonction précédente (*cf.* tableau 4.8). Autrement dit, il faut ramener globalement les points de jonction vers leurs points associés dans la carte.

Tableau 4.8 : Fonction de correction de la pose relative

entrée	Paires de points {jonction,carte} mis en correspondance.
sortie	Correction de la pose : ${}^{R}_{c}T_{corr} = {}^{R}_{c}T_{init} + (\Delta x, \Delta z, \Delta\theta)_{R,carte}$.

Les deux dernières étapes, la mise en correspondance et la correction de pose, forment une boucle itérative. C'est-à-dire que si la mesure d'erreur (recalculée après application de la correction) demeure trop grande, il faut retourner à la fonction précédente. L'appariement est donc refaite à chaque itération. Ceci signifie qu'à mesure que la pose relative du véhicule est corrigée, les paires associées sont modifiées pour ramener les points de jonction vers des points de la carte locale de plus en plus rapprochés (au sens de leurs endroits physiques sur les parois). Cette condition améliore la convergence du processus vers une solution correcte.

4.3.4 Limite du développement

Comme nous avons vu à la section 4.3.1, le problème de la localisation métrique, à l'aide de connaissances *a priori* sur le milieu et d'informations sensorielles, a été beaucoup étudié dans le domaine de la robotique mobile. Cependant, la conception de notre module de relocalisation métrique se distingue des approches existantes par le type d'informations qu'il exploite : référentiels locaux détectés dans l'environnement et informations topologiques, plutôt que points/segments/formes polygonales détectés dans l'environnement et informations odométriques ou de contrôle. Mais il est clair que l'implantation des trois fonctions de ce module ne mènerait pas à des algorithmes réellement originaux. La première fonction revient en fin de compte à un calcul géométrique assez simple et les deux autres fonctions empruntent l'approche iconique que nous avons vue plus haut.

De même que pour le module collecteur de données, nous choisissons donc de mettre de côté l'implantation du module de relocalisation, pour favoriser le développement du détecteur de référentiels locaux et de la navigation globale.

4.4 Détecteur de référentiels locaux : conception

4.4.1 Conventions typographiques dans la notation

La présentation de ce module fait intervenir divers paramètres, variables et symboles qui représentent des quantités de natures différentes. Afin de rendre plus claire la distinction entre ces différents types de quantités, nous nous servons de typographies différentes dans leur notation. Les quantités géométriques (point, vecteur, repère, transformation, distance, angle) sont notées en italiques ; des exemples figurant dans la suite sont P_g, $\vec{V}_N$, $\{W\}$. Les droites et les courbes sont identifiées en caractères calligraphiques ; des exemples sont $\mathcal{D}_l$, $\mathcal{P}_d$. Dans des expressions algébriques, les variables et coefficients sont notés en italiques tandis que les noms de fonctions sont en caractères droits ; un exemple est $\mathrm{dist}(z_1, z_2)$. Les structures de données (scalaires, vecteurs de scalaires) utilisées dans diverses fonctions de ce module sont notées en caractères droits ; des exemples sont $\mathrm{Dlat_g}$, Glarg. Enfin, les variables figurant seulement au sein de la description d'algorithmes sont notées en caractères dactylographiques ; des exemples sont `cote`, `dist`.

Si le nom d'une quantité comporte un indice, la typographie de celui-ci sera normalement identique au reste du nom, sauf si l'indice se réfère à un autre type de quantité. Le tableau 4.9 résume les conventions typographiques employées dans la notation des paramètres, variables et symboles dans le reste de ce chapitre.

Tableau 4.9 : Conventions typographiques dans la notation

type de quantité	typographie
quantité géométrique	$Quantite_{ind}$
droite ou courbe	$\mathcal{D}_{ind}$ $\mathcal{C}_{ind}$
expression algébrique	fonction(var_1, var_2)
structure de donnée	Structuredonn
variable au sein d'un algorithme	`variable`

4.4.2 Critères importants

4.4.2.1 Position correcte

Pour que les mesures d'arpentage soient interprétées correctement lors de la reconstruction de la géométrie de la mine, il faut que les séparations entre les éléments topologiques soient cohérentes avec le modèle GNOMine. Or, ce sont les *référentiels locaux* (notion définie à la section 3.4.3 du chapitre 3) qui indiquent les endroits précis de raccordement entre galeries et intersections. La figure 4.5 illustre la question de l'emplacement du référentiel local à l'intérieur d'une jonction.

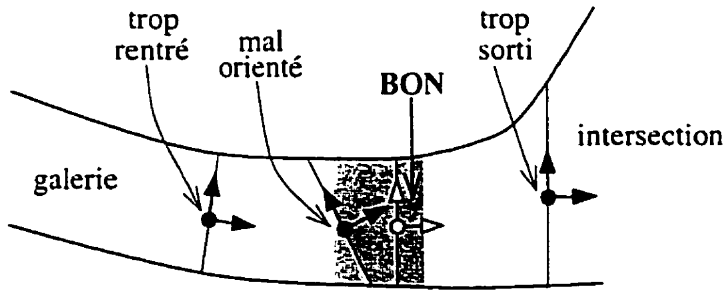


Figure 4.5 : Position du référentiel local dans la jonction

La région ombragée dans la figure 4.5 indique la *zone acceptable*[3] pour la position

[3]La *zone acceptable* d'une jonction est une notion relative ; c'est la partie de la jonction où, en se déplaçant vers la galerie, la variation de la largeur mesurée devient négligeable.

du référentiel local (que nous dénoterons souvent "RL" ci-après). D'une part, le RL peut être placé trop loin vers l'intersection, dans la zone de la jonction où les parois accusent un élargissement important (voire une discontinuité s'il y a occlusion de parois) : ceci affectera la largeur modélisée de la galerie. D'autre part, le RL peut être trop "rentré" dans la galerie ; ceci affectera la longueur modélisée de la galerie. Pour l'orientation du RL, nous voulons que celle-ci reflète l'orientation de l'axe de la galerie à son extrémité.

4.4.2.2 Stabilité géométrique

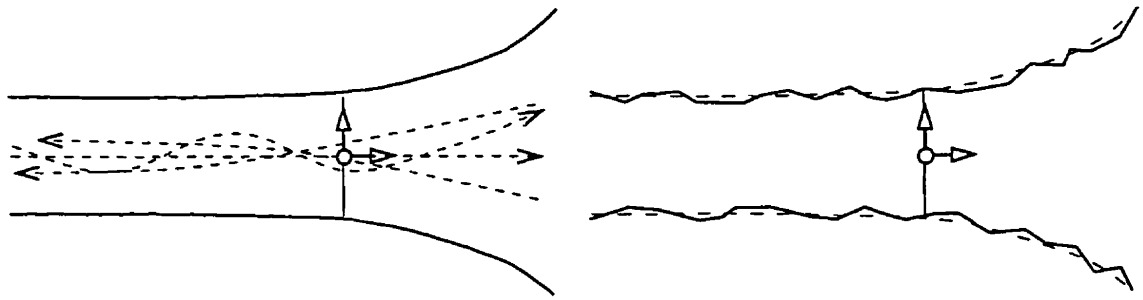
Un deuxième critère primordial dans la conception de ce module est la stabilité géométrique des RLs détectés. Idéalement, il faut extraire de la structure des parois de la jonction une caractéristique géométrique qui soit *invariante* face à la répartition des points mesurés par le robot sur les parois. Les principaux facteurs qui influencent la répartition des mesures sont

- le bruit et les erreurs de mesure introduits par le principe sensoriel ;
- la densité des mesures d'arpentage prises (dans certaines limites) ;
- la trajectoire suivie par le véhicule (*cf.* figure 4.6(a)).

Aussi, nous voudrions que le RL détecté soit fonction de la forme des parois à une échelle globale, mais qu'il ne soit pas trop affecté par les aspérités locales des parois (figure 4.6(b)).

4.4.2.3 Chemin suivi dans une intersection

L'invariance du RL détecté n'est possible que dans la mesure où le robot observe *toute* la structure d'une jonction. Dans une galerie, ceci est toujours vrai car le robot traverse forcément les jonctions aux deux extrémités. Par contre, dans une intersection de degré $n \geq 3$, le robot ne traverse que deux des jonctions, à savoir la



(a) face aux variations de trajectoire. (b) face aux aspérités locales.

Figure 4.6 : Stabilité du RL détecté

jonction d'entrée et la *jonction de sortie* correspondant au passage par l'intersection. Comme l'indiquent les exemples de la figure 4.7, les autres jonctions ne sont observées que partiellement. Il est normal dans de tels cas que le RL ne soit pas placé au même endroit que si la jonction avait été observée complètement.

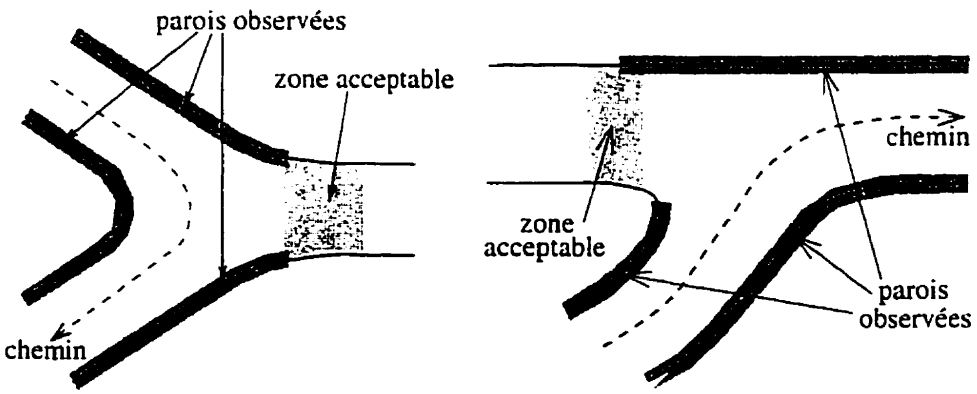


Figure 4.7 : Observations partielles de jonctions

Pour la création de la carte métrique locale de l'intersection, ceci ne pose pas de problème car le référentiel principal est associé à la jonction d'entrée, qui est observée au complet. De plus, le relocalisateur métrique est conçu en tenant compte de cette

variabilité du RL associé à une jonction : la position dans la carte métrique existante du RL "re-détecté" n'est considérée qu'approximative, d'où la nécessité d'une correction itérative. (Si les RLs pouvaient être détectés de manière invariante partout dans la mine, alors la relocalisation métrique deviendrait une tâche triviale.) Expérimentalement, nous ne traiterons la détection de RLs que dans les jonctions d'entrée et sortie d'une intersection.

4.4.2.4 Processus rétroactif

Nous verrons au chapitre 5 que le module détecteur de transitions (dans la navigation globale) génère deux types de signaux de transitions différents :

la notification de transition exploite le senseur frontal pour alerter les autres modules de l'approche d'un nouvel élément topologique ;

la confirmation de transition exploite les mesures latérales pour confirmer aux autres modules la transition à un nouvel élément topologique.

Nous pouvons maintenant donner une nouvelle définition, celle-ci perceptuelle, d'une jonction :

> *définition* : une jonction est formée des points de deux parois, à gauche et à droite du véhicule, mesurés entre les instants de notification et de confirmation d'une transition entre éléments topologiques.

D'autre part, la détection du RL peut être vue comme un processus rétroactif qui analyse les mesures accumulées depuis la notification de transition, mais qui n'est activé que lorsque ce module reçoit la confirmation de transition.

4.4.3 Notion importante : le segment minimal

4.4.3.1 Définition dans un milieu simplifié

Nous cherchons à identifier une caractéristique de la jonction qui soit fonction uniquement de la forme des parois et non pas de la répartition des points de mesure accumulés lors d'un passage particulier du robot. De plus, cette caractéristique doit être en accord avec notre modèle de décomposition topologique de la mine.

Supposons pour l'instant que nous avons affaire à un environnement simplifié, où les parois sont parfaitement lisses et les galeries sont rigoureusement droites et de largeur constante. Prenons une jonction dans laquelle la galerie aboutit dans l'intersection par un élargissement graduel des parois, comme dans la figure 4.8. Nous procédons alors à l'algorithme suivant :

1. **Choisir comme point de départ $P_g = E_g$:**
   partir de l'extrémité de la paroi gauche du côté de l'intersection.

   `dist` = infini ; `cote` = droite ;

2. **Si (`cote` == droite) :** *recherche sur la paroi droite.*

   - **trouver le point P_d le plus proche de $P_g \Longrightarrow$ segment $[P_g P_d]$;**

   Sinon : *recherche sur la paroi gauche.*

   - **trouver le point P_g le plus proche de $P_d \Longrightarrow$ segment $[P_g P_d]$;**

   $\Delta\overline{GD}$ = `dist` $- \overline{P_g P_d}$; `dist` = $\overline{P_g P_d}$;

3. **Si $[P_g P_d]$ a bougé :** *Raccourcissement* $\Delta\overline{GD} > 0$.

   - **changer de côté :** `cote` : droite (gauche) $\Rightarrow$ gauche (droite) ;
   - **remonter à l'étape 2.**

   Sinon : *le segment $[P_g P_d]$ n'a pas bougé* ($\Delta\overline{GD} == 0$).

- **solution $[GD] = [P_g P_d]$; $\Longrightarrow$ fin de l'algorithme.**

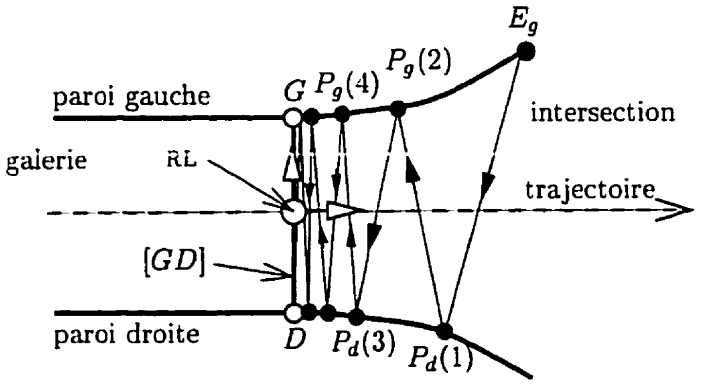


Figure 4.8 : Algorithme itératif de recherche du segment minimal

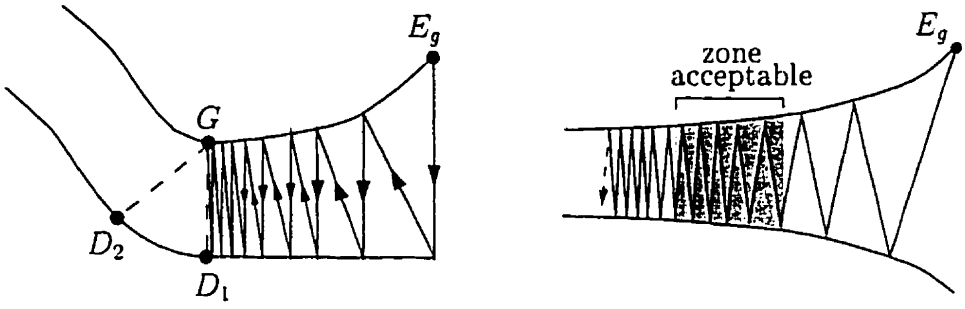
Nous appelons *segment minimal* de la jonction la solution $[GD]$ de l'algorithme itératif précédent. Dans l'environnement simplifié (parois lisses, galeries droites de largeur constante), le segment minimal a la propriété d'avoir une solution unique qui se situe à la frontière de la zone de largeur constante (l'extrémité de la galerie) et dont l'orientation est perpendiculaire à celle de l'axe de la galerie. Le RL associé à la jonction est alors défini comme le repère cartésien situé au milieu du segment $[GD]$ et d'axe Z perpendiculaire à celui-ci (voir figure 4.8).

4.4.3.2 Projection vers l'environnement réel

Plusieurs difficultés se présentent lorsque nous projetons la notion de segment minimal vers l'environnement *réel*, c'est-à-dire dont les parois sont rugueuses, où les galeries ne sont ni rectilignes ni de largeur constante et où les parois d'une jonction peuvent accuser non pas un élargissement graduel mais plutôt des discontinuités de largeur dues aux occlusions entre parois dans une intersection.

Les schémas 4.9(a) et 4.9(b) illustrent deux situations où notre algorithme préliminaire de recherche du segment minimal s'avère insuffisant. Dans le premier cas, l'extrémité de la galerie est de forme telle qu'il en résulte deux solutions possibles pour

le segment minimal, entre le point G et deux points D_1 ou D_2 qui lui sont équidistants. Dans le deuxième cas, nous avons une galerie qui commence à s'élargir à un taux faible avant son extrémité, pour aboutir dans un élargissement plus prononcé à l'intersection. Ici, l'algorithme dépasse la zone acceptable pour le RL car les segments trouvés continuent à se raccourcir ($\Delta\overline{GD} > 0$) même si la largeur entre parois ne diminue que faiblement[4].



(a) Solution non-unique. (b) Solution rentre trop loin dans la galerie.

Figure 4.9 : Difficultés pour la recherche du segment minimal

Ainsi, il y aura de nombreuses conditions réelles où la propriété du segment minimal dans le milieu idéalisé ne sera plus vérifiée. Il est évident que nous devons modifier l'algorithme de base décrit précédemment afin de satisfaire les critères de la position correcte et de la stabilité géométrique du RL.

4.4.3.3 Approche des parois discrétisées

Le développement du détecteur de RLs pourrait consister à appliquer la recherche de segment minimal aux parois de jonction *discrétisées* sous forme de points (z, x) dans le plan. Cependant, cette approche est assujettie à plusieurs faiblesses inhérentes qui

[4]Ce cas de figure révèle un problème inhérent à la notion de segment minimal : qu'en général, bien que la *longueur* $\overline{GD}$ de la solution est relativement stable, sa *position* dans la jonction ne l'est pas forcément.

nuiraient à la stabilité des résultats.

Principalement, le fait de laisser les parois sous forme de points de mesure rend cette approche trop sensible aux deux facteurs combinés de la rugosité des parois et de la trajectoire suivie. La figure 4.10 montre l'influence de la trajectoire sur la répartition des points de mesure. Dans le premier cas, le robot suit une trajectoire idéalisée à travers une intersection, et dans le deuxième cas la trajectoire est fortement déviée. La distribution des points de mesure sur les parois devient très irrégulière dans le deuxième cas. De plus, des variations dans l'orientation du véhicule peuvent causer des "boucles" d'échelles différentes dans l'ordre de mesure des points (figure 4.10(c)).

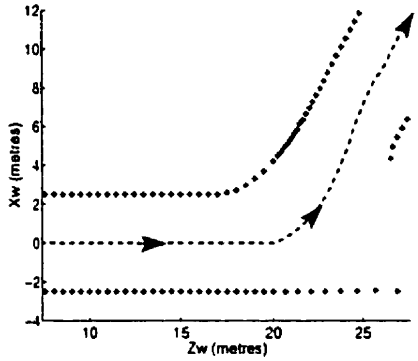
D'autre part, la discrétisation de parois rugueuses peut faire que deux passages par la même jonction ne mesurent pas exactement les mêmes aspérités. Le segment minimal pourrait donc converger vers deux endroits différents.

Un autre problème considérable pour l'approche discrète est la formation d'un "trou" de mesures dans l'une des parois de jonction lorsque le robot doit effectuer un virage de plus de 90° dans une intersection. Nous avons deux exemples de ce phénomène dans la figure 4.11. D'une part, ceci signifie que même dans une jonction d'entrée ou sortie, l'observation des parois peut n'être que partielle. Mais en plus, le facteur de la trajectoire suivie peut faire varier la longueur de la partie invisible de la paroi ; le segment minimal serait contraint à une partie de la jonction qui change selon le passage du robot.

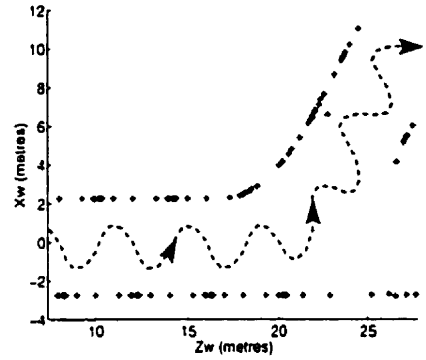
4.4.3.4 Approche des parois continues

Face aux difficultés d'obtenir une solution stable avec des parois discrétisées, nous adoptons une deuxième approche basée sur des parois *continues*. Essentiellement, il s'agit de faire deux choses :

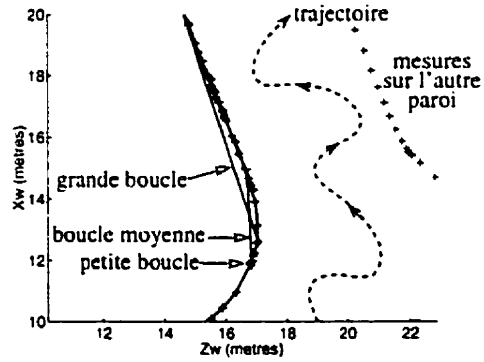
1. Modéliser les deux parois de jonction par des courbes paramétriques ;
2. Déterminer le segment minimal par la minimisation (sous contraintes) de la longueur du segment en fonction des positions sur les deux parois.



(a) trajectoire idéale.



(b) trajectoire fortement déviée.



(c) "boucles" de points de mesure.

Figure 4.10 : Influence de la trajectoire sur la répartition des mesures

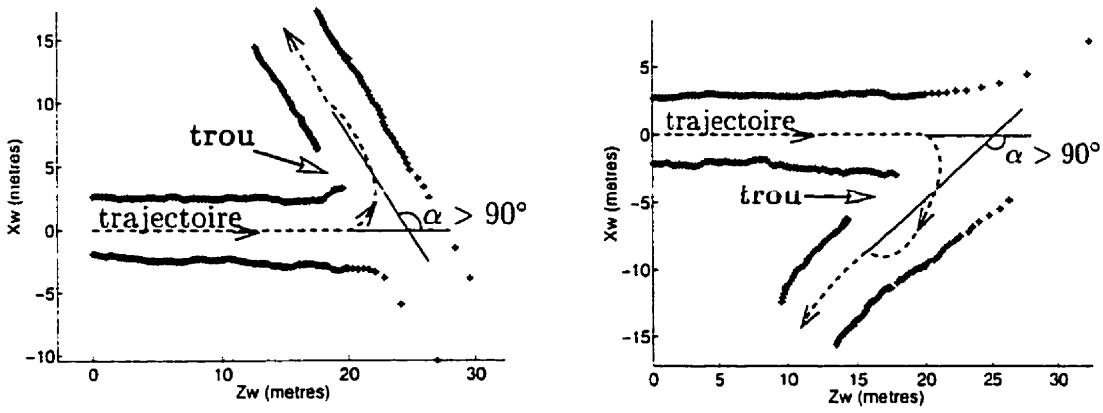


Figure 4.11 : Trous dans les parois dans des virages accentués

Par rapport à l'approche précédente, celle-ci représente trois avantages importants. La représentation des parois par des courbes continues permet en effet de a) lisser les aspérités, b) éliminer l'effet de répartition irrégulière des points et c) interpoler des sections de paroi non-observées lors de virages. Le reste de ce chapitre traitera donc de notre travail sur l'approche basées sur les parois continues.

4.4.4 Décomposition du module en fonctions

Le schéma d'interaction des différentes fonctions du module détecteur de référentiels locaux se trouve à la figure 4.12. Les paragraphes qui suivent décrivent les fonctions (les boîtes dans l'organigramme) en termes d'entrées, sorties et rôle.

4.4.4.1 Arpentage de la jonction

Cette fonction accumule les profils latéraux pendant les instants de notification et de confirmation de transition. Elle convertit ces mesures en listes de points décrivant les deux parois de jonction, de coordonnées $(z, x)_W$ dans le repère correspondant à la position du robot (point M_f) à l'instant de notification de transition (le repère

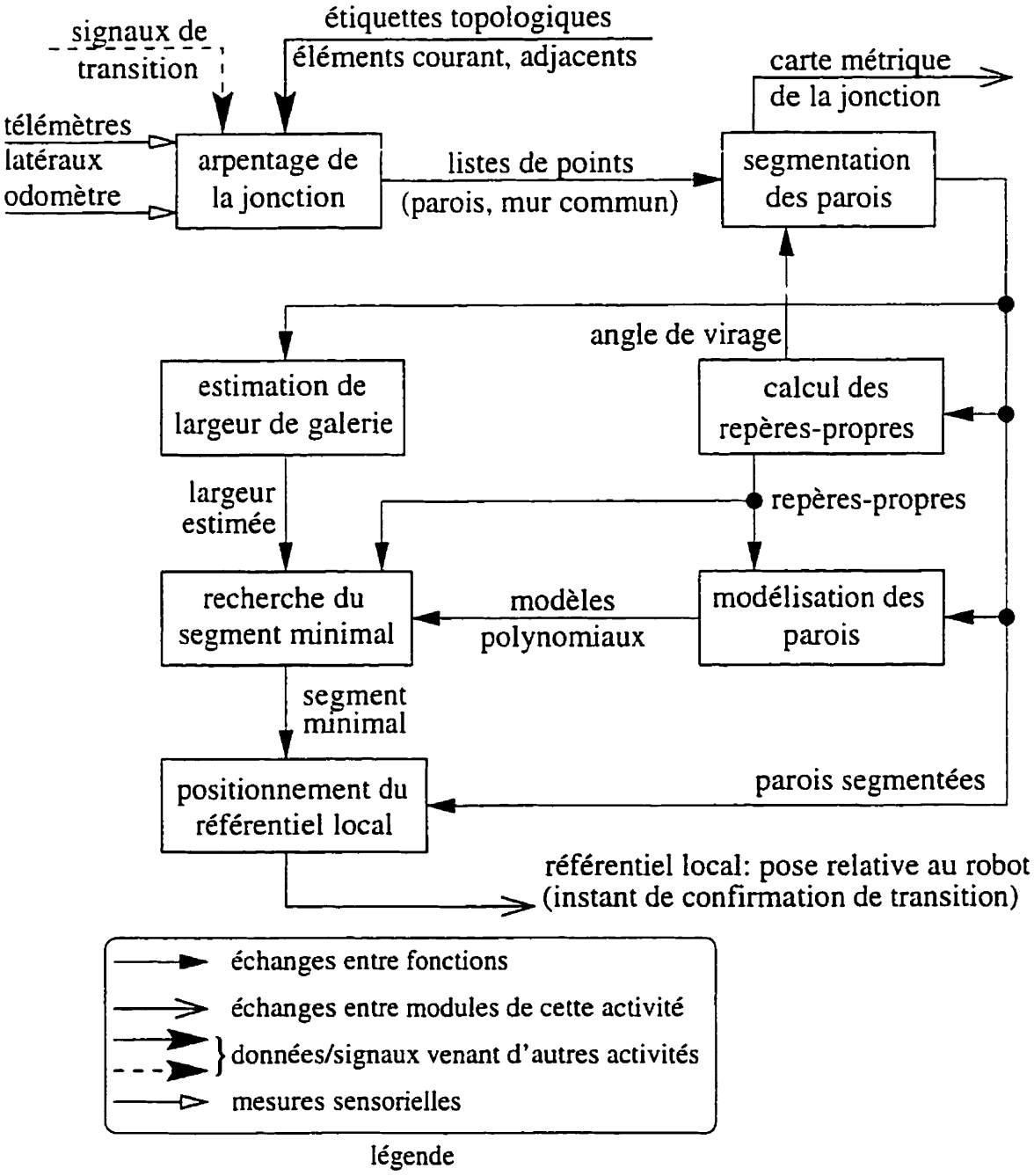


Figure 4.12 : Détecteur de référentiels locaux : échanges de données et signaux

$\{W\}$ pour le processus de détection des RLs). Lorsque les jonctions d'entrée et sortie d'intersection partagent un *mur commun* (*cf.* section 4.5.1), cette fonction accumule les mesures décrivant ce mur jusqu'à la confirmation de transition pour la jonction de sortie (*cf.* tableau 4.10).

Tableau 4.10 : Fonction d'arpentage de la jonction

entrée	- profils télémétriques instantanés latéraux ; - déplacements odométriques entre instants de mesure ; - signaux de transition ; - étiquettes topologiques des éléments courant et adjacents.
sortie	Listes de points $(z,x)_W$, une par paroi de jonction ou mur commun.

4.4.4.2 Segmentation des parois

Cette fonction doit isoler, dans les points de mesure accumulés, les sous-ensembles appartenant physiquement aux parois de jonction et/ou au mur commun (*cf.* tableau 4.11). Elle doit traiter les cas d'occlusion entre parois et de morcellement du mur commun dans des virages prononcés (*cf.* section 4.5.2).

Tableau 4.11 : Fonction de segmentation des parois

entrée	Listes de points $(z,x)_W$ décrivant chaque paroi de jonction ou mur commun.
sortie	Listes de points $(z,x)_W$ décrivant les parois segmentées. $\Longrightarrow$ *Carte métrique de la jonction.*

4.4.4.3 Estimation de largeur de galerie

À partir des mesures décrivant les deux parois de jonction, cette fonction estime la largeur locale de la galerie à son extrémité (*cf.* tableau 4.12). Cette information est

exploitée pour établir des contraintes à l'étape de minimisation.

Tableau 4.12 : Fonction d'estimation de largeur de galerie

entrée	Listes de points $(z,x)_W$ décrivant les parois segmentées.
sortie	Largeur estimée Glarg de la galerie dans la jonction.

4.4.4.4 Calcul des repères propres

Cette fonction associe à chaque paroi mesurée son propre repère pour la modélisation (*cf.* tableau 4.13). Ces repères propres sont spécifiés par leurs poses dans le repère $\{W\}$.

Tableau 4.13 : Fonction de calcul des repères propres

entrée	Listes de points $(z,x)_W$ décrivant les parois segmentées.
sortie	repères propres $\{P_i\}$ associés aux parois.

4.4.4.5 Modélisation des parois

À cette étape, chaque paroi mesurée est modélisée dans son repère propre par une fonction (en fait, un polynôme) décrivant une courbe plane. Les coefficients spécifiant les deux courbes sont fournis à l'étape de minimisation (*cf.* tableau 4.14).

Tableau 4.14 : Fonction de modélisation des parois

entrée	- listes de points $(z,x)_W$ décrivant les parois segmentées ; - repères propres $\{P_i\}$ associés aux parois.
sortie	Modèles (coefficients polynomiaux) des deux courbes.

4.4.4.6 Recherche du segment minimal

Une fois les parois modélisées, le segment minimal de la jonction est maintenant déterminé par un processus de minimisation d'une fonction de la distance entre les deux courbes, avec contraintes. Le résultat est la position du segment minimal (ses deux sommets) dans la jonction (*cf.* tableau 4.15).

Tableau 4.15 : Fonction de recherche du segment minimal

entrée	- modèles (coefficients polynomiaux) des deux courbes ; - repères propres $\{P_i\}$ associés aux parois ; - largeur estimée Glarg de la galerie dans la jonction.
sortie	Sommets du segment minimal dans le repère $\{W\}$.

4.4.4.7 Positionnement du référentiel local

La dernière étape dans ce module est d'associer un RL au segment minimal trouvé : le RL n'est pas spécifié dans le repère $\{W\}$, mais plutôt par sa pose *relative au robot* à l'instant de confirmation de transition. L'orientation du RL doit refléter celle de la galerie à son extrémité (*cf.* tableau 4.16).

Tableau 4.16 : Fonction de positionnement du référentiel local

entrée	- sommets du segment minimal dans le repère $\{W\}$. - déplacements odométriques entre les signaux de notification et confirmation ; - listes de points $(z, x)_W$ décrivant les parois segmentées.
sortie	RL : sa pose relative au robot à l'instant de confirmation de transition.

4.5 Détecteur de référentiels locaux : implantation

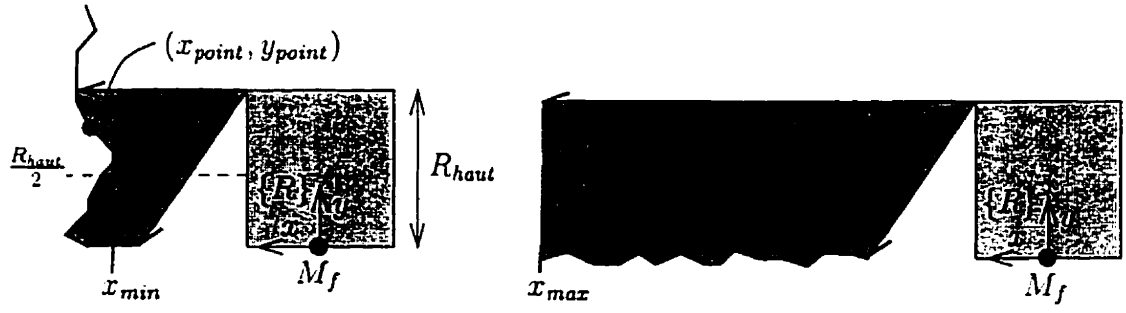
Dans cette section, nous allons détailler les algorithmes développés au sein des fonctions que nous venons d'énumérer.

4.5.1 Arpentage de la jonction

4.5.1.1 Prétraitement des mesures

Lorsque les profils latéraux correspondant à la jonction sont accumulés, nous leur appliquons trois étapes de prétraitement : la conversion en distances latérales Dlat, l'élimination de pics isolés de la largeur mesurée et la transformation des vecteurs $\text{Dlat}_{g,d}$ (des mesures Dlat à gauche et à droite) en points de parois $(z, x)_W$.

Les figures 4.13(a) et 4.13(b) illustrent les deux cas pour le calcul de Dlat. Si une paroi verticale se trouve relativement proche du télémètre latéral (figure 4.13(a)), suffisamment de points du profil tombent sur cette paroi ($y_{point} \geq R_{haut}/2$) ; nous retenons la valeur la plus contraignante en x, x_{min}. Si le robot se trouve dans un espace ouvert (*e.g.* une intersection), tous les points de profil tombent au sol (figure 4.13(b)) ; nous retenons alors la valeur en x la plus éloignée, x_{max}.



(a) dans une galerie.

(b) dans une intersection.

Figure 4.13 : Conversion du profil en distance latérale

Nous avons maintenant deux vecteurs, Dlat_g et Dlat_d, pour les parois mesurées

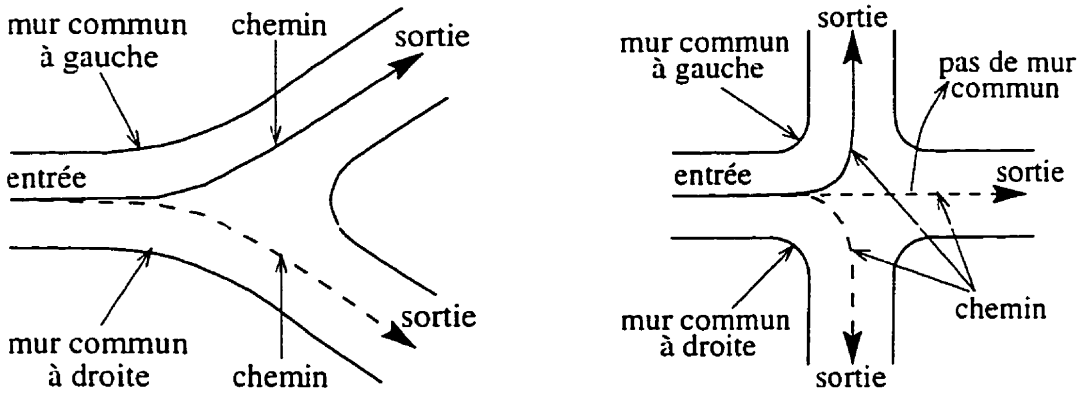
à gauche et à droite du robot. Cependant, la conversion initiale des profils (ainsi que le bruit de mesure télémétrique d'un système réel) introduisent des erreurs dans les valeurs Dlat : essentiellement, des pics isolés de un ou quelques éléments. Nous appliquons un filtrage médian seulement à ces plages limitées dans $\text{Dlat}_{g,d}$ de valeurs élevées.

Enfin, nous associons aux deux vecteurs leurs poses relatives de l'odométrie ; chaque élément est un point de coordonnées $(\text{Dlat}, 0, 0)_R$ dans le repère du robot, que nous transformons en un point $(z, x)_W$ dans le repère $\{W\}$.

4.5.1.2 Sélection du mur commun

Quand le robot traverse une intersection, les jonctions d'entrée et sortie pourront partager physiquement une même paroi verticale, que nous appelons le *mur commun.* Dans une intersection de degré $n = 2$, les deux murs d'intersection sont communs. Si $n = 3$, les deux jonctions ont toujours un mur commun, mesuré à gauche ou à droite du robot (figure 4.14(a)). Lorsque $n > 3$, il y aura un mur commun si les deux jonctions sont consécutives dans l'ordonnancement des galeries rattachées à l'intersection (figure 4.14(b)).

Cette fonction exploite donc de l'information provenant du navigateur topologique pour savoir s'il y a un mur commun. Si tel est le cas, les mesures décrivant cette paroi sont accumulées durant la traversée de *toute* l'intersection (depuis la notification d'entrée jusqu'à la confirmation de sortie). À la sortie de l'intersection, le robot fera la détection du RL de sortie et refera celle du RL d'entrée, en se basant sur le mur commun entier au lieu des tronçons séparés des deux jonctions. La raison principale est que cela nous permet d'interpoler adéquatement la partie non-observée de la paroi dans des virages accentués (*cf.* figure 4.11) à l'étape de modélisation des parois.



(a) Degré $n = 3$.

(b) Degré $n = 4$.

Figure 4.14 : Sélection du mur commun d'intersection

4.5.2 Segmentation des parois

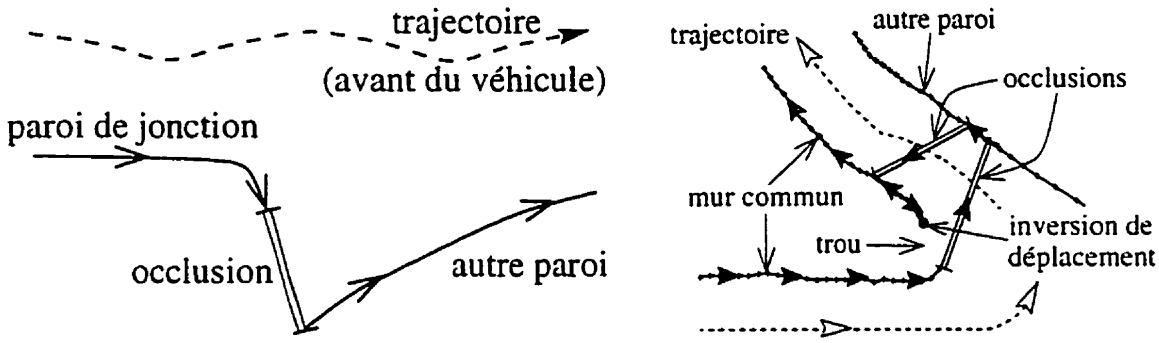
Dans le cas général, les points mesurés d'un côté du véhicule n'appartiennent pas tous à la paroi qui nous intéresse. L'objectif à cette étape est d'éliminer les parties des deux listes de points qui ne décrivent pas les parois voulues et de redresser l'ordre des points dans les listes. Pour ce faire, nous employons les deux processus de segmentation (basée sur les occlusions) et d'ordonnancement des points de mesure.

4.5.2.1 Segmentation

Le processus de segmentation d'une liste de points se base sur la détection de discontinuités (changements brusques) de position dues aux occlusions de parois éloignées par d'autres plus rapprochées. Nous imposons un *seuil* de détection : lorsque deux points consécutifs sont séparés d'une distance $d \geq d_{occl}$, nous considérons qu'il s'agit d'une occlusion.

Dans le cas d'une paroi de jonction, nous voulons supprimer la partie des mesures qui est au-delà d'une occlusion éventuelle (figure 4.15(a)). Pour un mur commun, la nécessité de segmenter les mesures survient lorsque le robot effectue un virage de

plus de 90° dans l'intersection[5]. Dans cette situation, la liste de points se conforme qualitativement au schéma de la figure 4.15(b) : elle comporte deux discontinuités, une partie centrale qui tombe sur une autre paroi et des inversions du sens de déplacement des mesures (indiqué dans les deux figures). Notons qu'une discontinuité peut devenir une plage de points non-détectés là où la distance latérale dépasse la limite de portée des senseurs.



(a) Paroi de jonction.

(b) Mur commun d'intersection.

Figure 4.15 : Nécessité de segmenter les parois mesurées

Nous avons déjà vu qu'en général la liste de points initiale sera désordonnée ; ceci affectera la segmentation et vice versa. Nous appliquons donc la segmentation à deux reprises, comme suit :

1. Segmentation avec un seuil de détection élevé ;

2. Ordonnancement de la sous-liste de points gardée ;

3. Re-segmentation de la sous-liste ordonnée, avec un seuil plus petit.

[5] L'angle approximatif de virage entre les jonctions d'entrée et sortie provient du calcul des repères propres. Il est équivalent à la différence entre les orientations des axes associés aux deux moitiés du mur commun (voir section 4.5.4).

4.5.2.2 Ordonnancement des points

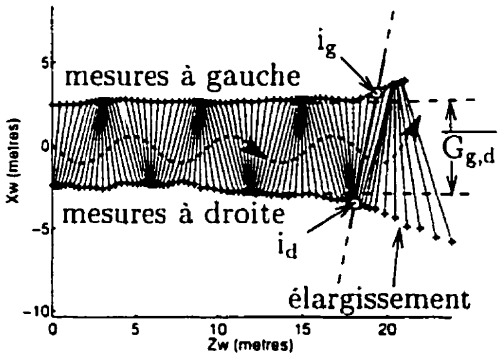
Le processus d'ordonnancement doit corriger l'ordre des points de paroi dans une liste qui peut comporter des boucles de mesures (*cf.* figure 4.10(c)) ou des inversions du sens de déplacement des mesures. Nous cherchons à rétablir *l'ordre idéal* des points le long de la paroi, dans un sens de parcours correspondant à la variation monotone du paramètre d'une courbe qui modéliserait la paroi.

L'algorithme de base ressemble à un tri de nombres et vise à minimiser la somme des longueurs des segments reliant les paires de points adjacents dans la liste. Les deux premiers points sont laissés tels quels ; la distance qui les sépare est la longueur initiale l_N (N=2) d'une nouvelle liste. Ensuite, chaque point successif (d'indice $N = 3, 4, \ldots$ dans la liste originale) est rajouté aux précédents à la position O_N qui minimise l'augmentation $(l_N - l_{N-1})$ de la longueur totale des segments de la nouvelle liste. Le vecteur $\hat{O}$ formé des indices O_N décrit la liste réordonnée.

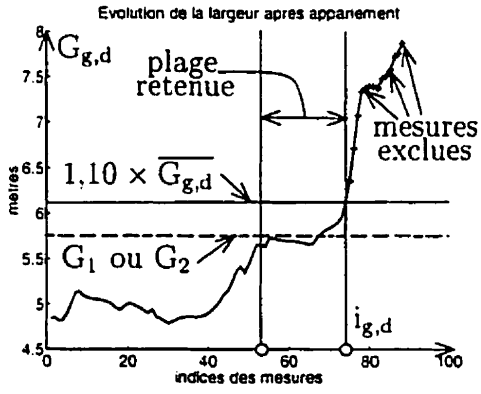
Cet algorithme ne parvient pas cependant à rétablir l'ordre idéal des points dans tous les cas ; il corrige bien les boucles de mesures (d'échelle locale dans la liste) mais peut laisser des inversions du sens de déplacement des mesures sur la paroi, comme dans la situation de la figure 4.15(b). Pour répondre à ce problème, nous permettons au réordonnancement de fonctionner en plusieurs passes. Chaque nouvelle passe doit améliorer une mesure de qualité qui est l'homogénéité de la "tendance générale" du vecteur $\hat{O}$ d'indices réordonnés. C'est-à-dire qu'à une échelle grossière, les indices dans $\hat{O}$ (auxquels nous appliquons un filtrage médian pour réduire les variations locales dues à la correction de boucles) devraient monter ou descendre de manière monotone. Nous estimons que la qualité du réordonnancement est suffisante si le vecteur $\hat{O}$ filtré varie de manière monotone sur au moins 3/4 de ses éléments. Le processus s'arrête lorsque ce critère est satisfait ou que la dernière passe n'a pas amélioré la qualité, auquel cas nous revenons au vecteur $\hat{O}$ de la passe précédente.

4.5.3 Estimation de largeur de galerie

Nous avons crée, à l'étape de prétraitement, deux vecteurs $Dlat_g$ et $Dlat_d$ des valeurs Dlat mesurées dans la jonction. Pour estimer la largeur locale de la galerie à son extrémité, nous pouvons simplement prendre la moyenne de la somme des éléments dans $Dlat_g$ et $Dlat_d$. Mais le résultat comporte alors deux sources d'imprécision : il inclut une partie de la jonction (vers l'intersection) qui ne correspond plus à la galerie et la correspondance entre mesures à gauche et à droite est influencée par la trajectoire suivie (figure 4.16(a)) alors que la valeur Glarg calculée ne devrait pas dépendre de ce facteur. Nous devons donc faire deux choses : ré-apparier les points sur les deux parois et exclure la partie de la jonction qui s'élargit trop par rapport à la galerie.



(a) Correspondance initiale entre mesures à gauche et à droite.



(b) Plage retenue dans $\widehat{G_g}$, $\widehat{G_d}$.

Figure 4.16 : Traitement des mesures pour estimer Glarg

L'appariement est basé sur le critère de distance minimale : pour chaque point, nous trouvons le point le plus proche sur l'autre paroi dans une fenêtre limitée de recherche. Nous obtenons deux nouveaux vecteurs, $\widehat{G_g}$ et $\widehat{G_d}$, formés des distances entre les points appariés et leurs moyennes $\overline{G_g}$ et $\overline{G_d}$. Soit i_g (respectivement i_d)

l'indice du premier élément (en partant de l'intersection) de $\widehat{G_g}$ (respectivement $\widehat{G_d}$) éloigné de moins de 10% de $\overline{G_g}$ (respectivement $\overline{G_d}$). Nous calculons G_1 (respectivement G_2) comme étant la moyenne des éléments de $\widehat{G_g}$ (respectivement $\widehat{G_d}$) dans la plage commençant à i_g (respectivement i_d) et de longueur équivalente sur la paroi[6] $\overline{G_g}$ (respectivement $\overline{G_d}$) (figure 4.16(b)). La valeur finale de Glarg est prise comme la plus petite de G_1 et G_2. Expérimentalement, cette méthode pour estimer Glarg s'avère relativement robuste par rapport aux variations de trajectoire.

4.5.4 Calcul des repères propres

4.5.4.1 Axes des abscisses

Il faut d'abord déterminer un axe des abscisses pour chaque paroi de jonction. D'une part, l'orientation de l'axe doit être telle que les points de paroi décrivent autant que possible une fonction analytique[7] pour l'étape de modélisation. La figure 4.17 illustre cette idée. D'autre part, l'orientation de l'axe doit être déterminée par une méthode qui soit robuste face à la répartition des mesures sur la paroi et aux points éventuellement erronés.

Pour chaque liste de points $(z, x)_W$, nous calculons une droite $x_W = a + b\, z_W$ selon un critère de moindres déviations absolues. L'algorithme est tiré de Press, Flannery, Teukolsky et Vetterling (1988) ; il s'agit d'une méthode statistiquement plus robuste à la présence de points erronés que la régression basée sur une le critère des moindres carrés. Nous appelons les deux droites obtenues les *axes médians* des deux parois. Le repère propre $\{P\}$ associé à la paroi est placé sur l'axe, au centre du domaine de

[6]"longueur équivalente sur la paroi" : pour n points de mesures ou éléments dans un vecteur, cette longueur vaut : $(n-1) \times d_{arp}$. Inversement, le nombre de points dans une plage de mesures de longueur équivalente l est : $n = \text{floor}(l/d_{arp}) + 1$. De plus, d_{arp} est la distance parcourue par le véhicule entre deux instants de mesure (supposée constante). Elle représente la distance "moyenne" entre deux points consécutifs sur une paroi, hormis le relief et les effets de changement de direction du véhicule.

[7]Une fonction $y = f(x)$ est dite *analytique* si, pour chaque abscisse x_i, il existe au plus une ordonnée $y_i = f(x_i)$. Les points de mesure sur une paroi sont considérés comme appartenant à une telle fonction, hormis les effets de la rugosité.

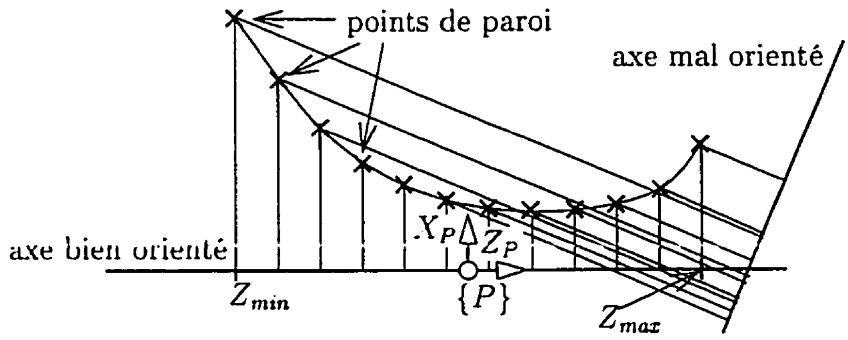


Figure 4.17 : Axe médian et repère propre d'une paroi de jonction

définition $[Z_{min}, Z_{max}]$.

4.5.4.2 repère propre d'un mur commun

Les parties du mur commun arpentées dans les jonctions d'entrée et sortie d'une intersection peuvent être vues comme ses deux "moitiés". Soit $\mathcal{D}_1$ et $\mathcal{D}_2$ les axes médians associés aux deux moitiés du mur commun et θ_1 et θ_2 leurs orientations par rapport à l'axe Z_W (figure 4.18). De plus, soit P l'intersection des deux droites et P_c un point "à l'intérieur" du mur commun, calculé comme le centroïde des points des deux moitiés du mur. Si le mur commun est presque rectiligne ($|\theta_1 - \theta_2| < 5°$), l'axe des abscisses est pris comme la moyenne des deux droites $\mathcal{D}_1$ et $\mathcal{D}_2$; l'origine de $\{P\}$ est alors la projection de P_c sur cette droite. Si par contre ($|\theta_1 - \theta_2| \geq 5°$), nous calculons les bissectrices $\mathcal{B}_D$ et $\mathcal{B}_N$ à $(\mathcal{D}_1, \mathcal{D}_2)$, d'orientations θ_D et θ_N. Nous devons identifier les bissectrices "intérieure" et "extérieure" au mur commun. L'intérieure est celle dont le vecteur directeur ($\vec{V}_D$ ou $\vec{V}_N$, voir la figure 4.18) se trouve dans le même quadrant du plan découpé par $(\mathcal{D}_1, \mathcal{D}_2)$ que P_c. Nous prenons la bissectrice extérieure comme axe des abscisses et le point P comme origine du repère propre.

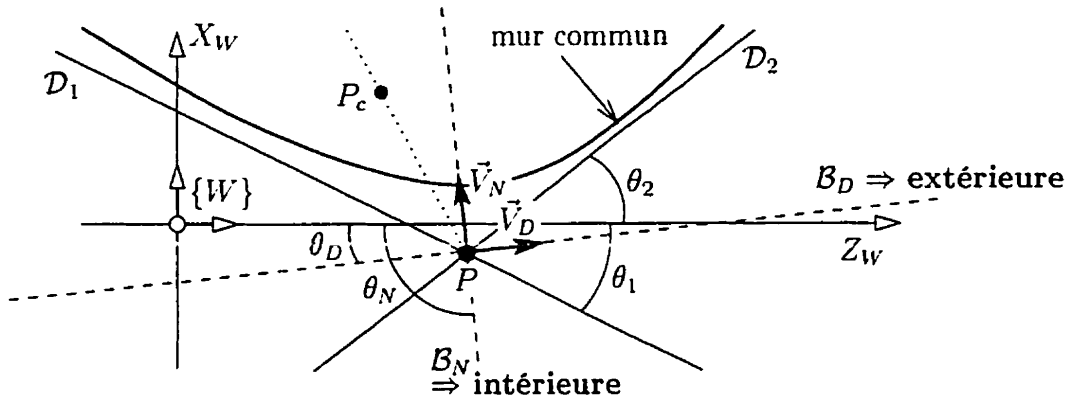


Figure 4.18 : repère propre d'un mur commun

4.5.5 Modélisation des parois

4.5.5.1 Choix d'une classe de courbes

Afin d'assurer la stabilité de la détection des RLs, plusieurs critères s'imposent quant au choix de la classe de courbes qui modélisera les parois. Nous avons besoin d'un modèle qui soit :

- capable de représenter des parois de forme non-symétrique et qui comportent des "coins" (sections de courbure élevée à l'échelle du véhicule) et/ou des changements de courbure ;
- robuste aux translations et rotations (dans certaines limites) du repère propre trouvé ;
- robuste au bruit (et erreurs) de mesure et à la répartition des points sur la paroi ;
- robuste à la présence de trous (parties non-visibles) dans les parois.

Nous avons testé quelques classes de courbes en générant des mesures avec une fonction non-symétrique additionnée de bruit et en évaluant la qualité de la reconstruction

de courbe soumise aux facteurs énumérés ci-dessus. Un modèle qui s'avère satisfaisant est celui des courbes *polynomiales*, c'est-à-dire :

$$x_P = \mathcal{P}_d(z_P) = \sum_{i=0}^{d} p_i \, z_P^i \ , \qquad d \ : \text{degré du polynôme.} \tag{4.1}$$

Un polynôme de degré 5 est suffisant pour représenter la plupart des parois rencontrées. Dans certains cas, notamment pour un mur commun qui comporte un coin aigu (changement d'orientation $\geq 90°$ entre les galeries d'entrée et sortie), un polynôme de degré 9 est nécessaire pour modéliser correctement la forme de la paroi.

4.5.5.2 Processus de modélisation

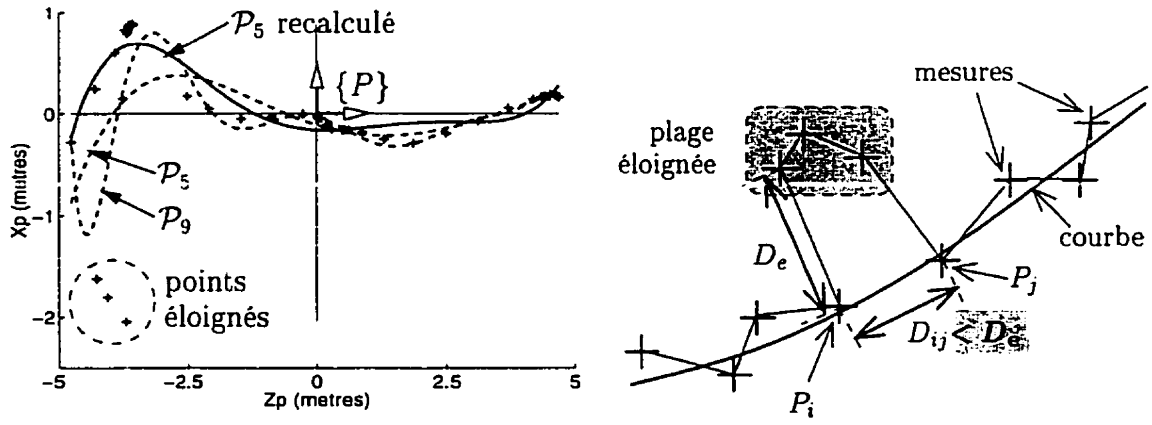
Soit $(z, x)_W$ un point de mesure, $P = (z, x)_P$ ce même point exprimé dans le repère propre et $P' = (\mathcal{P}_d(z_P), z_P)$ le point de même abscisse sur la courbe reconstruite. Les coefficients p_i du polynôme sont calculés par régression non-linéaire sur l'ensemble des M points de paroi. Nous commençons par un polynôme de degré $d = 5$. Les deux critères d'évaluation du modèle trouvé sont basés sur les déviations absolues entre les paires (P, P') ; à savoir, si

$$\frac{1}{M} \sum_{p=1}^{M} |\mathcal{P}_5(z_p) - x_p| \leq d_{arp} \quad \text{ET} \quad \max(|\mathcal{P}_5(z_p) - x_p|, p = 1, \ldots, M) < 1 \text{ mètre.}$$

alors la courbe pour $d = 5$ est adéquate. Sinon, nous remodélisons la paroi avec $d = 9$.

Dans certains cas, des points de mesure erronés (ou éventuellement des aspérités d'amplitude importante) loin de la courbe $\mathcal{P}_5$ peuvent forcer à eux-seuls la modélisation à $d = 9$. La figure 4.19(a) illustre ce genre de situation. Pour le corriger, nous isolons d'abord tous les points P de déviation de l'ordre du mètre. Parmi ceux-là, nous supprimons les plages de points consécutifs éloignés qui sont cernées par des points (P_i, P_j) relativement rapprochés (figure 4.19(b)). Ensuite, nous recalculons la courbe $\mathcal{P}_5$, sans ces plages, avant de passer aux critères d'évaluation.

Dans la figure 4.20, nous avons l'exemple de la modélisation des parois, dont un



(a) Influence des points éloignés sur le modèle.

(b) Plage de points à supprimer.

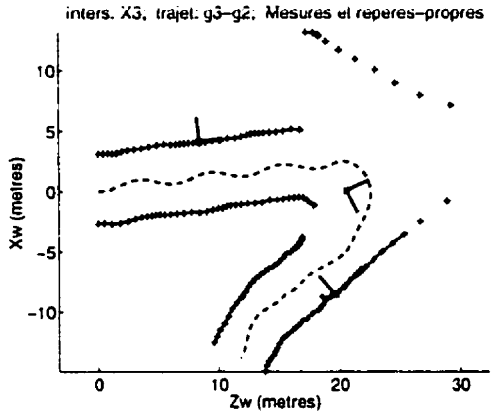
Figure 4.19 : Suppression des points très éloignés de $\mathcal{P}_5$

mur commun qui comporte un trou, mesurées lors de la traversée d'une intersection à 4 branches. Le premier graphique (4.20(a)) montre l'ensemble des mesures, la trajectoire (du point M_f à l'avant du robot) et les repères $\{P_i\}$. Les trois autres graphiques montrent les deux parois de jonctions et le mur commun, dans leurs repères propres, avec leurs courbes polynomiales.

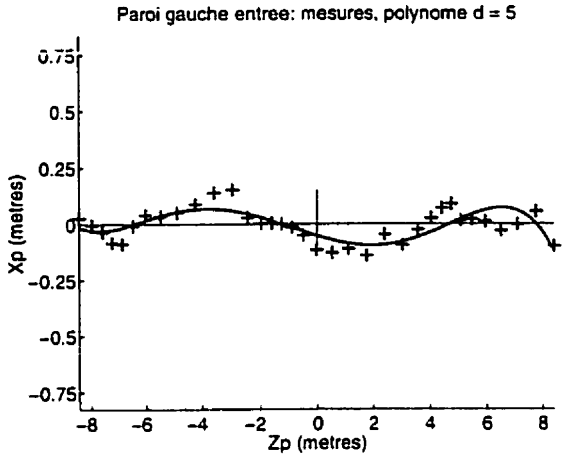
4.5.6 Recherche du segment minimal

4.5.6.1 Conditions initiales

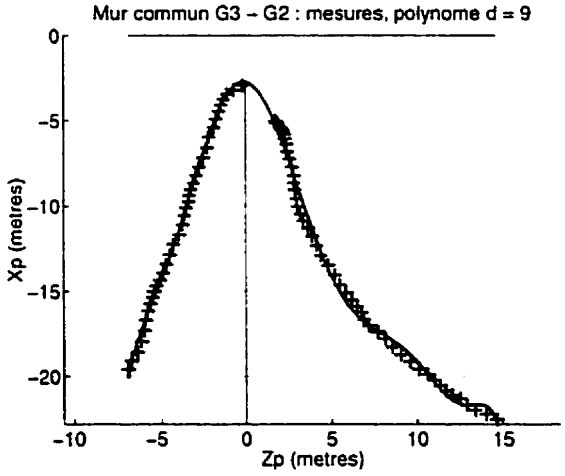
Les conditions initiales pour la minimisation sont les abscisses de départ sur les deux courbes C_1 et C_2 et les directions de recherche initiales. Soit une jonction formée d'une paroi de jonction à gauche (C_1) et d'un mur commun à droite (C_2), comme dans la figure 4.21. Le point P_1 (sur C_1) est à la limite de l'intervalle $[Z_{min}, Z_{max}]_1$ correspondant à l'intersection. Le point P_{com} (sur C_2) est trouvé soit comme l'extremum des x_2 (si le mur commun comporte un coin), soit comme le point le plus proche de P_1 (si le mur commun est proche de rectiligne). Puis, nous remplaçons le point P_1 par



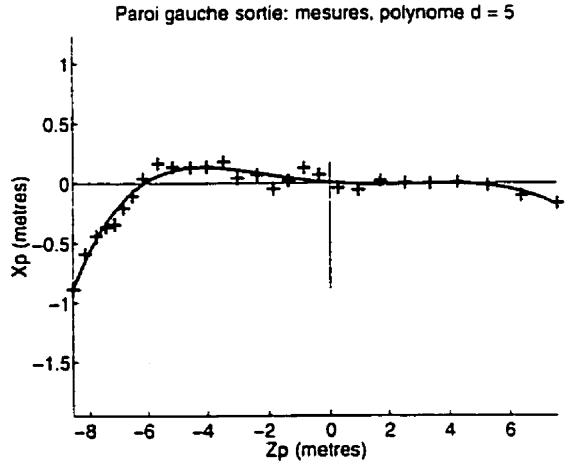
(a) Mesures et repères propres trouvés.



(b) Paroi gauche d'entrée ($d = 5$).



(c) Mur commun ($d = 9$).



(d) Paroi gauche de sortie ($d = 5$).

Figure 4.20 : Exemple de modélisation de parois ($d_{arp} = 50$ cm)

le point le plus proche de P_{com}, P_j. Le segment $[P_j\, P_{com}]$, d'abscisses (z_{1i}, z_{2i}) sur les deux courbes, est le segment de départ pour la recherche du segment minimal. Dans d'autres types de jonctions, les points de départ sur les deux courbes sont identifiés par la combinaison appropriée des méthodes ci-dessus.

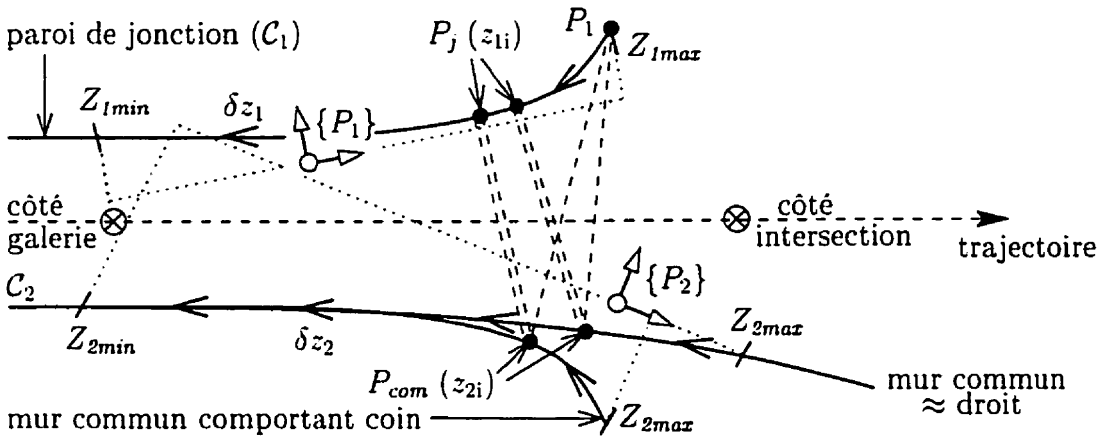


Figure 4.21 : Abscisses initiales, directions de recherche et bornes en z_1, z_2

Les directions initiales de recherche doivent faire varier z_1 et z_2 de manière à amener le segment *vers l'intérieur de la galerie.* Nous exploitons les poses du robot aux instants de notification et confirmation de transition (indiqués dans l'exemple de la figure 4.21) pour déterminer δz_1 et δz_2.

4.5.6.2 Fonction de distance à minimiser

Les expressions de la courbe polynomiale et de sa dérivée en z_c, pour les courbes C_1 et C_2 de degré d_c $(c = 1, 2)$, sont

$$x_c(z_c) = \mathcal{P}_{d_c}(z_c) = \sum_{i=0}^{d_c} p_{ic}\, z_c^i\,, \qquad \dot{x}_c(z_c) = \frac{d}{dz_c}\mathcal{P}_{d_c}(z_c) = \sum_{i=1}^{d_c} i\, p_{ic}\, z_c^{(i-1)}. \qquad (4.2)$$

La fonction de distance que nous allons minimiser est la longueur (au carré) du segment $[P_1, P_2]$ (abscisses z_1 et z_2) joignant les deux courbes. Pour calculer cette

distance, nous transformons d'abord le point $P_2 = (z_2, x_2)^T$, sur C_2, du repère $\{P_2\}$ vers $\{P_1\}$. Cette transformation géométrique est décomposable en une rotation et une translation :

$$ {}^1_2T \cdot P_2 = {}^1_2R \cdot P_2 + {}^1\vec{V}_2\,, \quad {}^1_2R = \begin{bmatrix} \cos(\Delta\theta) & -sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix}, \quad {}^1\vec{V}_2 = \begin{bmatrix} t_z \\ t_x \end{bmatrix}. \quad (4.3) $$

où $\Delta\theta$ est la différence d'orientation entre $\{P_1\}$ et $\{P_2\}$. La distance au carré entre P_1 et P_2 est alors

$$ \begin{aligned} \text{dist}(z_1, z_2) &= \left\| (z_1, x_1)^T - {}^1_2T \cdot (z_2, x_2)^T \right\|^2 \\ &= (z_1 - \cos(\Delta\theta)z_2 + \sin(\Delta\theta)x_2(z_2) - t_z)^2 + \\ &\quad (x_1(z_1) - \sin(\Delta\theta)z_2 - \cos(\Delta\theta)x_2(z_2) - t_x)^2. \end{aligned} \quad (4.4) $$

Et le gradient $\vec{g}(z_1, z_2)$ de la fonction $\text{dist}(z_1, z_2)$ est formé des dérivées de la fonction dans chaque dimension :

$$ \begin{aligned} \mathbf{g_1}(z_1, z_2) &= \frac{d}{dz_1}\text{dist}(z_1, z_2) \\ &= 2\,[\, z_1 - (\cos(\Delta\theta)\, z_2 - \sin(\Delta\theta)\, x_2(z_2) + t_z) + \\ &\quad \dot{x}_1(z_1)\,(\, x_1(z_1) - (\,\sin(\Delta\theta)\, z_2 + \cos(\Delta\theta)\, x_2(z_2) + t_x)\,)\,]\,; \end{aligned} \quad (4.5) $$

$$ \begin{aligned} \mathbf{g_2}(z_1, z_2) &= \frac{d}{dz_2}\text{dist}(z_1, z_2) \\ &= 2\,[\,(\cos(\Delta\theta) - \sin(\Delta\theta)\,\dot{x}_2(z_2)\,) \times \\ &\quad (\cos(\Delta\theta)\, z_2 - \sin(\Delta\theta)\, x_2(z_2) + t_z - z_1) + (\sin(\Delta\theta) + \cos(\Delta\theta)\,\dot{x}_2(z_2)\,) \\ &\quad \times(\sin(\Delta\theta)\, z_2 + \cos(\Delta\theta)\, x_2(z_2) + t_x - x_1(z_1)\,)\,]. \end{aligned} \quad (4.6) $$

4.5.6.3 Méthode de minimisation

Pour minimiser la fonction $\text{dist}(z_1, z_2)$, nous employons la méthode des gradients conjugués en deux dimensions, à l'aide de l'algorithme de *Polak-Ribiere* trouvé dans

Press et al. (1988). À partir d'un point de départ donné dans l'espace de la fonction, l'algorithme cherche un minimum de la fonction en procédant par minimisations le long de directions (droites) successives dans cet espace. Chaque nouvelle direction de recherche est calculée de manière à être *conjuguée* à la précédente, en exploitant la connaissance du gradient de la fonction. La précision du minimum trouvé est contrôlée par un paramètre de tolérance *tol*. À savoir, la minimisation s'arrête lorsque $|f_{i+1} - f_i| < tol \times |f_i|$, où f_i et f_{i+1} sont les valeurs de la fonction à deux itérations (minimisations le long de droites) successives.

4.5.6.4 Contraintes pour la minimisation

Il est nécessaire de *contraindre* la recherche du segment minimal afin que la solution demeure dans la zone acceptable de la jonction.

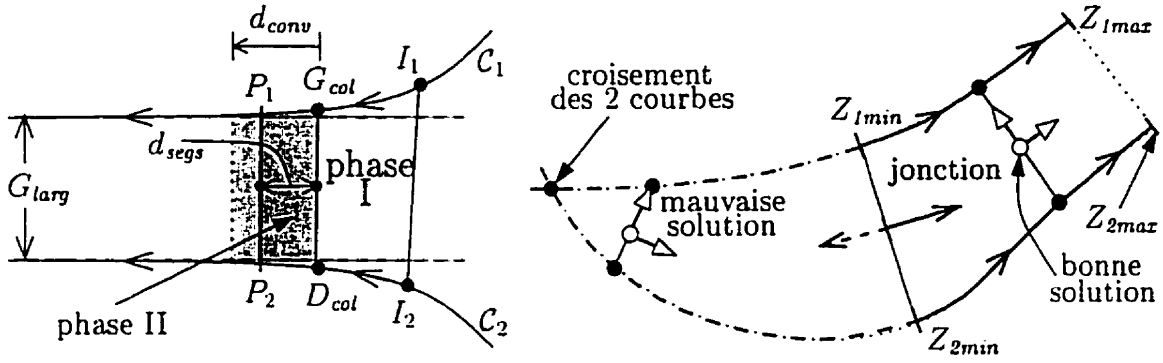
Le premier problème posé par l'étape de minimisation est que la solution peut rentrer trop profondément dans la galerie. Nous avons vu ce même problème dans le cas des parois discrétisées (*cf.* figure 4.9(b)). Ici, nous évitons ce comportement de la solution en faisant trois choses. Premièrement, nous limitons le déplacement dans chaque dimension de la solution, en une itération, à un seuil $lim\Delta z$. En fait, nous normalisons le déplacement calculé dans *Polak-Ribiere* comme suit :

$$\begin{bmatrix} \Delta z_1 \\ \Delta z_2 \end{bmatrix}_{norm} = \frac{lim\Delta z}{\max(|\Delta z_1|, |\Delta z_2|, lim\Delta z)} \begin{bmatrix} \Delta z_1 \\ \Delta z_2 \end{bmatrix}_{P-R}.$$

Deuxièmement, nous forçons en tout temps la solution à rester à l'intérieur d'un *domaine admissible* défini par les abscisses initiales et par les bornes des intervalles $[Z_{min}, Z_{max}]_{1,2}$ correspondant aux extrémités des parois de la jonction vers la galerie.

Troisièmement, la minimisation se déroule en *deux phases* : la phase initiale et la *phase de convergence*. Tant que la distance $\overline{P_1P_2}$ dépasse Glarg, la largeur estimée de la galerie, de plus de 10%, le processus est en phase I; la limite $lim\Delta z$ est relativement élevée (environ 20% de Glarg, ou plus selon l'amplitude du gradient au point

initial). Dès que $\overline{P_1P_2} \leq 1{,}10 \times$ Glarg, la solution rentre dans la zone de convergence et $[G_{col}D_{col}]$ est le segment correspondant (voir figure 4.22(a)). Nous limitons maintenant le déplacement de la solution à $lim\Delta z \simeq 1\%$ de Glarg. Soit d_{segs} la distance entre les milieux du segment courant et de $[G_{col}D_{col}]$. La minimisation s'arrête lorsque $d_{segs} \geq d_{conv}$. Expérimentalement, le seuil de convergence d_{conv} a été fixé à 5% de Glarg.



(a) Deux phases de minimisation.

(b) Solutions avec/sans contrainte de direction.

Figure 4.22 : Contraintes sur la minimisation

Enfin, un autre problème important est que les courbes modélisées ne sont pas forcément convexes l'une vis-à-vis de l'autre et peuvent même se croiser à l'extérieur des domaines en z_1 et z_2 qui définissent la jonction (voir figure 4.22(b)). Sans contraintes, la minimisation pourrait converger vers un segment qui est à l'extérieur de la jonction. Pour éviter cela, nous devons, pour les premières itérations, contraindre les composantes du vecteur $\vec{z}_{12}$ indiquant la direction de minimisation à avoir les mêmes signes que δz_1 et δz_2, les directions initiales de recherche.

4.5.7 Positionnement du référentiel local

À la section 4.4.3, le RL associé à la jonction a été défini comme le repère cartésien situé au milieu du segment minimal et d'orientation donnée par celui-ci. Cependant, l'orientation de la solution $[GD]$ après minimisation n'indique pas forcément bien la perpendiculaire à l'axe de la galerie à son extrémité. Ainsi, nous calculons l'orientation du RL indépendamment de $[GD]$, en nous basant sur la forme des parois au voisinage de $[GD]$.

Plus précisément, nous prenons sur chacune des deux courbes la moyenne des tangentes sur une fenêtre de Glarg/4 vers l'intérieur de la galerie par rapport à $[GD]$. L'orientation Z du RL est alors la moyenne de ces deux quantités. Nous prenons les courbes modélisées plutôt que les points de parois afin d'ignorer la rugosité de celles-ci.

4.6 Détection des référentiels locaux : résultats

4.6.1 Cas expérimentaux

4.6.1.1 Intersections

Nous avons testé le module de détection de RLs expérimentalement dans une série d'intersections de galeries générées grâce à notre banc de simulation en *mine virtuelle* (voir l'annexe A). Pour produire les profils télémétriques et les déplacements odométriques (qui sont les données sensorielles d'entrée de ce module), nous construisons l'environnement, puis nous exécutons une série de trajectoires dans cet environnement.

La figure 4.23 montre les cinq intersections utilisées dans nos essais. Il y a trois intersections de degré 3, qui sont I_1, I_7 et I_{10} ; les deux intersections de degré 4 sont X_2 et X_3. Dans tous les cas, les galeries sont de largeur initiale (avant application de la rugosité) 5 mètres et le paramètre de déviation moyenne fractale des surfaces est de 40 centimètres (*cf.* section A.1 de l'annexe A). La déviation fractale des surfaces

se traduit non seulement en des aspérités locales, mais aussi en des variations plus lentes de la *largeur* des galeries, comme le montrent les figures à la section 4.6.2.

4.6.1.2 Trajectoires

Nous employons 12 chemins topologiques différents dans les intersections montrées dans la figure 4.23 ; ces chemins sont indiqués dans le tableau 4.17. Pour chacun

Tableau 4.17 : Les 12 chemins de traversée d'intersections

	I_1	I_7	I_{10}	X_2	X_3
chemins	$G_1 \rightarrow G_2$ $G_1 \rightarrow G_3$	$G_1 \rightarrow G_2$ $G_3 \rightarrow G_2$	$G_1 \rightarrow G_2$ $G_2 \rightarrow G_3$	$G_1 \rightarrow G_3$ $G_1 \rightarrow G_4$ $G_4 \rightarrow G_2$	$G_1 \rightarrow G_3$ $G_3 \rightarrow G_1$ $G_3 \rightarrow G_2$

de ces chemins, nous testons la stabilité de la détection des RLs dans les jonctions d'entrée et sortie, en faisant varier la déviation sinusoïdale appliquée à la trajectoire idéale (*cf.* section A.2 de l'annexe A). Nous faisons varier l'amplitude, la période et le délai de phase de la déviation, de manière à obtenir 33 cas de test pour chaque chemin. Le tableau 4.18 montre l'amplitude et l'inclinaison maximale (par rapport à la trajectoire idéale) de la sinusoïde décrite par *l'avant* du robot (point M_f), pour les valeurs d'amplitude et de période de déviation appliquées à la trajectoire de *l'arrière* du robot (point M_r).

Tableau 4.18 : Amplitude et inclinaison maximale des trajectoires du point M_f versus l'amplitude et la période des déviations appliquées à M_r

	4 m	6 m	8 m	10 m	période
0 cm	0 cm ; 0°	0 cm ; 0°	0 cm ; 0°	0 cm ; 0°	
10 cm	47 cm ; 8,9°	33 cm ; 6,0°	26 cm ; 4,5°	21 cm ; 3,6°	
25 cm	112 cm ; 21,4°	80 cm ; 14,7°	63 cm ; 11,1°	53 cm ; 8,9°	
amplitude					

Par ailleurs, la distance parcourue entre deux instants de mesure, d_{arp}, est fixée à 25 centimètres.

4.6.1.3 Simulation des signaux de transition

Comme le module détecteur de transitions (de la navigation globale) n'était pas encore développé au moment de faire ces test, il a fallut *simuler* l'envoi des signaux de transition par ce module. Il s'agit de quatre instants différents au cours de la traversée d'intersection, signalant le début (notification) et la fin (confirmation) des jonctions d'entrée et sortie.

Nous supposons que le télémètre frontal (exploité pour les signaux de notification) a une portée maximale de l'ordre de 20 mètres. La notification d'entrée doit correspondre à une position JE_d du robot où celui-ci commence à observer la structure de l'intersection. Pour nos tests, nous fixons JE_d à 17 mètres avant la frontière entre la galerie d'entrée et l'intersection[8] (sur la trajectoire de M_f). Nous fixons la position JE_f, correspondant à la confirmation d'entrée, à 5 mètres après cette même frontière. Ainsi, dans tous les cas de test, la jonction d'entrée correspond à une longueur de 22 mètres parcourue sur la trajectoire de M_f.

Le procédé est le même pour la jonction de sortie. Nous fixons JS_d à 5 mètres avant la frontière entre l'intersection et la galerie de sortie et JS_f à 8 mètres après cette frontière. La jonction de sortie correspond donc à une longueur de 13 mètres parcourue sur la trajectoire de M_f.

4.6.2 Résultats et analyse

Les figures 4.24, 4.25 et 4.26 contiennent les résultats de la détections de RLs pour les 12 chemins. Les graphiques montrent les mesures de parois (prises suivant la trajectoire non-déviée) et les RLs détectés dans les jonctions d'entrée et sortie pour

[8]Les frontières entre galeries et intersections dans les mines virtuelles d'essai sont indiquées en tirets dans la figure 4.23.

les 33 cas de test des 12 traversées d'intersections. Notons que l'orientation de l'axe Z d'un RL est connu à une rotation de π près ; ainsi, les RLs peuvent pointer soit vers l'intérieur des galeries, soit dans le sens inverse, selon les cas.

Les tableaux en dessous de chaque graphique résument la stabilité géométrique des RLs par rapport aux variations de trajectoire. Les quantités Δpos et $\Delta\theta$ sont les *écarts maximaux* en position et en orientation de la distribution des 33 RLs détectés. Δpos équivaut à la diagonale du rectangle de dimensions données par les coordonnées extrémales de la distribution (voir figure 4.27(a)). Les quantités σpos et $\sigma\theta$ sont les *écarts-types* en position et en orientation de la distribution des 33 RLs. Nous calculons σpos par la formule :

$$\sigma pos = \sqrt{\sigma_Z^2 + \sigma_X^2}.$$

C'est donc le rayon d'un cercle placé au centroïde des RLs, représentant la distribution de leurs positions à la fois en Z_W et X_W. Les figures 4.27(b) à 4.27(d) montrent de manière graphique les statistiques de position des RLs pour trois des chemins : le meilleur cas ($I_1 : G_1 \rightarrow G_3$, jonction d'entrée), un cas intermédiaire ($I_{10} : G_2 \rightarrow G_3$, jonction d'entrée) et le pire cas ($X_3 : G_3 \rightarrow G_2$, jonction de sortie). La moyenne des Δpos pour les 24 jonctions (12×2 pour entrée et sortie) est de 81.2 centimètres ; le pire cas est $\Delta pos = 1,55$ mètres. La moyenne des $\Delta\theta$ pour les 24 jonctions est de 4.0° et le pire cas est $\Delta\theta = 9,13°$.

4.6.3 Conclusion

Dans tous les cas expérimentaux, le RL détecté est situé dans la zone acceptable à l'intérieur de sa jonction, là où la largeur mesurée passe d'un état instable (dans l'intersection) à un état relativement stable (dans la galerie). L'*orientation* du RL indique bien celle de l'axe de la galerie à son extrémité (avec une précision d'en moyenne $\pm 2°$, d'après la valeur moyenne de $\Delta\theta$ ci-dessus). Nous pouvons donc affirmer que le premier critère important pour la détection des RLs, celui de la position correcte, est

satisfait.

De plus, la robustesse de la *position* du RL par rapport à la trajectoire suivie est relativement bonne. La position du RL est connue avec une précision d'en moyenne $\pm$ 40 centimètres, si nous approximons le rectangle de diagonale Δpos à un cercle de rayon $\Delta pos/2$. Même dans le pire des 12 cas testés, Δpos représente 30% de la largeur des galeries ($\approx$ 5 mètres), ce qui est raisonnable. (Par exemple, une erreur sur la position du RL de l'ordre de 1,5 mètres entre deux passages par une intersection, les dimensions de la carte métrique étant typiquement de 10 à 20 mètres, n'est probablement pas au delà des capacités d'une approche de relocalisation métrique.) Ainsi, notre implantation du détecteur de RLs satisfait aussi le deuxième critère important, celui de la stabilité géométrique.

Il est sûr qu'en gardant l'approche générale que nous avons adopté pour ce module, il serait possible d'améliorer les résultats actuels en modifiant certaines étapes. Par exemple, les valeurs expérimentales des paramètres influençant l'estimation de largeur de galerie (seuil pour les mesures trop éloignées, longueur prise sur les parois pour estimer Glarg) et les contraintes sur la minimisation ($lim\Delta z$, seuil pour passer des phases I à II, longueur d_{conv} de la zone de convergence) ne sont pas forcément optimales. Cependant, nous rencontrerions ici la difficulté d'établir des nouveaux paramètres qui améliorent la stabilité des RLs dans *tous* les cas expérimentaux et ne dégradent la performance dans aucun.

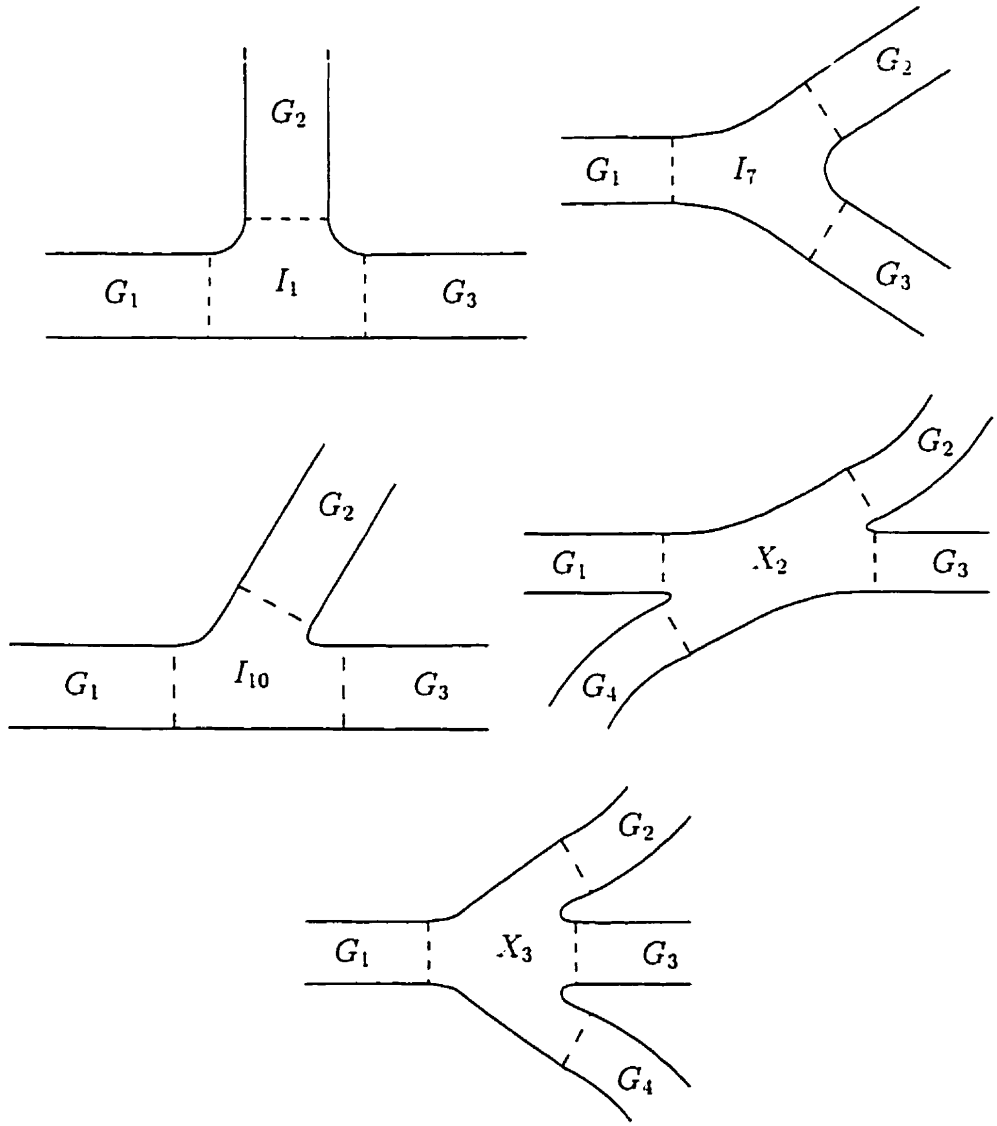
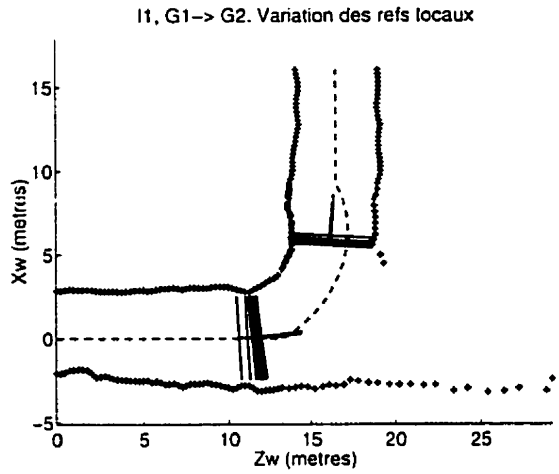
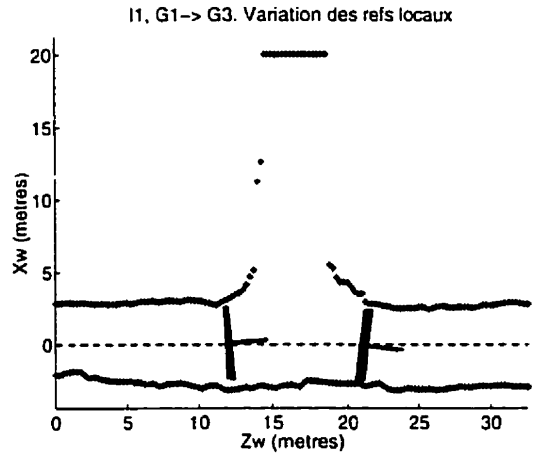


Figure 4.23 : Les cinq intersections d'essai



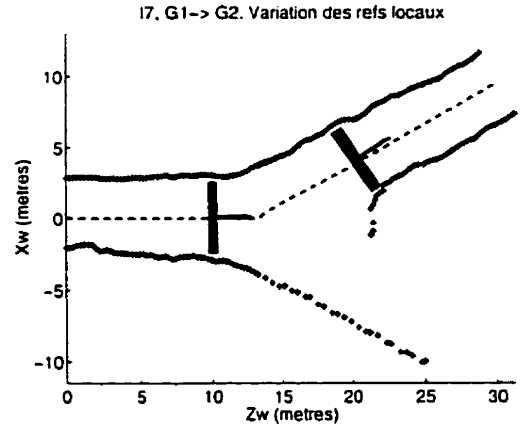
(a) chemin $I_1 : G_1 \rightarrow G_2$.



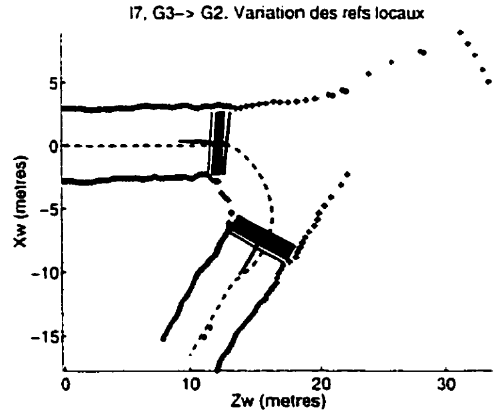
(b) chemin $I_1 : G_1 \rightarrow G_3$.

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	1,39 m	4,25°	7,45 cm	0,97°
sortie	0,57 m	3,80°	1,16 cm	0,77°

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,39 m	3,43°	1,07 cm	0,72°
sortie	0,56 m	2,18°	1,71 cm	0,46°



(c) chemin $I_7 : G_1 \rightarrow G_2$.

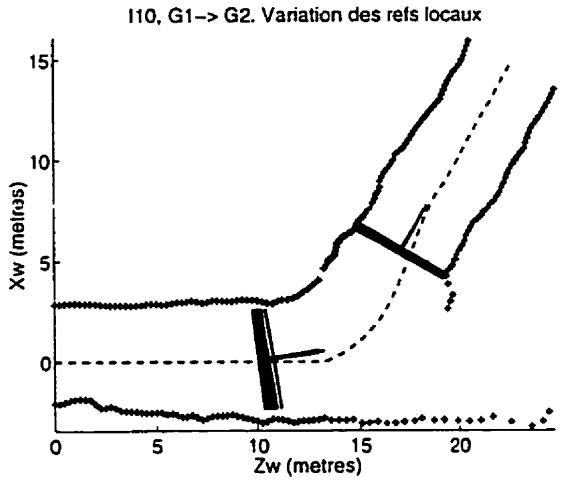


(d) chemin $I_7 : G_3 \rightarrow G_2$.

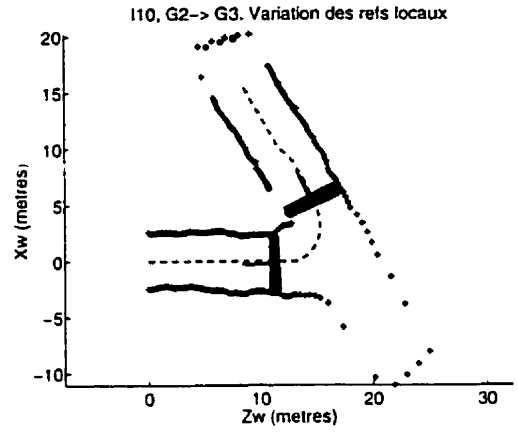
	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,51 m	2,55°	1,26 cm	0,47°
sortie	0,70 m	2,77°	2,05 cm	0,66°

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	1,31 m	2,99°	5,66 cm	0,59°
sortie	1,49 m	2,20°	8,93 cm	0,54°

Figure 4.24 : Résultats pour les 12 traversées d'intersections



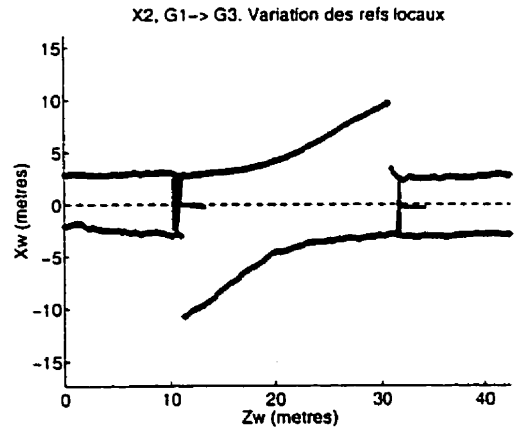
(a) chemin $I_{10} : G_1 \rightarrow G_2$.



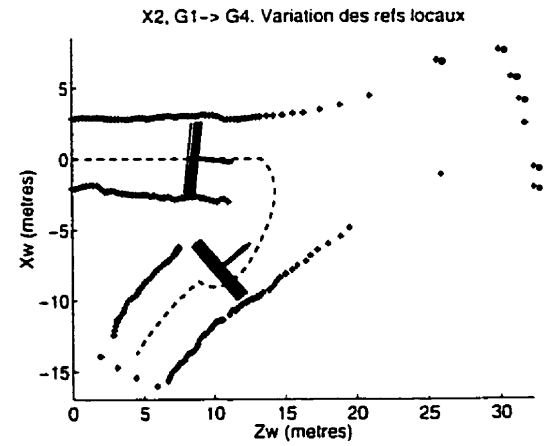
(b) chemin $I_{10} : G_2 \rightarrow G_3$.

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,77 m	2,35°	3,06 cm	0,58°
sortie	0,36 m	1,84°	0,63 cm	0,48°

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,81 m	4,49°	3,34 cm	0,89°
sortie	1,06 m	5,24°	4,52 cm	1,55°



(c) chemin $X_2 : G_1 \rightarrow G_3$.

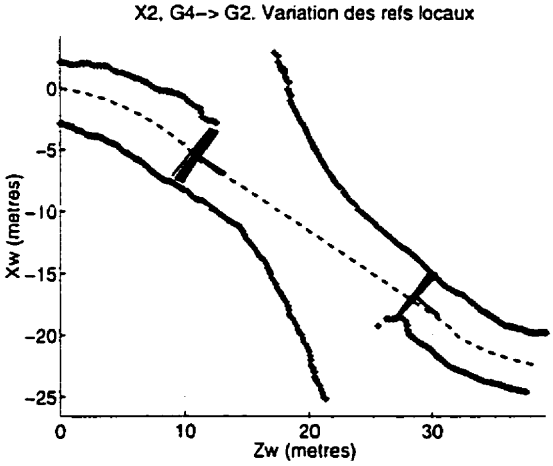


(d) chemin $X_2 : G_1 \rightarrow G_4$.

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,67 m	6,00°	2,17 cm	1,23°
sortie	0,25 m	3,64°	0,16 cm	0,76°

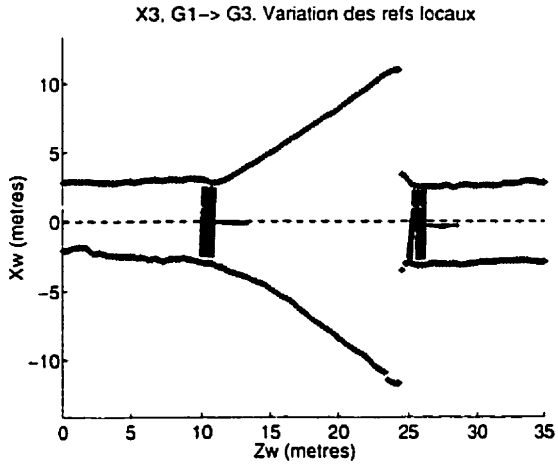
	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,74 m	1,69°	2,93 cm	0,46°
sortie	0,79 m	2,94°	2,33 cm	0,61°

Figure 4.25 : Résultats pour les 12 traversées d'intersections (suite)



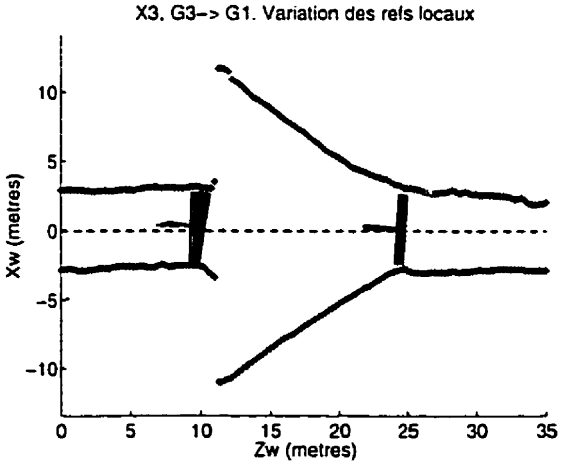
(a) chemin X_2 : $G_4 \rightarrow G_2$.

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,74 m	6,41°	2,37 cm	1,47°
sortie	0,38 m	5,95°	0,56 cm	1,26°



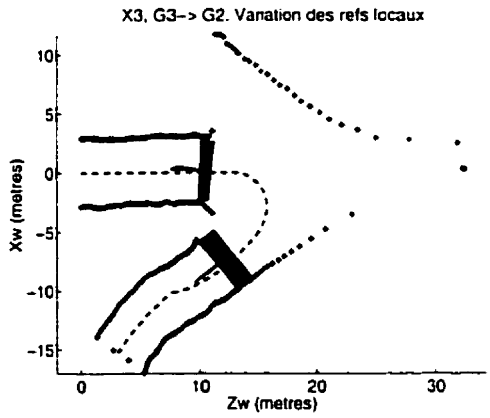
(b) chemin X_3 : $G_1 \rightarrow G_3$.

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,96 m	2,25°	4,67 cm	0,54°
sortie	0,99 m	7,52°	12,48 cm	2,41°



(c) chemin X_3 : $G_3 \rightarrow G_1$.

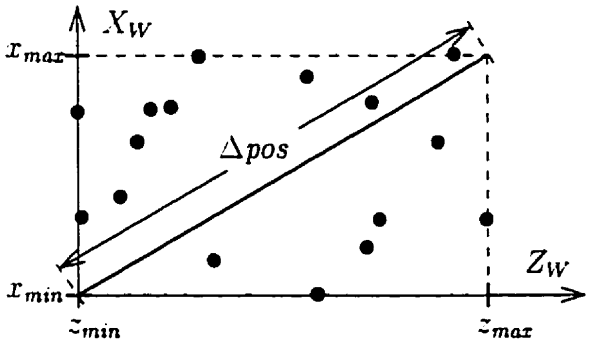
	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,96 m	8,84°	5,88 cm	2,73°
sortie	0,64 m	2,36°	1,66 cm	0,44°



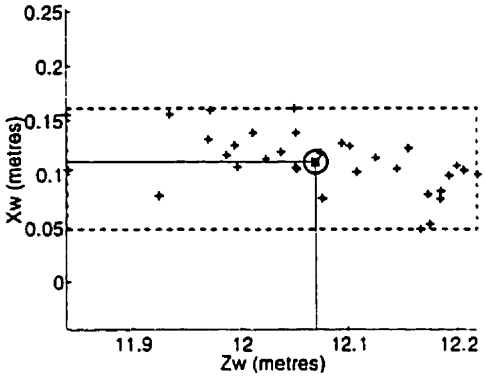
(d) chemin X_3 : $G_3 \rightarrow G_2$.

	Δpos	$\Delta\theta$	σpos	$\sigma\theta$
entree	0,86 m	9,13°	3,57 cm	2,36°
sortie	1,55 m	1,18°	16,18 cm	0,31°

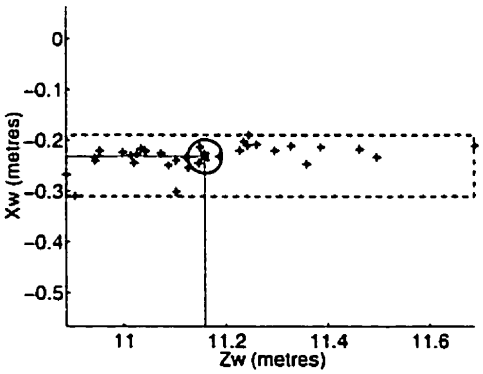
Figure 4.26 : Résultats pour les 12 traversées d'intersections (fin)



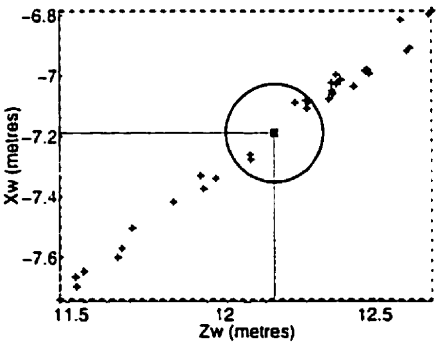
(a) Calcul de Δpos.



(b) Cas $I_1 : G_1 \rightarrow G_3$, entrée.



(c) Cas $I_{10} : G_2 \rightarrow G_3$, entrée.



(d) Cas $X_3 : G_3 \rightarrow G_2$, sortie.

Figure 4.27 : Statistiques de la position des RLs détectés

Chapitre 5

Activité de navigation globale

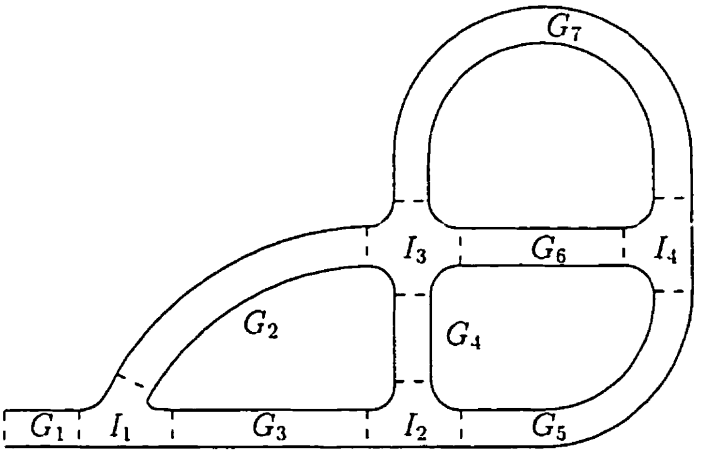
5.1 Introduction

Dans ce chapitre, nous présentons en un premier temps les aspects de localisation topologique dans l'activité de navigation globale : le graphe topologique et la conception du navigateur topologique. En un deuxième temps, nous traitons du module détecteur de transitions : sa conception et l'implantation de ses fonctions. En un troisième temps, nous présentons des résultats expérimentaux pour la détection des transitions.

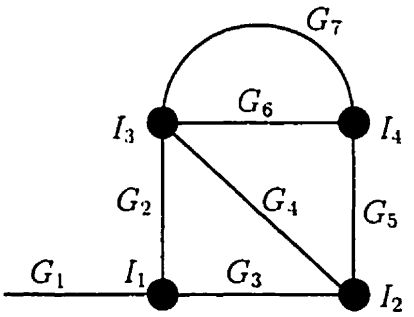
5.2 Le graphe topologique *a priori*

5.2.1 Définition

Nous avons déjà défini le graphe topologique *a priori* au début du chapitre 3 : il s'agit d'une représentation de la *topologie* de la mine (ou de la région de la mine) à explorer par le robot. Dans ce graphe, les arcs représentent les galeries de la mine et les nœuds représentent les intersections de galeries. Les figures 5.1(a) et 5.1(b) montrent un exemple de mine avec son graphe topologique équivalent.



(a) Exemple de mine.



(b) Graphe équivalent.

Figure 5.1 : Représentation topologique d'une mine

Le graphe topologique joue un rôle essentiel dans la navigation du robot, et ce à deux niveaux. Premièrement, le chemin à suivre dans la mine, appelé *chemin global*, est spécifié en termes des éléments topologiques du graphe, c'est-à-dire qu'il consiste en une succession (liste) d'éléments adjacents dans le graphe. Comme nous le verrons à la section 5.3, le module navigateur topologique combine les structures du graphe topologique et du chemin global afin de guider l'activité de navigation locale. À un deuxième niveau, la détection des transitions entre éléments topologiques à mesure que le robot se déplace (section 5.4) lui permet de se localiser globalement dans la

mine.

5.2.2 Décomposition topologique d'une mine

L'existence d'un graphe topologique *a priori* de la mine suppose bien sûr que nous l'ayons *construit* et donc que nous ayons établi des règles générales pour décomposer la mine en éléments topologiques. Nous avons proposé une définition de ce qui constitue un élément topologique à la section 3.4.2 du chapitre 3. D'autre part, le modèle GNOMine (*cf.* section 3.2.1) fournit des définitions géométriques des différents types d'éléments topologiques. Cependant, ces deux séries de règles ne précisent pas les différentes formes géométriques que peut avoir une intersection. Mais la question de ce qui peut constituer une intersection dans la mine a clairement des répercussions importantes pour la construction du graphe topologique et donc pour l'activité de navigation globale.

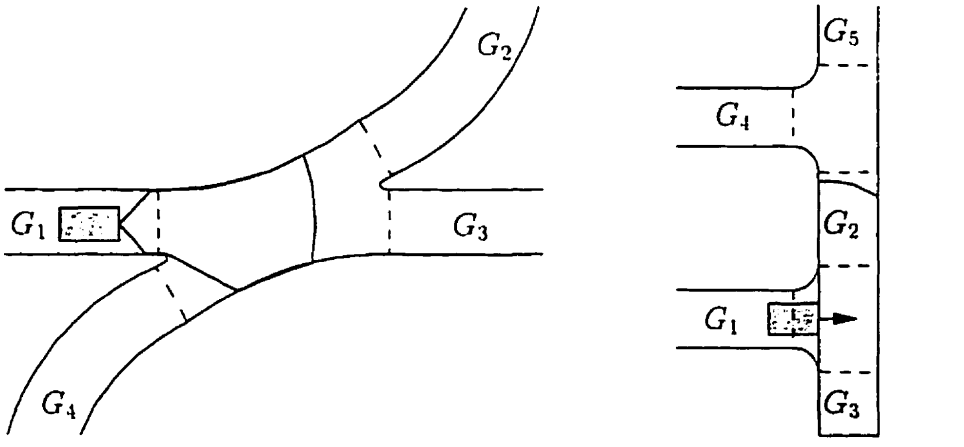
Dans le cas général, le robot n'aperçoit pas forcément toutes les galeries de sortie dès qu'il pénètre dans une intersection. Ceci dépend de la géométrie de l'intersection (dimensions et complexité) et des paramètres α_{front} et d_{max} du senseur frontal. Cette possibilité d'une information sensorielle incomplète pose un problème pour notre stratégie de navigation globale et pourra même causer que le robot "se perde" dans la mine en effectuant un mauvais branchement.

Afin de minimiser ce risque, la segmentation topologique de la mine doit prendre en compte les paramètres sensoriels du robot et doit se baser sur certains critères pour la géométrie de ce que le graphe *a priori* représente comme une seule intersection. De manière qualitative, les critères importants sont

- la visibilité mutuelle entre galeries qui débouchent dans l'intersection (une définition de cette notion est proposée dans Béranger (1996)) ;
- l'existence de trajectoires simples amenant le robot entre les galeries débouchant dans l'intersection ;

- une borne supérieure pour les dimensions d'une intersection.

Nous proposons deux exemples pour illustrer l'application de ces critères à la construction du graphe topologique. Dans la situation de la figure 5.2(a), le robot voit une occlusion correspondant à la galerie G_4, mais confond G_2 et G_3 comme une seule ouverture. Cependant, en avançant droit devant il pourra bientôt distinguer ces deux galeries. Ainsi, cette intersection est modélisée par un seul nœud de degré 4. Par contre, dans la figure 5.2(b), le robot ne verra jamais la galerie G_4 sans tourner à gauche, même avec $\alpha_{front} = \pi$ et D_{max} élevé. Cette intersection-ci correspond donc non pas à un seul nœud de degré 4 mais plutôt à deux nœuds de degré 3 : (G_1, G_2, G_3) et (G_2, G_4, G_5).



(a) Un nœud de degré 4.

(b) Deux nœuds de degré 3.

Figure 5.2 : Modélisation topologique des intersections

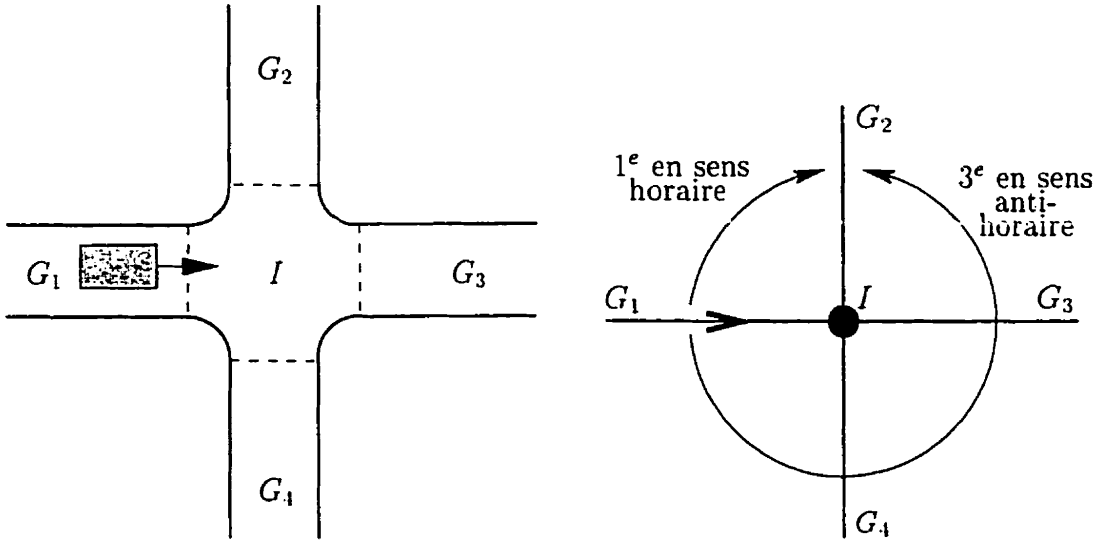
5.3 Conception du navigateur topologique

5.3.1 Considérations pour ce module

Rappelons que le rôle de ce module-ci est d'exploiter le graphe topologique, le chemin global et les signaux de transition afin de déterminer la position du robot dans le graphe de la mine. Si nous supposons que, d'une part, le graphe *a priori* ne contient pas d'erreurs de modélisation topologique et que, d'autre part, le robot identifie toujours correctement son environnement, alors le fonctionnement du navigateur topologique devient très simple. Il doit uniquement mettre à jour un pointeur vers la nouvelle position du robot dans le graphe chaque fois qu'un signal de confirmation de transition est reçu. De plus, ce module rend disponible aux autres modules les étiquettes topologiques des éléments courant, précédent et suivant dans le parcours ainsi que celles des galeries rattachées à une intersection. Dans l'exemple de la figure 5.3, le robot s'approche d'une intersection à quatre branches et le chemin global indique que le branchement doit s'effectuer vers la galerie G_2. L'ordonnancement (circulaire) des arcs connectés au nœud en question correspond à l'ordonnancement des galeries. La bonne branche à suivre est donc indiquée au contrôleur en branchement selon cet ordonnancement, *i.e.* "première en sens horaire" ou bien "troisième en sens anti-horaire" (figure 5.3(b)).

Cependant, si nous considérons de façon réaliste les problèmes potentiels qu'un système autonome peut rencontrer, la fonctionnalité de ce module devient plus complexe que ce que nous venons de voir. De fait, parmi toutes les situations qui pourraient causer que le robot se trompe sur sa position globale, nous énumérons les suivantes :

Erreurs dans le graphe topologique : l'information provenant du graphe peut induire en erreur le robot sur la structure de l'environnement (en particulier le nombre et le degré des intersections), soit à cause d'une connaissance *a priori* incomplète ou incertaine de la mine, soit à cause d'une modélisation inadéquate



(a) Environnement réel.

(b) Modèle topologique.

Figure 5.3 : Rôle du navigateur topologique

par rapport aux capacités sensorielles du robot (*cf.* section 5.2.2).

Erreurs lors d'un branchement : lorsque le robot débouche dans une intersection, une information sensorielle incorrecte ou incomplète sur les galeries de sortie peut faire que le contrôleur en branchement amène le robot dans la mauvaise galerie. Comme sources possibles de ce type de problème, il y a la géométrie de l'intersection, le bruit télémétrique et le traitement des profils frontaux, qui tous trois peuvent provoquer soit la détection de "fausses branches", soit au contraire la non-détection de vraies branches de sortie.

Erreurs dans la détection des transitions : le détecteur de transitions (section 5.4) pourrait être induit en erreur dans certains cas limites. Ce module pourrait "manquer" une intersection, par exemple une intersection de degré trois mais dont une des galeries de sortie demeure invisible au robot. Inversement, l'élargissement d'une galerie ou bien l'apparition d'une fausse branche dans

les profils frontaux pourrait amener ce module à signaler une intersection qui n'existe pas en réalité.

Dans ce contexte, nous devons rajouter au navigateur topologique un rôle supplémentaire, celui de surveiller les observations du milieu et la détection des transitions afin de vérifier que ceux-ci concordent avec ce que prédit le graphe topologique. Ce module doit alors détecter tout cas de divergence et choisir une stratégie pour corriger la situation. S'il est possible de déduire immédiatement et avec certitude la cause de l'erreur (parmi les situations ci-dessus), le robot pourra corriger soit le graphe, soit sa position globale, soit les deux, et pourra peut-être poursuivre le parcours indiqué par le chemin global sans modification. Mais dans la plupart des situations, il sera nécessaire d'interrompre le parcours initial et de planifier de nouveaux déplacements afin d'identifier la cause de la divergence et de retrouver un lieu connu sur le chemin global.

La replanification de chemin en situation d'incertitude, l'identification et la correction de fausses hypothèses sur le milieu et la correction d'erreurs dans le graphe topologique sont des problèmes relativement complexes. Il s'agit en effet de sujets de recherche abordés dans les travaux à la section 2.1.4 de la revue bibliographique. Dudek et al. (1996) analysent l'exploration de graphes *a priori* inconnus avec comme unique donnée sensorielle le degré des nœuds. Leur exploration emploie alternativement la recherche par niveaux et la recherche en profondeur et on développe un raisonnement sur les identités, connues ou inconnues, des nœuds et arcs rencontrés. Dans Dudek et al. (1993), on propose une approche pour vérifier que le graphe *a priori* d'un milieu est correct ; ici, le robot dispose d'un ou plusieurs marqueurs pour aider à distinguer entre eux les nœuds du milieu.

Kuipers et Byun (1987) introduisent un algorithme appelé *rehearsal procedure* pour permettre à un robot de distinguer un lieu rencontré d'un lieu déjà répertorié dans sa carte du milieu (initialement inconnu) ayant la même "signature sensorielle". Le robot doit explorer un voisinage du lieu en question et voir si ces autres lieux

correspondent oui ou non aux connaissances dans la carte. Engelson et McDermott (1992) proposent des méthodes de correction d'erreurs qui peuvent s'introduire dans la carte topologique d'un milieu. L'analyse des observations externes ainsi que les mesures de déplacement (odométrie) peuvent amener le robot à corriger soit la position géométrique d'un lieu dans sa carte, soit sa propre position métrique et topologique actuelle. De même, le robot pourra décider de fusionner deux nœuds voisins dans sa carte ou inversement de scinder en deux un nœud de la carte qui aurait englobé deux lieux distincts.

Dans la décomposition fonctionnelle qui suit, nous retenons néanmoins les deux suppositions simplificatrices déjà citées, à savoir que, d'une part, le graphe topologique de la mine est correct et que, d'autre part, le système identifie toujours correctement la topologie de l'environnement. Notons qu'une façon de satisfaire la deuxième supposition serait qu'un conducteur humain remplace les modules de contrôle local du véhicule[1] et supervise le processus de détection des transitions. Le conducteur s'assurerait lui-même d'effectuer les bons branchements et corrigerait des erreurs commises par le détecteur de transition quand et si elles se produisent.

5.3.2 Décomposition du module en fonctions

Le schéma d'interaction des différentes fonctions du module navigateur topologique se trouve à la figure 5.4. Les paragraphes qui suivent décrivent les fonctions (les boîtes dans l'organigramme) en termes d'entrées, sorties et rôle.

5.3.2.1 Interface au chemin global

Cette fonction sert d'interface à la structure de données du chemin global. Elle maintient un pointeur vers l'élément topologique actuel dans le parcours et incrémente ce pointeur chaque fois qu'elle reçoit une confirmation de transition du détecteur de

[1] possibilité déjà soulevée à la section 3.4.2 ;

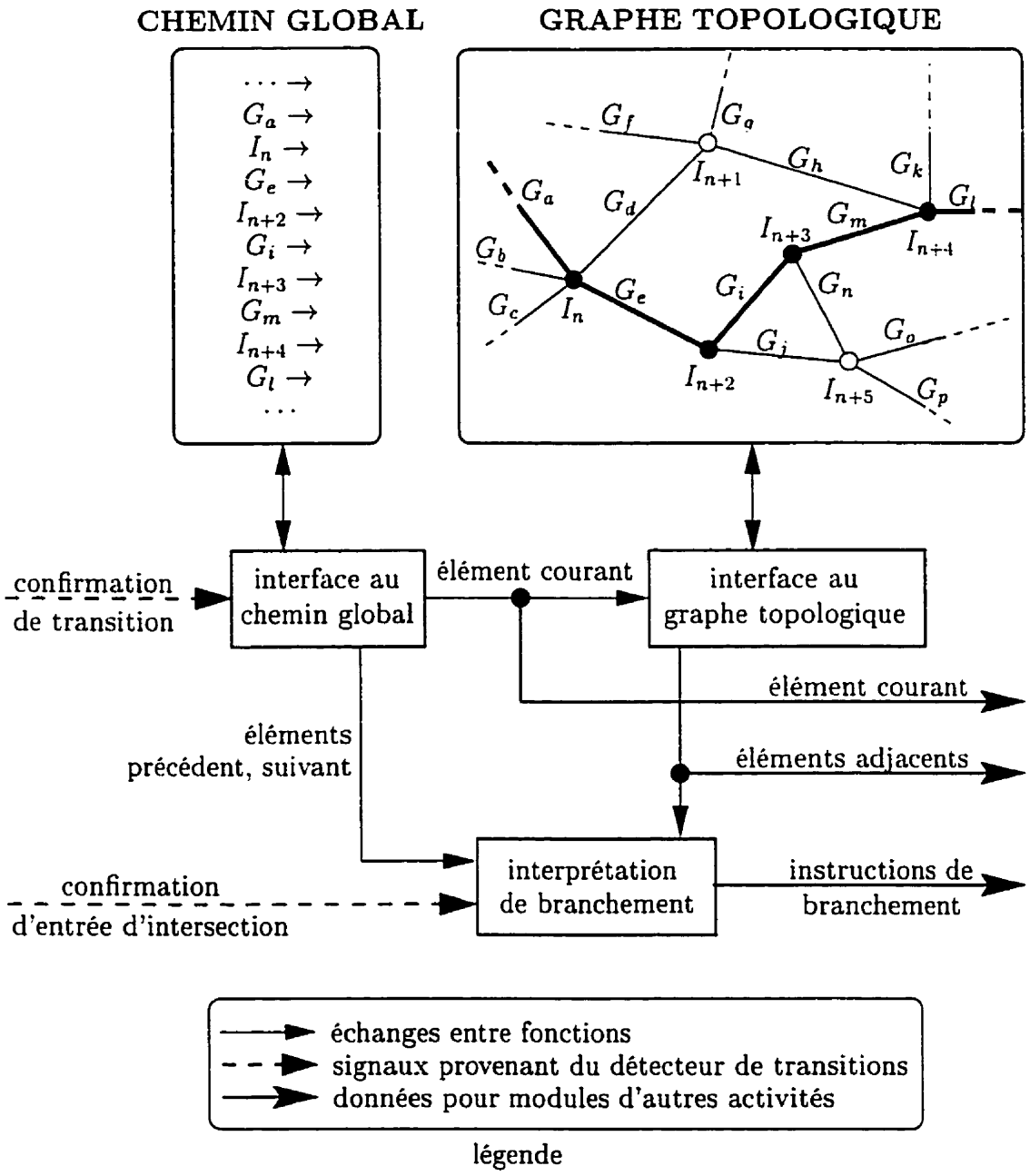


Figure 5.4 : Navigateur topologique : échanges de données et signaux

transitions. Cette fonction retourne l'étiquette topologique du nouvel élément courant et celles des éléments précédent et suivant dans le parcours (*cf.* tableau 5.1).

Tableau 5.1 : Fonction d'interface au chemin global

entrée	- structure de données du chemin global ; - confirmations de transitions (entrée ou sortie d'intersection).
sortie	- élément topologique courant ; - éléments précédent et suivant dans le parcours.

5.3.2.2 Interface au graphe topologique

Cette fonction sert d'interface à la structure de données du graphe topologique. Elle maintient un pointeur vers l'élément courant dans le graphe et retourne les étiquettes de tous les éléments (nœuds ou arcs) adjacents à l'élément courant (arc ou nœud), dans leur ordonnancement circulaire (*cf.* tableau 5.2).

Tableau 5.2 : Fonction d'interface au graphe topologique

entrée	- structure de données du graphe topologique ; - élément topologique courant.
sortie	- éléments adjacents à l'élément courant, en ordre circulaire.

5.3.2.3 Interprétation de branchement

Lorsque le robot rentre dans une intersection, cette fonction combine l'information des étiquettes des arcs précédent et suivant dans le parcours et de tous les arcs (en ordre circulaire) adjacents au nœud courant, pour formuler deux instructions de branchement correspondant aux sens horaire et anti-horaire, de la forme "i^e galerie à partir de celle d'arrivée en sens [horaire/anti-horaire]" (*cf.* tableau 5.3).

Tableau 5.3 : Fonction d'interprétation de branchement

entrée	- éléments précédent et suivant dans le parcours ; - éléments adjacents à l'élément courant, en ordre circulaire ; - confirmations d'entrée d'intersection.
sortie	- deux instructions de branchement (sens horaire et anti-horaire).

5.3.3 Limite du développement

Comme nous l'avons déjà noté, la fonctionnalité du navigateur topologique est très simple, voire même triviale, en vue des suppositions simplificatrices posées à la section 5.3.1. De ce fait, il ne serait pas très intéressant d'un point de vue de recherche d'implanter ce module. Nous le laissons donc à l'état conceptuel au profit du développement du module détecteur de transitions.

5.4 Conception du détecteur de transitions

5.4.1 Conventions typographiques dans la notation

Comme dans le chapitre précédent, nous appliquons dans la suite certaines conventions de typographie afin de distinguer dans la notation les différents types de quantités auxquelles nous nous référons. Lorsqu'il s'agit d'une quantité relative à la géométrie de l'environnement, des profils télémétriques ou de la trajectoire du robot (point, distance, angle), ou d'une quantité relative à la dimension du temps, le paramètre en question est noté en italiques ; des exemples figurant dans la suite sont D_{occ}, $\delta\varphi$, t_I. Plusieurs paramètres représentent des fenêtres d'analyse exprimées en nombre d'instants de mesure ou en nombre d'itérations (dans un algorithme) et sont notés en caractères obliques ; des exemples sont *fenl*, *nbInsl*, *minItersStab*. Les variables figurant seulement au sein de la description d'algorithmes sont notées en caractères dactylographiques ; des exemples sont `limvis`, `ensRLs`. Les symboles

dénotant des caractéristiques importantes résultant d'un traitement utilisent les petites capitales ; des exemples sont LV, NR. Enfin, les abréviations désignant les signaux de transition sont notés en caractères obliques ; un exemple est *NotifEn*.

Si le nom d'un paramètre comporte un indice, la typographie de celle-ci sera normalement identique au reste du nom, sauf si l'indice se réfère à un autre type de quantité. Le tableau 5.4 résume les conventions typographiques employées dans la notation des paramètres, variables et symboles dans le reste de ce chapitre.

Tableau 5.4 : Conventions typographiques dans la notation

type de quantité	typographie
quantité géométrique ou temporelle	$Quantite_{ind}$
fenêtre d'analyse en nombre d'instants	*fenetre*
variable au sein d'un algorithme	`variable`
symbole d'une caractéristique importante	SYMBOLE
abréviation désignant un signal de transition	*signalTrans*

5.4.2 Exploiter les senseurs frontal et latéral

Tandis que le module précédent raisonne sur l'environnement au niveau d'abstraction topologique, le détecteur de transitions, lui, doit jouer un rôle *perceptuel* dans le positionnement global du robot. Comme nous l'avons vu au chapitre 3, ce module-ci est responsable d'envoyer des *signaux* au reste du système chaque fois qu'il se produit une *transition de structure*, ou autrement dit que le robot change d'élément topologique.

La configuration sensorielle du robot procure deux sources d'information pour cette tâche : le télémètre frontal et les télémètres latéraux. La discussion sur les besoins de la navigation globale (section 3.3.3) a déjà démontré la nécessité d'exploiter le senseur frontal afin de résoudre certaines situations ambiguës que les senseurs latéraux ne peuvent régler (rappel des figures 3.5 et 3.6). Comme le senseur frontal voit *en*

avance sur le véhicule, l'analyse perceptuelle qui lui est rattachée détecte forcément une structure particulière avant que le robot ne l'atteigne. Le signal envoyé sert donc d'alerte aux autres modules de l'approche d'une transition (arrivée d'une intersection ou d'une nouvelle galerie) ; nous appelons ce signal la **notification de transition**.

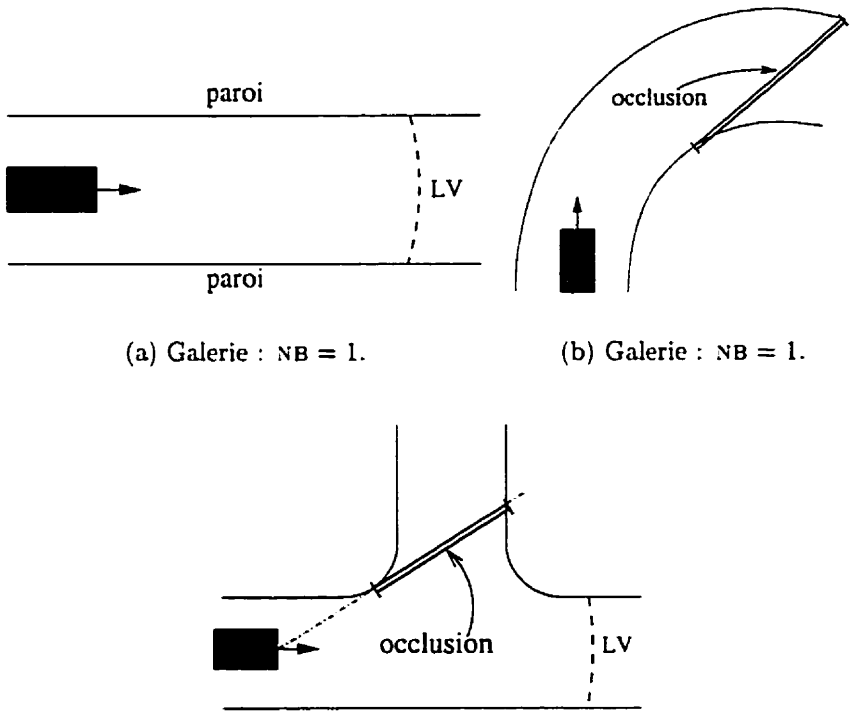
Parallèlement, les senseurs latéraux observent l'environnement *autour* du véhicule (plutôt que devant lui). Nous pouvons nous attendre à ce que la même transition de structure qui a déclenché un signal de notification se manifeste aussi dans les mesures latérales un peu plus loin dans le parcours. Il est avantageux d'exploiter cette deuxième source de renseignements perceptuels pour deux raisons. D'une part, elle nous permet de confirmer l'existence de la structure physique détectée auparavant dans les mesures frontales. D'autre part, l'analyse des mesures latérales permet de localiser plus précisément dans la mine l'endroit de la transition de structure. En effet, alors que l'analyse frontale correspond à observer une structure à une certaine distance (proportionnelle à la portée maximale d_{max}), l'analyse latérale correspond au *passage* du robot par cet endroit. Cette dernière analyse donne lieu à un deuxième signal que nous appelons la **confirmation de transition**.

Ainsi, nous aboutissons à deux signaux distincts indiquant au système à deux moments différents un changement d'élément topologique. Au cours de la traversée d'une intersection, le robot rentre dans l'intersection puis en ressort (pour entrer dans une nouvelle galerie). Ces deux transitions sont marquées par quatre événements qui sont, dans le temps : la **notification d'entrée** (d'intersection), la **confirmation d'entrée**, la **notification de sortie** (d'intersection) et la **confirmation de sortie**. Dans ce qui suit, nous désignerons souvent les quatre signaux par leurs abréviations respectives : *NotifEn*, *ConfirmEn*, *NotifSo*et *ConfirmSo*.

5.4.3 Exploiter la redondance temporelle

Dans l'optique d'assurer la fiabilité des signaux générés par ce module, il faut tenir compte de trois facteurs qui influencent la qualité des données sensorielles reçues : la

nature très variée de la structure locale des parois, le bruit de capture et de traitement des mesures télémétriques et la trajectoire suivie par le robot. Des exemples précis démontrent aisément cette réalité. Considérons d'abord le contexte de l'analyse des profils frontaux pour le signal *NotifEn*. Il nous faut définir ce qui constitue "voir l'arrivée d'une intersection", c'est-à-dire la *signature sensorielle* qui provoquera le signal. Un choix naturel pour celle-ci est le nombre de branches[2] (dénoté NB) observées dans le champ de vue frontal : le robot observe une branche unique dans une galerie, mais ce nombre augmente à mesure que d'autres galeries rentrent dans son champ de vue à l'approche d'une intersection (figure 5.5).



(a) Galerie : NB = 1.

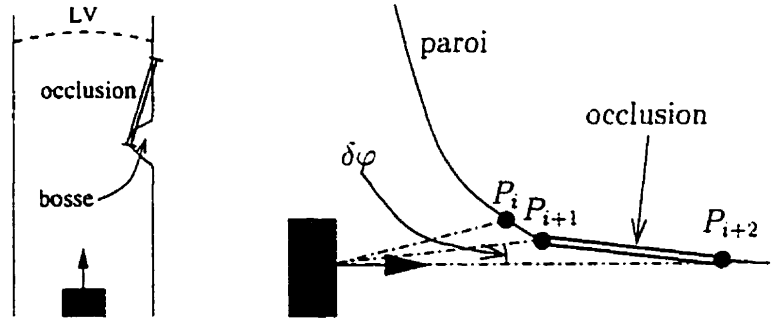
(b) Galerie : NB = 1.

(c) Intersection : NB > 1.

Figure 5.5 : NB comme signature pour la notification d'entrée

[2]Ici, le terme *branche* désigne une occlusion ou une région à la limite de visibilité (ou portée) (dénotée LV) détectée dans le profil frontal.

Il semblerait alors logique d'envoyer *NotifEn* dès que NB passe de 1 à plusieurs. Cependant, cette analyse trop simple est induite en erreur dans bien des cas. La figure 5.6 illustre deux situations où des occlusions apparaissent de façon transitoire dans le champ frontal. Dans l'une (figure 5.6(a)), c'est une aspérité importante qui occulte une partie de la paroi de galerie ; dans l'autre (figure 5.6(b)), c'est une paroi presque tangente à son angle de vue qui crée une fausse occlusion (distance relativement grande entre deux points consécutifs du profil). Dans les deux cas, il suffit d'un léger déplacement du robot pour que l'occlusion disparaisse.

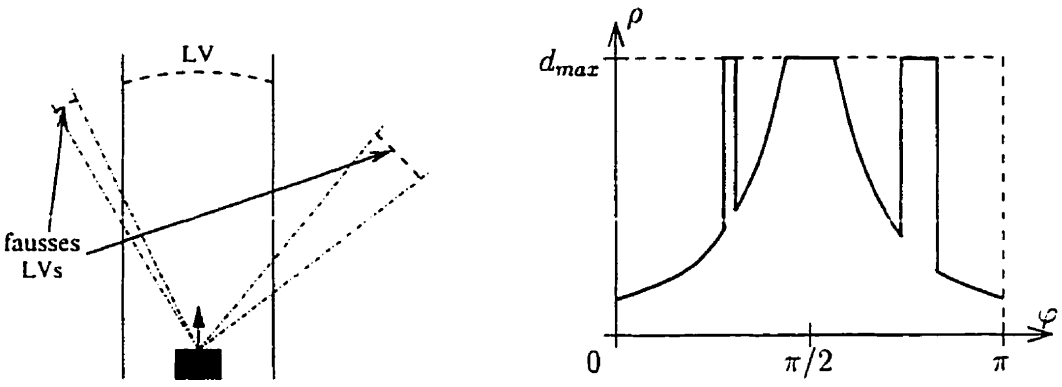


(a) Bosse sur la paroi. (b) Fausse occlusion.

Figure 5.6 : Occlusions transitoires dans le champ de vue frontal

Les régions à la limite de visibilité (dénotées LVs) sont des parties du champ frontal où aucune mesure n'est faite à cause d'un signal laser de retour trop faible. Une LV indique normalement que les parois dans ces directions sont au-delà de la portée d_{max} et donc qu'un espace navigable (une ouverture) s'y trouve. Mais cette deuxième sorte de branche peut aussi être trompeuse, soit lorsque la rugosité ou la nature des parois crée des "trous" dans le profil, soit à cause d'un traitement erroné des mesures (figure 5.7).

Le même genre de problème se présente dans le contexte de l'analyse des mesures latérales (signaux de confirmation). Par exemple, pendant que le véhicule circule dans



(a) Dans l'environnement. (b) Profil en coordonnées polaires.

Figure 5.7 : Fausses LVs dans le champ de vue frontal

une galerie de largeur constante, la largeur mesurée, Dlat peut néanmoins fluctuer en fonction de la trajectoire suivie (figure 5.8). Si l'orientation du véhicule s'éloigne beaucoup de celle de l'axe de la galerie, l'augmentation de la largeur apparente pourrait confondre une stratégie cherchant à reconnaître l'évasement d'une jonction. Des pics sporadiques dûs au bruit dans l'évolution de Dlat pourraient eux aussi déclencher des signaux erronés.

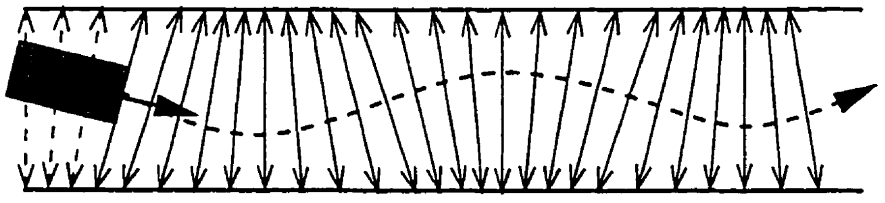


Figure 5.8 : Fluctuations de Dlat en fonction de la trajectoire

Il est évident d'après ces quelques cas de figure que, autant pour l'analyse latérale que frontale, toute décision basée sur un changement *instantané* de l'information sensorielle sera inadéquate et générera des signaux souvent erronés. La reconnaissance des transitions est au contraire un processus *dynamique* qui exploite la *redondance*

temporelle de l'information captée et les signatures sensorielles définies dans la suite utiliseront les mesures accumulées sur une fenêtre de temps plutôt qu'à un seul instant.

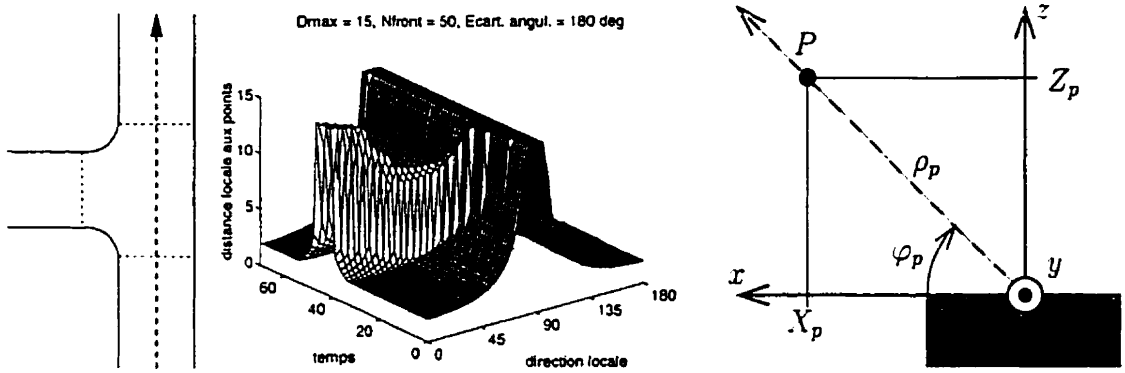
5.4.4 Signal de notification d'entrée d'intersection

Le nombre de branches NB présents dans le champ frontal semble un bon indice de l'approche d'une intersection, mais nous ne pouvons nous fier à un changement instantané de NB, comme nous venons de voir. Il en découle l'idée de *suivre* l'évolution de cette caractéristique au cours du temps et de baser *NotifEn* sur les observations répétées de plusieurs branches que nous pouvons corréler sur une fenêtre de temps. Ceci implique d'abord un algorithme pour extraire les branches d'un seul profil, puis une méthode pour faire le suivi des branches détectées. La première de ces tâches ne demande pas un algorithme très complexe dans la mesure où nous acceptons un certain niveau d'erreurs de segmentation dues au bruit ou à la mauvaise interprétation des mesures. De fait, une telle fonction de détection de branches fait partie de l'implantation de ce module (section 5.4.8).

Par contre, le suivi temporel des branches constitue un problème plus difficile. Sa difficulté est causée par la nature imprévisible et bruitée du signal qu'il doit traiter : changement continu de la position angulaire d'une branche dans le champ de vue, transformation d'une occlusion en LV ou vice-versa, apparition ou disparition de branches liée à la géométrie de l'intersection, fausses branches ou branches manquées de manière aléatoire. Pour être capable d'associer correctement les branches détectées dans le temps, il faudrait introduire une boucle de filtrage prédictif dans l'interprétation du signal. Mais cette approche ne nous satisfait pas ni du point de vue de la fiabilité, ni pour la simplicité d'implantation que nous visons.

À l'opposé de la stratégie ci-dessus, il y a l'idée de laisser accumuler les profils bruts sur une certaine durée avant d'en faire un traitement. Nous qualifions ce genre d'approche de *traitement par lots*. L'exemple de la figure 5.9 montre le résultat d'accumuler les profils frontaux au cours d'une traversée d'intersection. La surface

dans la figure 5.9(b) est formée des profils instantanés représentés en coordonnées polaires dans le référentiel local du véhicule (distance ρ versus direction φ, définies dans 5.9(c)) et qui évoluent dans le temps (dimension t).

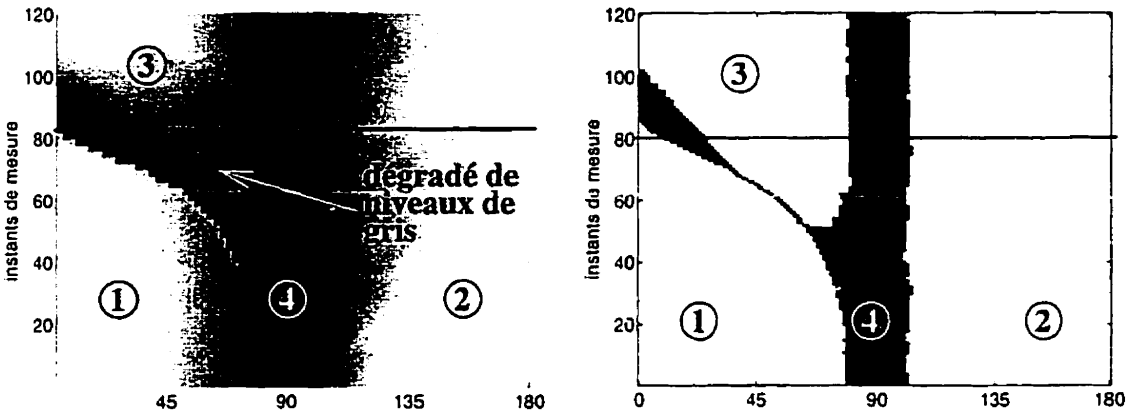


(a) Traversée d'intersection ; (b) Surface dans l'espace (ρ, φ, t) ; (c) Point P : (ρ_p, φ_p) dans le profil frontal.

Figure 5.9 : Évolution des profils frontaux lors d'une traversée d'intersection

L'image dans la figure 5.10(a) représente la même surface que dans 5.9(b), mais cette fois vue de haut, c'est-à-dire projetée dans le plan (φ, t). Le niveau de gris dans l'image est proportionnel à la hauteur de la surface, donc à la distance ρ. La région noire, que nous appelons la *trace* des profils frontaux, montre de manière implicite (sans traitement) l'évolution des LVs et des occlusions éloignées du robot. En appliquant une segmentation à cette image, il est possible d'en isoler la trace ainsi que les régions (blanches) qu'elle sépare ; ces régions sont numérotées dans la figure 5.10(a).

Le nombre de régions blanches (dénoté NR) est un indicateur de la topologie de l'environnement devant le robot. Les régions 1 et 2 sont toujours présentes quand le robot est dans une galerie ; lorsqu'apparaissent de nouvelles régions (ici, la région 3), c'est que le milieu contient des structures autres que simplement deux parois de galeries et donc qu'une intersection prend forme progressivement dans le champ



(a) Image des profils non-traités ; (b) Image des profils segmentés.

Figure 5.10 : Deux représentations différentes pour le traitement par lots

frontal[3]. De plus, la représentation des mesures brutes sous cette forme nous procure en quelque sorte un filtrage temporel implicite. En effet, le bruit sensoriel pourra rajouter des perturbations locales aléatoirement dans l'image formée, mais ne changera pas (sauf cas dégénéré) sa topologie. Une mesure (NR $>$ 2) est donc équivalente à observer (NB $>$ 1) branches pendant le lapse de temps correspondant à la fenêtre en t de l'image traitée. De ce fait, (NR $>$ 2) (pour un nombre d'instants à établir) est une signature sensorielle adéquate pour affirmer que le robot s'approche en effet d'une intersection.

Remarquons qu'un critère de performance important du point de vue de la navigation locale est le délai (ou la distance) *avant* l'intersection que la notification est envoyée[4]. La trace des profils est entourée de dégradés de niveaux de gris ; celle

[3]Une description plus détaillée de la relation entre la topologie et la géométrie de la trace des profils, d'une part et celles de l'intersection, d'autre part, se trouve dans Debanné, Hervé et Cohen (1997).

[4]Le trait horizontal dans les images de la figure 5.10 marque l'instant $t = t_e$ où le véhicule (point M_r) traverse la frontière entre la galerie et l'intersection dans la mine virtuelle.

indiquée par une flèche à la base de la région 3 (donc là où une "trace secondaire" se détache de la "trace principale") retardera la détection de la nouvelle région blanche. Un seuillage de l'image nous vient à l'idée pour éliminer ces dégradés. Mais alors se pose le problème du choix du seuil car, les traces secondaires étant minces et liées à des occlusions de distance ρ variable, ce traitement pourra les sectionner et résulter en une perte d'information importante.

L'approche que nous adoptons finalement consiste à combiner les deux stratégies du traitement profil par profil et du traitement par lots. Nous détectons les branches à chaque instant de mesure, pour ensuite laisser accumuler les profils traités et en former une image (figure 5.10(b)). Cette deuxième image correspond à une binarisation de la première en passant par l'analyse des LVs et occlusions; ainsi, nous la qualifions d'*image binaire*[5] des profils accumulés. Cette méthode nous procure une représentation mieux exploitable pour générer la notification. En effet, l'apparition de traces secondaires (événement complémentaire de l'apparition de nouvelles régions blanches) est maintenant détectable plus tôt sur l'axe des t.

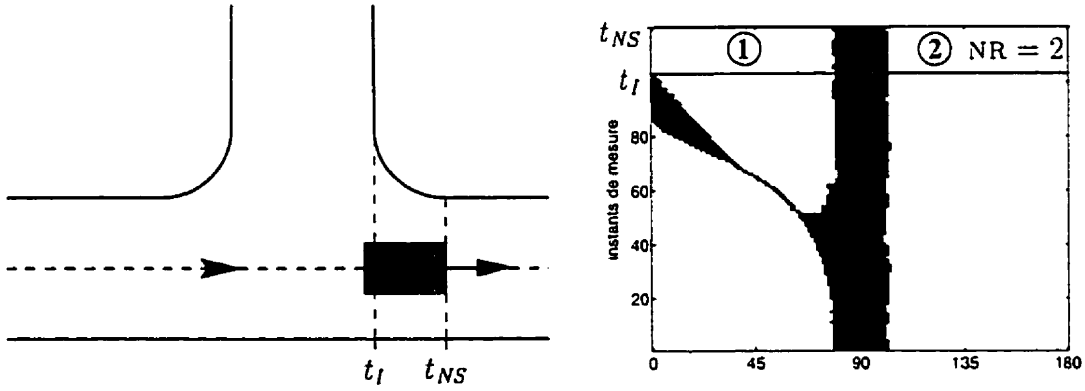
Nous avons mentionné le facteur du bruit de mesure dans les profils frontaux, qui est inévitable dans un système réel. La présence de bruit de capture affecte bien entendu l'approche préconisée : les contraintes ainsi que les résultats du traitement des profils, la qualité des images binaires produites et donc la difficulté de les segmenter. Cependant, dans l'implantation des fonctions de ce module, nous supposons des profils dont le niveau de bruit de capture est négligeable.

5.4.5 Signal de notification de sortie d'intersection

L'arrivée d'une intersection se manifeste perceptuellement par l'apparition de plusieurs branches (NB > 1) dans les profils frontaux, donc d'un nombre de régions (NR > 2) dans l'image des profils accumulés. Inversement, lorsque le robot sort de l'intersection

[5]Dans l'image binaire, les pixels noirs marquent, à chaque instant de mesure, les directions φ où des LVs ou occlusions sont identifiées.

et que les galeries autres que celle de sortie disparaissent du champ frontal, l'observation redevient celle d'une galerie unique (NB = 1, NR = 2). Nous pourrions rattacher *NotifSo* à cet événement. Cependant, cela signifie que le véhicule serait déjà engagé dans la galerie de sortie quand cet état a été observé pendant suffisamment longtemps pour envoyer le signal (l'instant t_{NS} dans l'exemple de la figure 5.11). Ceci n'est pas idéal car le détecteur de référentiels locaux ne retient les mesures latérales qu'à partir de cet instant (*cf.* section 4.4.2) et donc il lui manquerait le début de la structure de la vraie jonction de sortie. Nous voulons plutôt que le détecteur de référentiels locaux prennent des mesures latérales débutant avant la jonction et qu'il isole lui même les parois de jonction. Une notification de sortie dépendante de (NR = 2) viendrait donc *trop tard* dans la traversée d'intersection.



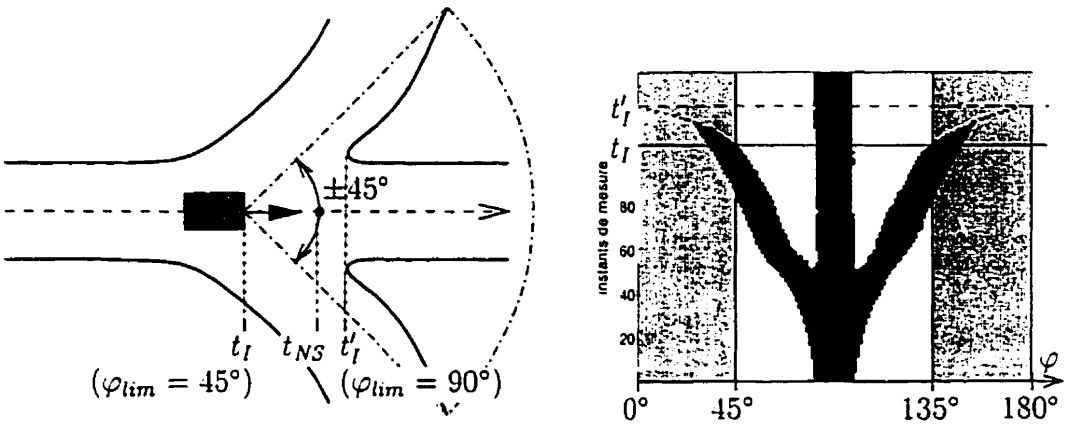
(a) Robot engagé dans la jonction de sortie.

(b) Trace des profils.

Figure 5.11 : Notification de sortie basée sur (NR = 2) vient trop tard

Un moyen d'assurer que *NotifSo* vienne plus tôt tout en gardant la même idée de base serait d'exclure les portions du champ frontal au-delà d'un certain angle. Les figures 5.12(a) et 5.12(b) montrent le champ frontal limité à $|\varphi - 90°| \leq \varphi_{lim}$ pour $\varphi_{lim} = 45°$. Ici, le robot est à une position où la mesure NB revient à 1 ; l'instant t_{NS} est toujours décalé dans le temps (pour laisser NB se stabiliser à 1), mais le signal vient

plus tôt dans la trajectoire que si nous gardons le champ de vue entier. Le problème qui se pose maintenant est celui du choix de φ_{lim} : celui-ci doit être suffisamment inférieur à 90° pour garantir que t_{NS} précédera la jonction de sortie dans tous les cas. Mais pour le φ_{lim} choisi, il risque d'y avoir plusieurs cas où *NotifSo* viendrait *trop tôt* (avant même que le robot ne rentre dans l'intersection).



(a) Dans l'environnement ; (b) Dans l'image binaire des profils.

Figure 5.12 : Limiter le champ frontal en angle pour l'analyse de NB, NR

Il semble donc qu'une signature pour *NotifSo* basée sur l'évolution du nombre de branches observées est problématique pour le reste de notre système. Pour cette raison, nous prenons en fin de compte comme *NotifSo* le même signal que *ConfirmEn*. Celle-ci coïncide avec la fin des mesures latérales de la jonction d'entrée (voir la section suivante) ; à ce moment le robot est rentré dans l'intersection mais n'a pas encore atteint la jonction de sortie. C'est donc un instant adéquat (ni trop tôt, ni trop tard) pour *NotifSo* compte tenu des besoins du détecteur de référentiels locaux.

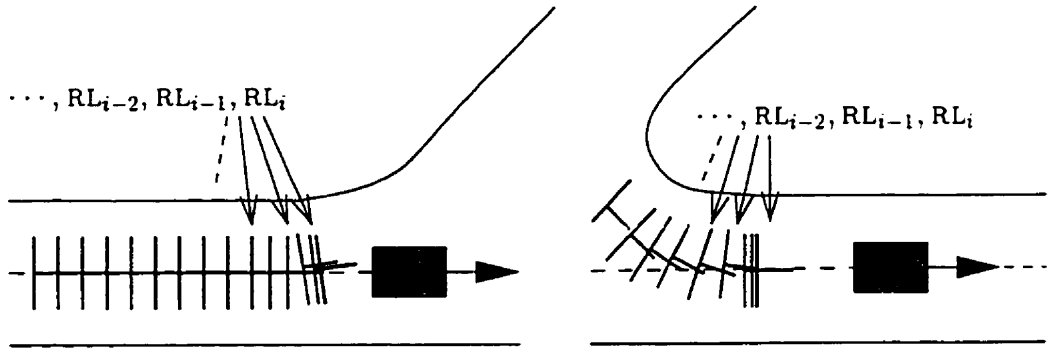
5.4.6 Signaux de confirmation de transition

Les confirmations de transition sont basées sur l'analyse des mesures latérales (*cf.* section 5.4.2). Nous devons identifier un événement dans l'évolution des mesures latérales qui caractérise de manière fiable les transitions de structure et que nous puissions localiser géométriquement de manière stable. Ceci dit, nous avons déjà une caractéristique associée aux jonctions qui satisfait le critère de stabilité géométrique : le référentiel local (RL), présenté au chapitre 4. Bien sûr, nous avons exploité la notion de RL par une détection *statique* sur des parois de jonction qui sont déjà cernées dans les mesures latérales. D'autre part, où que se trouve le robot, il pourra toujours calculer un RL sur les sections de parois qu'il vient de mesurer ; dans ce sens, la détection d'un RL n'est pas un événement intéressant en soi.

Par contre, si nous exploitons la notion de RL de façon *dynamique*, il devient possible de reconnaître une transition de structure. Plus précisément, il s'agit de faire la détection *périodiquement* sur une section de parois qui suit le véhicule, jusqu'à ce que les RLs atteignent une *position stable derrière le véhicule.* Ce principe est illustré dans la figure 5.13 pour les jonctions d'entrée et de sortie dans une intersection. Le processus de détection périodique de RLs est toujours déclenché par une notification de transition. Par conséquent, notre système peut distinguer une vraie transition de structure d'une structure quelconque ayant un RL stable (par exemple, un élargissement de galerie), car cette dernière n'aura pas été notifiée par l'analyse des profils frontaux.

5.4.7 Deuxième phase de l'analyse frontale

Nous avons défini *NotifSo* comme étant identique au signal *ConfirmEn*. Ainsi, selon la définition perceptuelle d'une jonction au chapitre 4, la jonction de sortie *perçue* débute en général peu après la fin des mesures latérales de la jonction d'entrée. Le détecteur de RLs fonctionnera correctement dans cet état des choses car, en agissant rétroactivement après la traversée, il pourra segmenter les parois de toute l'intersection et isoler



(a) Jonction d'entrée ; (b) Jonction de sortie.

Figure 5.13 : Détection périodique de RLs

la *vraie* jonction de sortie. Par contre, l'analyse pour *ConfirmSo* n'est pas rétroactive mais plutôt progressive durant la traversée d'intersection. N'ayant pas à l'avance la forme de toute l'intersection, ce processus sera induit en erreur dans certains cas.

Notamment, un problème particulier survient dans des intersections relativement longues. L'intersection d'essai X_2 employée dans les tests à la section 4.6 est un bon exemple. La figure 5.14 illustre la détection périodique de RLs depuis la position P_{NS} du robot (point M_f), à *NotifSo*, jusqu'à *ConfirmSo* (position P_{CS}). Ce dernier événement arrive beaucoup trop tôt dans la traversée, avant même le début de la vraie jonction de sortie. La cause en est que la largeur Dlat devient suffisamment stable au milieu de l'intersection pour que les RLs convergent derrière le véhicule et déclenchent le signal. L'analyse pour *ConfirmSo* a donc besoin d'une autre source d'information pour éviter le problème illustré ici.

La solution que nous trouvons est d'exploiter l'événement du nombre de branches revenant à 1 dans le champ frontal à la sortie de l'intersection. Dès *NotifEn*, une deuxième phase de l'analyse laisse accumuler une fenêtre d'image binaire "en avance" sur le robot[6], fenêtre qui suit ensuite le déplacement, jusqu'à ce que NR redescende

[6]c'est-à-dire que l'instant recherché dans l'intervalle $[t_1,t_2]$ de l'image traitée est t_1. Pour *NotifEn*,

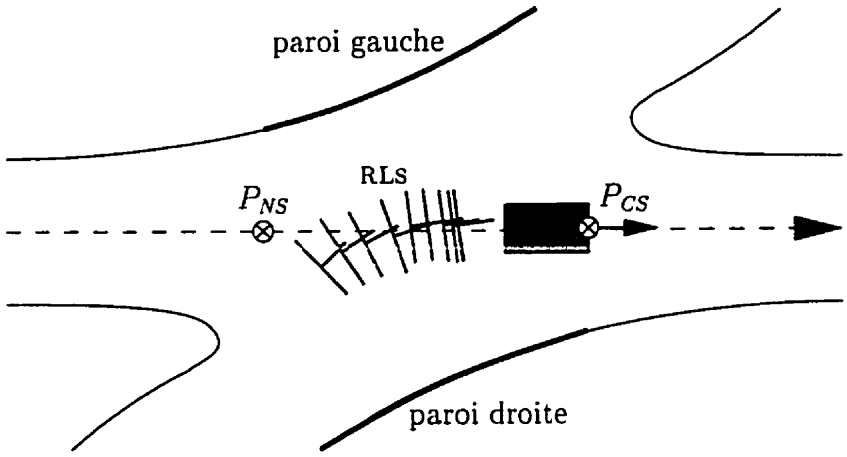


Figure 5.14 : Convergence des RLs au milieu d'une intersection

à 2 dans l'image. Nous dénotons ci-après cet événement par l'abréviation *1Branche*. Comme cet événement arrive trop tard dans l'intersection (question déjà vue), nous reculons d'une quantité fixe la position de trajectoire associée à *1Branche* pour être sûr qu'elle se situe avant la jonction d'entrée. L'analyse des mesures latérales commence à cette pose du robot (P_{1b}, qui est derrière la pose actuelle du robot). Nous pouvons dire que P_{1b} est une pose où le robot "ne verra bientôt qu'une seule branche".

Cette deuxième phase de l'analyse frontale est *non-causale*[7] dans le sens qu'elle exploite des mesures captées *après* l'événement (dans le temps ou dans la trajectoire) qu'elle doit reconnaître[8]. Autrement dit, l'instant t_{p2} quand l'événement *1Branche* est détecté arrive plus tard que l'instant de mesure t_{1b} lorsque le robot se trouvait en P_{1b}. Par conséquent l'analyse pour *ConfirmSo* doit connaître le décalage $(t_{p2} - t_{1b})$ afin d'éliminer les mesures latérales prises avant t_{1b}. D'autre part, pour assurer que

l'instant recherché dans $[t_1,t_2]$ est t_2 ; nous disons alors que l'image traitée est "en retard" sur le robot. Dans les deux cas, c'est l'instant t_2 qui correspond à la pose actuelle du robot et donc au profil frontal le plus récent.

[7]Nous employons les termes *causal* et *non-causal* ici par analogie à la notion de causalité en traitement du signal.

[8]Les analyses pour *NotifEn* et les confirmations sont au contraire *causales* : l'instant de détection et l'instant de mesure correspondant à la pose détectée sont les mêmes.

ConfirmSo soit, elle, causale, nous imposons que son instant de détection, t_{CS}, et l'instant de mesure t'_{CS} correspondant à la pose P'_{CS} trouvée, soient tels que :

$$t'_{CS} = t_{CS} \quad \text{et} \quad t_{CS} \geq t_{p2} \quad : \text{causalité de } \textit{ConfirmSo}.$$

Ainsi, l'analyse latérale ne pourra pas identifier une pose P'_{CS} qui précède la pose P_{p2}. Nous reviendrons à ces notions à l'étape de l'implantation (section 5.5).

Une autre conséquence de la non-causalité de l'analyse pour *1Branche* est que nous n'exploitons pas cet événement pour *NotifSo*. Du point de vue de la cohérence du système, nous désirons que tous les signaux de transition soient causaux afin de simplifier les interactions entre le détecteur de transitions et les autres modules, notamment ceux de l'activité d'arpentage.

5.4.8 Décomposition du module en fonctions

Le schéma d'interaction des différentes fonctions du module détecteur de transitions se trouve à la figure 5.15. Les paragraphes qui suivent décrivent les fonctions (les boîtes dans l'organigramme) en termes d'entrées, sorties et rôle.

5.4.8.1 Détection des branches

Cette fonction prend chaque profil télémétrique du senseur frontal et en isole les occlusions (discontinuités en distance ρ) et les LVs (plages de directions φ sans retour de signal laser). Les branches sont identifiées par leurs types et leurs positions angulaires dans le profil (*cf.* tableau 5.5).

Tableau 5.5 : Fonction de détection des branches

entrée	profils télémétriques instantanés frontaux.
sortie	branches détectées : leurs types et positions angulaires dans les profils.

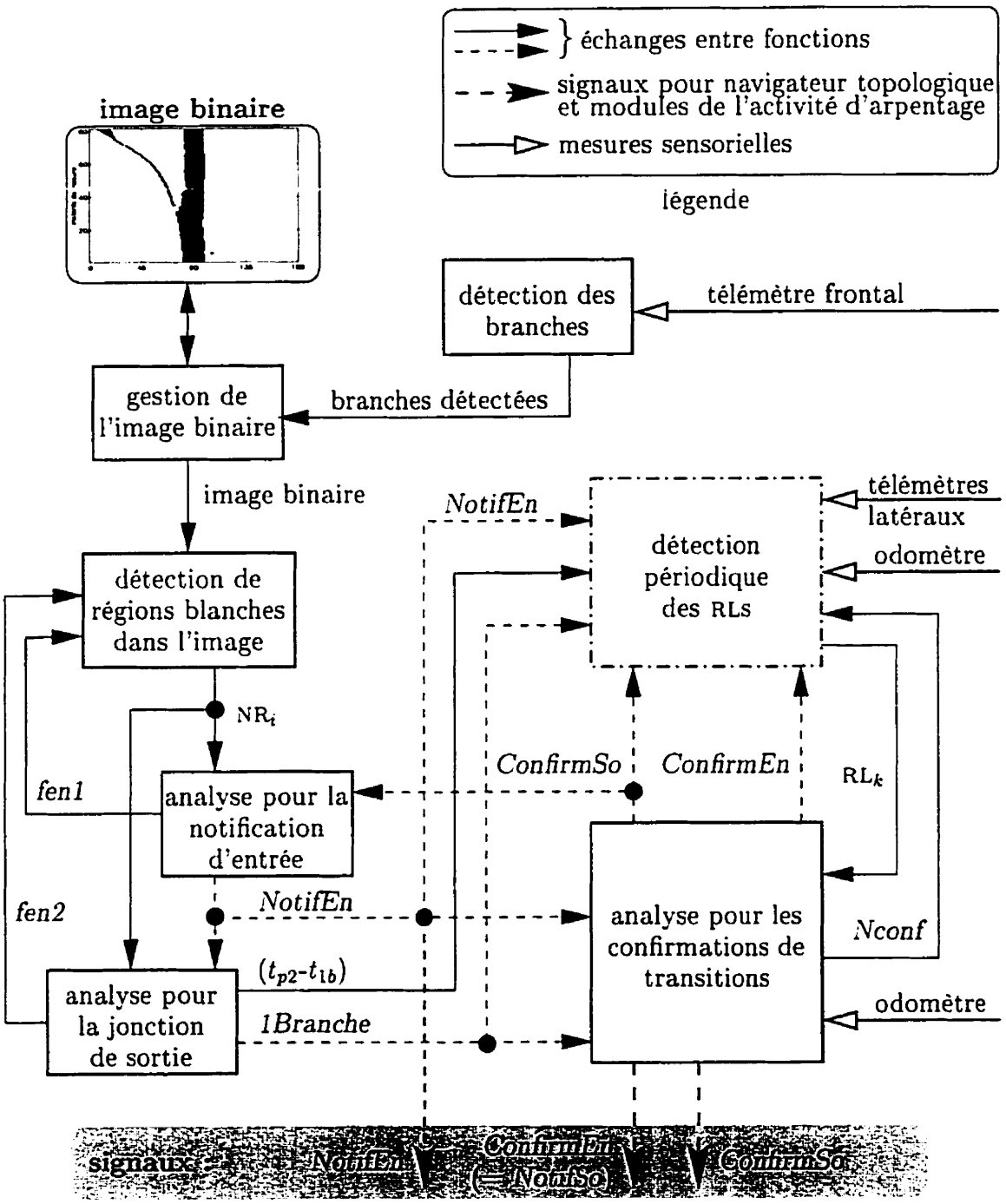


Figure 5.15 : Détecteur de transitions : échanges de données et signaux

5.4.8.2 Gestion de l'image binaire

L'image binaire est une représentation matricielle des profils frontaux traités au cours du temps. Elle prend la forme d'une file (PAPS) qui ne garde en mémoire qu'un nombre fixe des mesures les plus récentes. Cette fonction reçoit de la fonction précédente l'information des branches détectées à chaque instant de mesure sous forme d'un vecteur binaire (valeur 1 pour branche présente à la direction φ_p ; valeur 0 pour branche absente). Elle s'occupe de faire le "roulement" de ces vecteurs binaires dans la file et d'ajuster la longueur de la file (donc le nombre d'instants dans l'image binaire) selon la phase (1 ou 2) de l'analyse frontale en cours (*cf.* tableau 5.6). Nous ne jugeons pas utile de décrire davantage l'implantation de cette fonction à la section 5.5.

Tableau 5.6 : Fonction de gestion de l'image binaire

entrée	- branches détectées à chaque instant de mesure ; - nombre d'instants de mesure requis pour l'analyse frontale (*fen1* ou *fen2*).
sortie	image binaire sur la fenêtre temporelle (nombre d'instants) spécifiée.

5.4.8.3 Détection de régions blanches dans l'image

Cette fonction doit détecter le nombre de régions blanches NR_i dans l'image binaire ; cette information est mise à jour à chaque instant de mesure t_i (*cf.* tableau 5.7). Les mesures NR_i sont exploitées par les deux phases de l'analyse frontale.

Tableau 5.7 : Fonction de détection de régions blanches dans l'image

entrée	image binaire à l'instant de mesure t_i.
sortie	nombre de régions blanches NR_i à l'instant t_i.

5.4.8.4 Analyse pour la notification d'entrée

Cette première phase de l'analyse frontale est déclenchée par le signal *ConfirmSo*. Elle surveille la mesure NR_i, calculée sur un nombre d'instants *fen1* "en retard" sur le robot et envoie un signal *NotifEn* lorsque NR_i dépasse 2 (*cf.* tableau 5.8).

Tableau 5.8 : Fonction d'analyse pour la notification d'entrée

entrée	- signal *ConfirmSo* de l'intersection précédente ; - NR_i (mis à jour en continu).
sortie	- nombre d'instants requis pour l'image binaire (*fen1*) ; - signal de notification d'entrée : *NotifEn*.

5.4.8.5 Analyse pour la jonction de sortie

Cette deuxième phase de l'analyse frontale a pour but d'aider le processus pour *ConfirmSo* à localiser le début de la jonction de sortie dans les mesures latérales. Elle est déclenchée par le signal *NotifEn* et surveille la mesure NR_i, calculée sur un nombre d'instants *fen2* "en avance" sur le robot, jusqu'à l'événement *1Branche* (*cf.* tableau 5.9).

Tableau 5.9 : Fonction d'analyse pour la jonction de sortie

entrée	- signal *NotifEn* ; - NR_i (mis à jour en continu).
sortie	- nombre d'instants requis pour l'image binaire (*fen2*) ; - signal lorsque *1Branche* est détecté (instant t_{p2}) ; - décalage ($t_{p2} - t_{1b}$) associé à *1Branche*.

5.4.8.6 Analyse pour les confirmations de transition

L'analyse des mesures latérales est déclenchée soit par *NotifEn* (pour *ConfirmEn*), soit par l'événement *1Branche* (pour *ConfirmSo*). Elle suit l'évolution des RL_k, détectés à intervalle constant de *Nconf* instants sur les points de parois mesurés depuis le dernier événement déclencheur. Elle envoie le signal de confirmation lorsque les RL_k atteignent une position stable derrière le robot (*cf.* tableau 5.10).

Tableau 5.10 : Fonction d'analyse pour les confirmations de transition

entrée	- déplacements odométriques entre instants de mesure ; - signal *NotifEn* ; - signal de l'événement *1Branche* ; - les RL_k détectés périodiquement depuis l'événement déclencheur.
sortie	- *Nconf* : intervalle (en nombre d'instants) pour la détection des RLs ; - signal de confirmation : *ConfirmEn* ou *ConfirmSo*.

5.4.8.7 Détection périodique des RLs

Cette boîte dans la figure 5.15 contient en fait toute la fonctionnalité du détecteur de RLs, décrit au chapitre précédent aux sections 4.4.4 et 4.5, avec certaines modifications. La différence principale est que la détection de RLs est maintenant périodique à un taux spécifié par la fonction précédente. D'autre part, elle ne doit plus considérer le cas d'un mur commun d'intersection (*cf.* section 4.5.1). Lorsque le signal *ConfirmSo* ou *NotifSo* (= *ConfirmEn*) est reçu, la détection périodique s'arrête et les points de mesure en mémoire sont éliminés. Les signaux *NotifEn* et *NotifSo* relancent l'accumulation de points de mesure. La détection périodique est déclenchée par les signaux *NotifEn* et de *1Branche*. Lorsque ce dernier signal est reçu, les mesures latérales précédant l'instant t_{1b} sont d'abord éliminées (*cf.* tableau 5.11).

Toutefois, comme il s'agit toujours essentiellement de la même fonctionnalité que le détecteur de RLs au chapitre 4, cette partie de module n'est pas décrite davantage

Tableau 5.11 : Fonction de détection périodique des RLs

entrée	- profils télémétriques instantanés latéraux ; - déplacements odométriques entre instants de mesure ; - *Nconf* : intervalle (en nombre d'instants) pour la détection de RLs ; - signal *NotifEn* ; - signaux de confirmation : *ConfirmEn* et *ConfirmSo* ; - signal de l'événement *1Branche* ; - décalage dans le temps ($t_{p2} - t_{1b}$).
sortie	RL_k : les RLs détectés périodiquement au taux spécifié.

à la section 5.5.

5.5 Implantation du détecteur de transitions

Dans cette section, nous allons détailler les algorithmes développés au sein des fonctions que nous venons d'énumérer.

5.5.1 Détection des branches

5.5.1.1 Quelques constantes géométriques à définir

Tout d'abord, il convient de rappeler (*cf.* section A.3) que le profil télémétrique prend la forme d'une liste de N_{ray} points échantillonnés à intervalle angulaire constant $\delta\varphi$ dans le plan laser frontal. Le tableau 5.12 présente les définitions de plusieurs constantes utilisées dans la détection des branches.

Le schéma dans la figure 5.16 explique le calcul des constantes D_{occ} et φ_{LV} ; D_{occ} représente la distance entre deux points sur les parois gauche et droite d'une galerie de largeur G_{larg} et perçus à la portée maximale d_{max}. Comme l'apparition d'occlusions est normalement associée à des galeries de sortie dans une intersection, il est donc logique de relier le seuil D_{occ} à la largeur minimale d'une galerie, G_{larg}. L'écartement angulaire φ_{LV} est celui soutendu par un segment de longueur D_{occ} orthogonal à la

Tableau 5.12 : Constantes géométriques pour la détection des branches

constante	signification	calcul
G_{larg}	largeur minimale d'une galerie	
$\delta\varphi$	résolution angulaire des profils	$\alpha_{front}/(N_{ray}-1)$
D_{occ}	longueur minimale d'une occlusion	$G_{larg} - 2\,\delta\varphi\,d_{max}$
φ_{LV}	écart angulaire minimal d'une LV	$2\,[\arcsin(G_{larg}/(2\,d_{max})) - \delta\varphi]$
N_{LV}	nombre minimal de points d'une LV	$\varphi_{\text{LV}}/\delta\varphi + 1$

direction du robot et perçu à la portée d_{max}. Le terme en $(-2\,\delta\varphi)$ dans le calcul de D_{occ} et φ_{LV} rend compte de l'échantillonnage angulaire du plan frontal dans le profil.

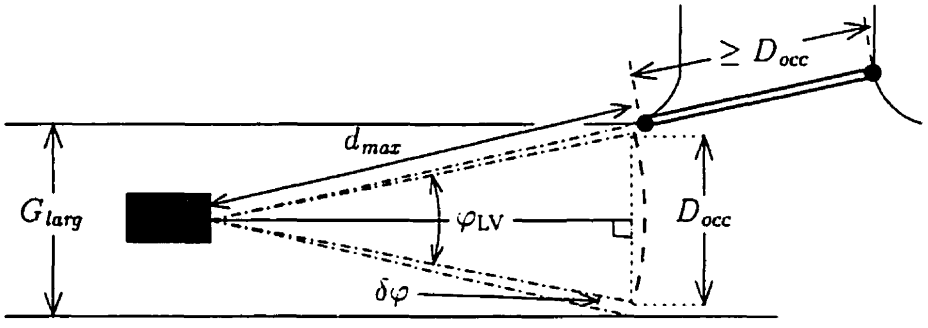


Figure 5.16 : Longueur d'un segment d'occlusion et angle soutendu par une LV

5.5.1.2 Quatre types de discontinuités dans les profils

Au départ, nous recherchons deux catégories distinctes de discontinuités dans le profil : les LVs qui sont des plages d'au moins N_{LV} directions φ sans mesure (représentées par des points sur le cercle de rayon d_{max}) (figure 5.17(a)) ; et les *occlusions fermées*, qui sont des paires de points consécutifs distants d'au moins D_{occ} (figure 5.17(b)). Cependant, la variété de profils présentés par l'environnement nous amène à faire des distinctions supplémentaires. En particulier, une plage sans mesure de moins de N_{LV} points est néanmoins intéressante si elle est cernée par des points distants d'au moins

D_{occ} ; nous appelons ce type de discontinuité une *occlusion ouverte*[9], par opposition à une occlusion fermée. Pour vérifier qu'il s'agit bien d'une vraie branche, nous faisons le test de la distance ($\geq D_{occ}$) sur plusieurs points de part et d'autre de la plage sans mesure (voir figure 5.18(a)). C'est-à-dire que l'occlusion ouverte doit satisfaire le critère :

$$\overline{P_g P_d},\ \overline{P_{g-1} P_d},\ \ldots,\ \overline{P_{g-m} P_d},\ \overline{P_g P_{d+1}},\ \ldots,\ \overline{P_g P_{d+m}} \overset{\text{tous}}{\geq} D_{occ}$$

où P_g et P_d sont les points cernant la discontinuité et m est une marge de validation. Lorsque la plage ($\rho_i = d_{max}$) est à l'extrémité du champ frontal, le nombre de points de mesure testables diminue à gauche ou à droite, mais une occlusion ouverte pourra toujours être détectée. La figure 5.18(b) montre par contre une fausse branche typiquement rejetée par le test pour les occlusions ouvertes : le dédoublement d'une LV par un point erroné à l'intérieur d'une plage ($\rho_i = d_{max}$) est évité grâce au test sur les distances.

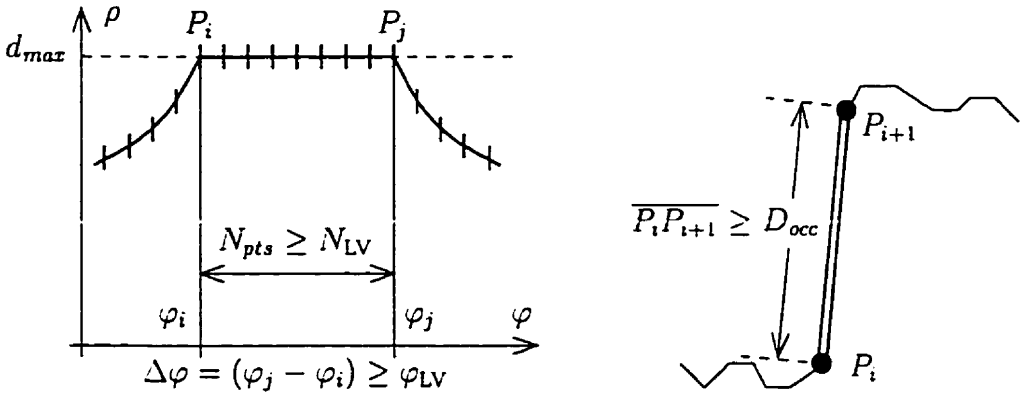
De la même manière, nous confirmons la validité d'une occlusion fermée en comparant les distances $\overline{P_{g-m} P_d},\ \ldots,\ \overline{P_{g-1} P_d},\ \overline{P_g P_d},\ \overline{P_g P_{d+1}},\ \ldots,\ \overline{P_g P_{d+m}}$ au seuil D_{occ}. Si le segment $[P_g P_d]$ ($= [P_i P_{i+1}]$) satisfait ce critère, nous considérons qu'il y a réellement occlusion entre deux parois (figure 5.19(a)). Sinon, le candidat d'occlusion fermée est rejeté comme fausse branche (figure 5.19(b)).

Nous nous retrouvons ainsi avec quatre types de discontinuités à reconnaître dans les profils frontaux : LV, occlusion ouverte, occlusion fermée et fausse branche. Ce sont les trois premiers types qui constituent les branches qui nous intéressent.

5.5.1.3 Algorithme de détection de branches

La série d'étapes suivante résume le processus de détection des branches :

[9] La LV, l'occlusion ouverte et l'occlusion fermée sont ainsi les trois manifestations possibles d'une ouverture (d'une branche) dans le champ frontal du robot.



(a) LV (coordonnées polaires) ; (b) Occlusion fermée.

Figure 5.17 : Définition d'une LV et d'une occlusion pour la détection

1. **Détection des LVs (type 1) :**

   (a) marquer toutes les plages ($\rho_i = d_{max}$) $\Longrightarrow$ masque `limvis` ;

   (b) plages telles que : $\Delta\varphi \geq \varphi_{LV}$, $N_{pts} \geq N_{LV}$ $\Longrightarrow$ masque `branches` ;

   (c) autres plages : masque `non_LV` = `limvis` ET (NON `branches`).

2. **Détection des occlusions ouvertes (type 2) :**

   Pour chaque plage dans `non_LV` : *Validation des occlusions ouvertes.*

   (a) isoler les points P_g, P_d (cas d'une plage à l'extrémité du profil) ;

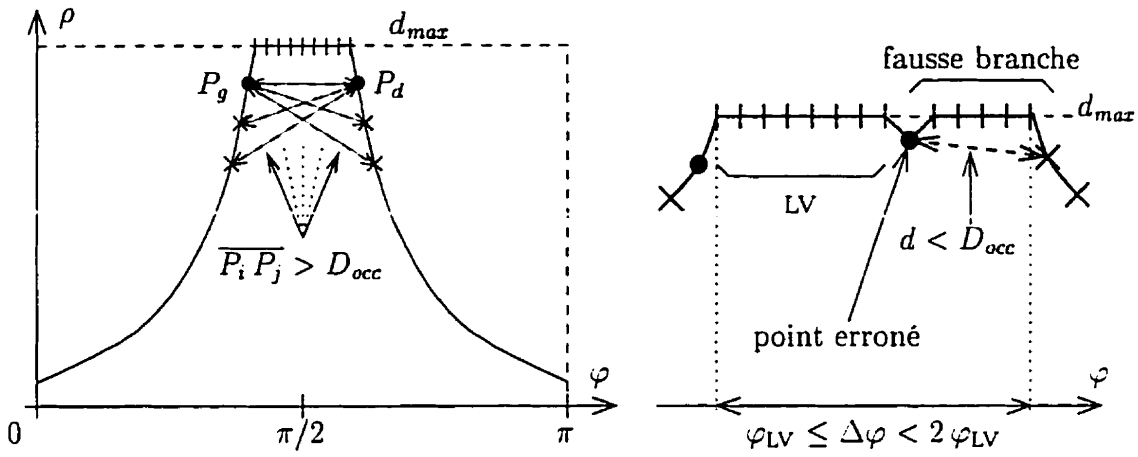
   (b) test sur les distantes : $\overline{P_{g-m}\,P_d}$, ... , $\overline{P_g\,P_d}$, ... , $\overline{P_g\,P_{d+m}} \geq D_{occ}$;

   (c) plages validées $\Longrightarrow$ rajouter à `branches`.

3. **Détection des occlusions fermées (type 3) :**

   (a) masque excluant les branches de type 1 et 2 :

   `detectoccl` = (NON (plages élargies dans `branches`)) ;



(a) Test sur les distances ; (b) Candidat rejeté (fausse branche).

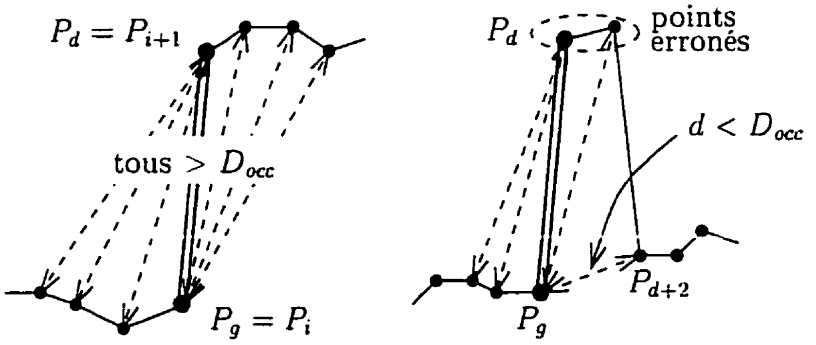
Figure 5.18 : Validation des candidats d'occlusion ouverte

(b) trouver segments $[P_i\, P_{i+1}]$ de longueur $\geq D_{occ}$ dans profil masqué par `detectoccl` ;

(c) validation des candidats d'occlusion fermée (test sur les distances) :
- segments $[P_i\, P_{i+1}]$ validés $\Longrightarrow$ rajouter à `branches`.

À la fin de l'algorithme, le masque `branches` contient l'information voulue : le nombre de branches trouvées NB (types 1,2 et 3) et leurs positions dans le profil. C'est un vecteur binaire de longueur N_{ray} qui formera une rangée de pixels dans l'image binaire. La figure 5.20 montre un exemple de détection de branches : le robot dans l'environnement et le profil, en coordonnées cartésiennes et polaires, qui contient une LV (en tirets gras)[10] et une occlusion fermée (en trait plein gras).

[10]Dans l'algorithme, nous prenons une valeur d'_{max} légèrement inférieure au d_{max} nominal (15 mètres dans cet exemple) pour détecter les plages pour le masque `limvis`. Ceci explique que la LV dans les figures 5.20(b) et 5.20(c) contient quelques points qui ne sont pas sur le cercle de rayon d_{max}.



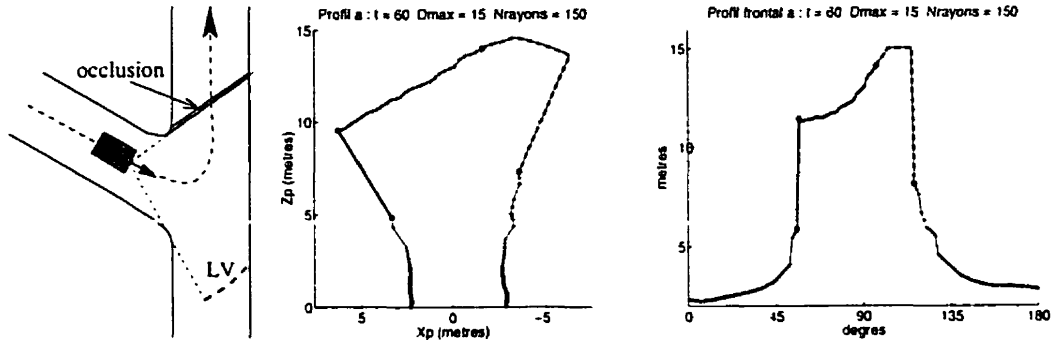
(a) Véritable occlusion ; (b) Fausse occlusion.

Figure 5.19 : Validation des candidats d'occlusion fermée

5.5.1.4 Cas de mauvaise interprétation

Il a été signalé à la section 5.4.4 que la fonction de détection de branches peut commettre des erreurs dues à la mauvaise interprétation des mesures. De fait, la méthode que nous venons de présenter se veut heuristique et n'est pas garantie de produire des résultats toujours corrects. Elle est notamment sensible à la présence de points de mesure erronés, ce qui peut l'amener à rejeter des vraies branches ou à en accepter de fausses. La figure 5.21 illustre deux cas posant problème aux étapes 1 et 2 de notre algorithme. Dans la figure 5.21(a), une seule LV relativement large ($\Delta\varphi > 2\,\varphi_{LV}$) est dédoublée (perçue comme deux LVs plus étroites) à cause d'un point de $\rho_p < d_{max}$ au milieu de la plage sans mesure. Au contraire, dans la figure 5.21(b), un point erroné au milieu d'une LV plus étroite fait que celle-ci est rejetée par les deux tests aux étapes 1 et 2.

L'étape 3 de l'algorithme est elle aussi vulnérable aux artefacts de mesure. Dans la figure 5.22(a), une occlusion fermée relativement longue ($d_{tot} \geq 2\,D_{occ}$) interrompue par un point "flottant dans le vide" est perçue comme deux occlusions successives. Dans la figure 5.22(b), une occlusion ouverte potentielle est rejetée à l'étape 2 à cause des points P_g et P_d proches de la limite d_{max} ; par conséquent, l'étape 3 détecte deux



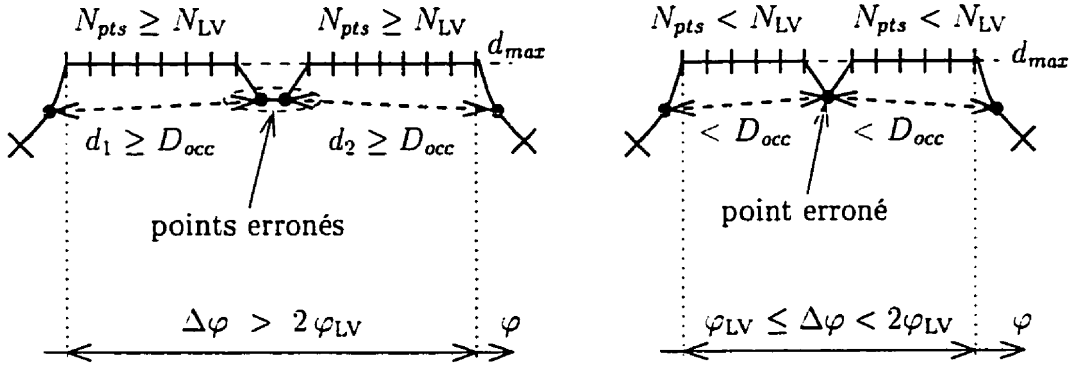
(a) situation ; (b) coordonnées cartésiennes ; (c) coordonnées polaires.

Figure 5.20 : Exemple de détection de branches

occlusions fermées en les segments $[P_a\, P_g]$ et $[P_d\, P_b]$.

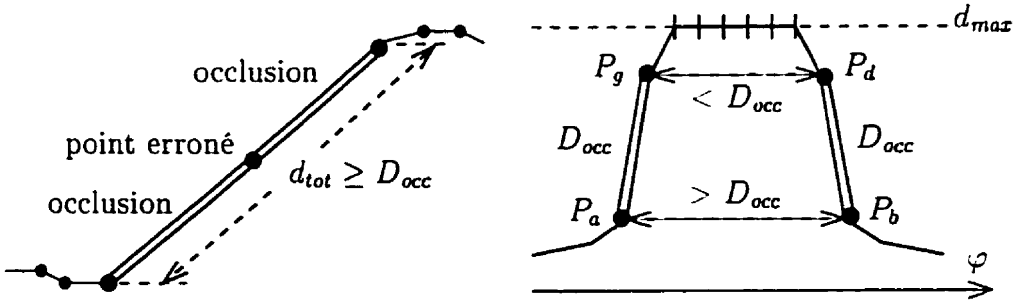
D'autre part, les résultats de la détection de branches sont nécessairement dépendants du choix des seuils et surtout de la valeur de D_{occ} d'où découlent φ_{LV} et N_{LV}. Bien qu'expérimentalement des valeurs "raisonnables" pour ces constantes soient choisies, elles ne peuvent être optimalement adaptées à tous les cas de géométrie de scène que le robot va rencontrer.

Il pourrait sembler avantageux d'améliorer notre approche de segmentation des profils. Nous pourrions chercher à faire une analyse plus "intelligente" des données, soit en multipliant les tests au sein de l'algorithme actuel, soit en menant une étude plus approfondie des relations entre la géométrie de scène et la formation des profils télémétriques. Nous pourrions aussi développer une approche qui soit plus robuste à la présence de points erronés et aux cas de structure dégénérés, par exemple une analyse multi-échelles effectuant des filtrages successifs du profil. Mais nous allons voir que l'algorithme actuel est adéquat compte-tenu de la représentation des résultats sous forme d'image binaire et de la redondance temporelle exploitée dans les analyses pour générer les signaux.



(a) Dédoublement d'une LV large ; (b) Rejet d'une LV plus étroite.

Figure 5.21 : Sensibilité aux points erronés dans les plages sans mesure



(a) Dédoublement d'une occlusion longue ; (b) Occlusion ouverte traitée comme deux occlusions fermées.

Figure 5.22 : Erreurs dans la détection des occlusions fermées

5.5.2 Détection des régions blanches dans l'image

En vue de la supposition que nous avons faite à la section 5.4.4 sur des profils sans bruit de capture, l'image obtenue par accumulation des vecteur binaires provenant de la fonction précédente est relativement "propre" : le taux de pixels erronés est faible. Par conséquent, il n'est pas nécessaire d'implanter un véritable algorithme de segmentation d'image en régions uniformes (accompagné de "nettoyage" du bruit). Au lieu de cela, nous implantons une idée beaucoup plus simple, à savoir de suivre le *contour* de l'image binaire et de compter le *nombre de changements de pixel blanc à noir* (ou de noir à blanc, les deux nombres étant identiques, voir la figure 5.23).

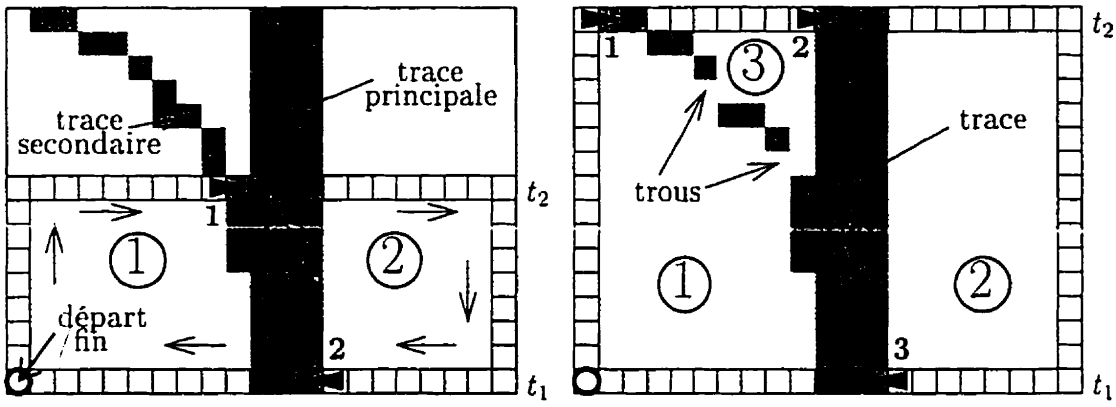
Une étude des différentes topologies de la trace des profils ainsi que des différentes positions de l'image binaire par rapport à la traversée d'intersection nous mène au résultat suivant. Soit N_{bn} le nombre de changements de blanc à noir dans le contour de l'image (hormis les artefacts, auxquels nous reviendrons ci-après), et NR_{blanc} et NR_{noir} les nombres de régions respectivement blanches et noires composant l'image. Nous avons la relation :

$$N_{\text{bn}} = \text{NR}_{blanc} + \text{NR}_{noir} - 1 .$$

Aussi, la topologie de la trace des profils est telle que si l'intervalle $[t_1, t_2]$ de l'image traitée commence là où le robot n'observe qu'une seule branche, alors il n'y a qu'une seule région noire (même si celle-ci contient des traces secondaires). La relation précédente se simplifie à : $N_{\text{bn}} = \text{NR}_{blanc}$; le nombre N_{bn} nous donne directement le nombre de régions blanches. C'est le cas dans les images de la figure 5.23 ci-dessous.

Cependant, durant la deuxième phase de l'analyse frontale, l'intervalle $[t_1, t_2]$ peut aussi bien commencer alors que le robot voit déjà plusieurs branches. Dans ce deuxième cas, il n'y a plus équivalence entre N_{bn} et NR_{blanc}[11]. Pour cette raison, la

[11] Malgré l'information *a priori* du degré de l'intersection dans le graphe, le nombre de régions noires NR_{noir} dans la fenêtre traitée est néanmoins imprévisible.

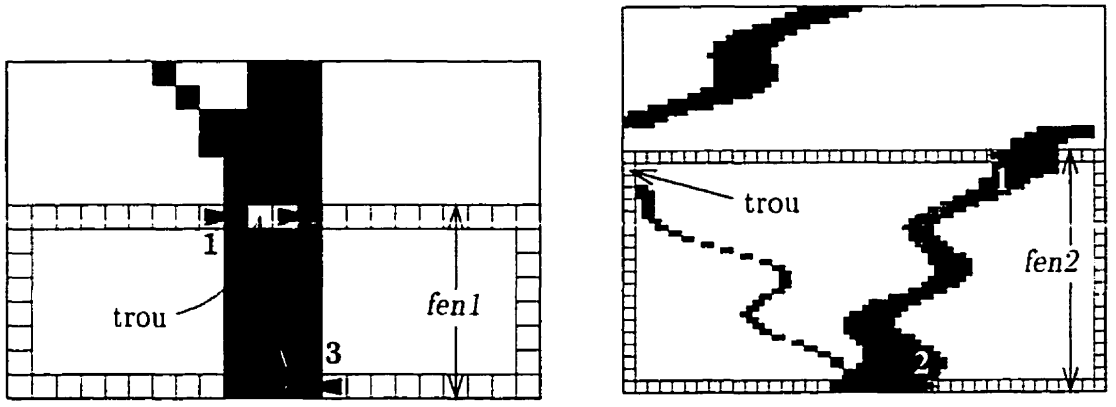


(a) $N_{bn} = 2$ et $NR_{blanc} = 2$; (b) $N_{bn} = 3$ et $NR_{blanc} = 3$.

Figure 5.23 : Changements de pixel blanc à noir dans le contour de l'image binaire

mesure NR_i fournie aux analyses frontales n'aura que deux valeurs significatives : a) $NR_i = 2$ et b) $NR_i > 2$.

La trace "idéale" des profils (*e.g.* celle de la figure 5.23(a)) subit des distorsions liées à deux facteurs : la détection des branches pourra y introduire des perturbations locales, tandis qu'une trajectoire déviée y imposera une déformation globale. Notre méthode pour compter NR_i a l'avantage d'être robuste par rapport aux distorsions *à l'intérieur de* l'image traitée. Par exemple, les trous dans la trace secondaire de la figure 5.23(b) n'affectent pas le nombre $N_{bn} = 3$. En revanche, le calcul de NR_i sera forcément affecté lorsque des distorsions se manifestent dans le *contour* de l'image traitée. Nous en avons deux illustrations à la figure 5.24. Dans le premier cas (5.24(a)) le dédoublement d'une LV (rappel de la figure 5.21(a)) à la limite de la fenêtre d'analyse pour *NotifEn* fait monter NR_i à (>2) avant que le robot ne perçoive l'intersection. Dans le second cas (5.24(b)), l'analyse pour *1Branche* détectera cet événement trop tôt car les influences combinées de la trajectoire et de la détection des branches font disparaître brièvement la trace secondaire à la gauche de l'image binaire.



(a) Analyse pour *NotifEn* ; $NR_i > 2$; (b) Analyse pour *1Branche* ; $NR_i = 2$.

Figure 5.24 : Influence de la détection des branches et de la trajectoire sur NR_i

Ces deux derniers cas de figure nous font voir que les deux phases de l'analyse frontale doivent exploiter *doublement* la redondance temporelle : de manière implicite dans la représentation de l'image binaire, mais aussi de manière explicite en observant la valeur recherchée de la mesure NR_i sur *plusieurs* instants consécutifs.

5.5.3 Analyse pour la notification d'entrée

L'analyse pour générer le signal *NotifEn* doit reconnaître les situations comme dans la figure 5.23(b) ci-haut : la mesure NR_i, évaluée sur un nombre d'instants *fen1*, augmente de 2 à (>2) car le robot voit l'arrivée d'une intersection. En même temps, l'analyse doit éviter de générer de fausses alertes dans des cas comme la figure 5.24(a) ci-haut (présence d'artefacts). Selon le principe de la redondance temporelle, le signal *NotifEn* n'est donc envoyé que lorsque ($NR_i > 2$) a été observé pendant *nbIns1* instants consécutifs. Remarquons que les traces secondaires doivent apparaître vers le centre du champs frontal si le robot suit une trajectoire idéale dans la galerie. Si sa trajectoire est déviée, celle-ci ne pourra quand même pas faire disparaître la trace secondaire de l'image (comme dans la figure 5.24(b)) et interrompre l'observation ($NR_i > 2$).

Le nombre d'instants *fen1* doit être supérieur à *nbIns1* afin d'assurer que le début de l'image traitée précède toujours l'apparition de traces secondaires. Les valeurs choisies expérimentalement pour ces paramètres vont refléter cette contrainte (section 5.6).

5.5.4 Analyse pour la jonction de sortie

L'événement *1Branche* survient lorsque toute trace secondaire a disparu de la fenêtre de temps $[t_i, t_i + (fen2\,\Delta t)]$[12] et que celle-ci ne contient plus qu'une trace principale (la nouvelle galerie). Comme dans l'analyse précédente, nous attendons d'observer ($\text{NR}_i = 2$) pendant *nbIns2* instants consécutifs afin d'éviter des fausse alertes causées par des situations telles que la figure 5.24(b). Quand le signal pour *1Branche* est envoyé, le robot (point M_f) a circulé dans la nouvelle galerie pendant une durée de $(fen2 + nbIns2)\,\Delta t$ depuis l'instant $t_i = t_f$ où la mesure NR_i pour la fenêtre $[t_i, t_i + (fen2\,\Delta t)]$ est définitivement redescendue à 2. À partir de t_f, nous remontons dans le temps de *nbRecul* instants pour déterminer t_{1b}. Au total, il y a un décalage entre l'instant courant, t_{p2}, et l'instant t_{1b}, de :

$$(t_{p2} - t_{1b}) = (fen2 + nbIns2 + nbRecul)\,\Delta t\ .$$

Le décalage a donc une valeur constante; cependant, les valeurs finales de t_{p2} et t_{1b} vont dépendre de t_f. Normalement, l'image binaire formée en débouchant dans la nouvelle galerie ne subit pas de distorsions du contour et t_f correspond au moment où toutes les galeries autres que celle de sortie disparaissent du champ frontal (*cf.* figure 5.11). Si l'image subit des perturbations (cas peu probable), alors la détection de *1Branche* sera retardée.

Comme il est sûr qu'à partir de t_f la fenêtre d'analyse sera entièrement au-delà de l'intersection, nous n'avons pas la contrainte que *fen2* soit plus longue que *nbIns2*

[12] Δt est la durée (constante) entre instants consécutifs de mesure.

comme dans l'analyse pour *NotifEn*.

5.5.5 Analyse pour les confirmations de transitions

5.5.5.1 Définition des paramètres utilisés

Le signal de confirmation *ConfirmEn* ou *ConfirmSo* est envoyé lorsque les RL_k détectés périodiquement satisfont deux critères simultanément : la stabilité géométrique et la convergence à un endroit derrière le véhicule. Le tableau 5.13 définit les paramètres utilisés dans l'évaluation de ces deux critères. Rappelons aussi que *Nconf* est l'intervalle (en nombre d'instants de mesure) pour la détection des RL_k.

Tableau 5.13 : Paramètres utilisés dans l'analyse pour les confirmations

critère	paramètre	signification
stabilité géométrique	$max\Delta pos$	déplacement maximal permis entre RL_{k-1} et RL_k
	$max\Delta\theta$	changement d'orientation maximal permis entre RL_{k-1} et RL_k
	minItersStab	nombre d'itérations que ce critère doit être vérifié
rester derrière le véhicule	$\overline{M_f\,\mathrm{RL}}_{min}$	distance minimale requise entre les RL_k et M_f
	minItersDerr	nombre d'itérations que ce critère doit être vérifié

5.5.5.2 Algorithme pour la confirmation de transition

Le processus pour générer la confirmation de transition est comme suit :

1. **Rentrer les nouvelles mesures à l'itération k :**
   - ensRLs = [ensRLs ; RL_k=$(X, Z, \theta)_k$] ; *dernier* RL *calculé.*
   - $(\Delta x, \Delta z, \Delta\theta)_{odom}$, pose à (k−1) $\Longrightarrow$ nouvelle pose de M_f : $(X, Z, \theta)_R$;
2. **Évaluer le critère de la stabilité géométrique :**

- $\Delta pos = \sqrt{(X_k - X_{k-1})^2 + (Z_k - Z_{k-1})^2}$; $\Delta\theta = |\theta_k - \theta_{k-1}|$;
- **si** $(\Delta pos \leq max\Delta pos) \implies itersPosStab = itersPosStab + 1$;
  sinon $\implies itersPosStab = 0$;
- **si** $(\Delta\theta \leq max\Delta\theta) \implies iters\theta Stab = iters\theta Stab + 1$;
  sinon $\implies iters\theta Stab = 0$;
- $itersStab = \min(itersPosStab, iters\theta Stab)$.

3. **Évaluer le critère de rester derrière le véhicule :**

- $\overline{M_f\,\mathrm{RL}} = \sqrt{(X_R - X_k)^2 + (Z_R - Z_k)^2}$; *distance entre* RL_k *et* M_f.
- **si** $(\overline{M_f\,\mathrm{RL}} \geq \overline{M_f\,\mathrm{RL}}_{min}) \implies itersDerr = itersDerr + 1$;
  sinon $\implies itersDerr = 0$.

4. **Les deux critères sont-ils satisfaits simultanément ?**

- **si** $((itersStab \geq minItersStab)$ ET $(itersDerr \geq minItersDerr))$
  $\implies$ **envoyer signal** *ConfirmEn* ou *ConfirmSo* ; **arrêter.**
  sinon $\implies$ k = k + 1 ; $t = t + (Nconf\,\Delta t)$; **retourner à l'étape 1.**

5.5.6 Déroulement normal dans l'environnement

La figure 5.25 illustre le déroulement normal de la détection des transitions au cours de la traversée d'une intersection. Ce schéma indique les zones de convergence des RLs dans les jonctions d'entrée et sortie ; les positions du robot (M_f) aux signaux de transition : P_{NE}, P_{CE}=P_{NS}, P_{CS} ; et les positions du robot P_{1b} et P_{p2}, aux instants t_{1b} et t_{p2} associés à l'événement *1Branche*. La distance D_E dans la figure représente l'écart entre la zone de convergence d'entrée et P_{CE}, et D_S l'écart entre les positions P_{p2} et P_{CS}. Les bornes inférieures sur ces deux distances ont les expressions suivantes en fonction des paramètres utilisés dans les analyses :

$$D_E \geq \max((minItersStab - 1) \times (d_{arp}\,Nconf)\,,$$

$$\overline{M_f \, \mathrm{RL}}_{min} + (minItersDerr - 1) \times (d_{arp}\, Nconf)) ;$$

$$D_S \geq (\max(minItersStab,\ minItersDerr) - 1) \times (d_{arp}\, Nconf) .$$

La formule pour D_S ci-dessus suppose que dès que l'analyse pour *ConfirmSo* commence, les RL_k atteignent une position stable (dans la zone de convergence de sortie) à au moins $\overline{M_f \, \mathrm{RL}}_{min}$ derrière le robot.

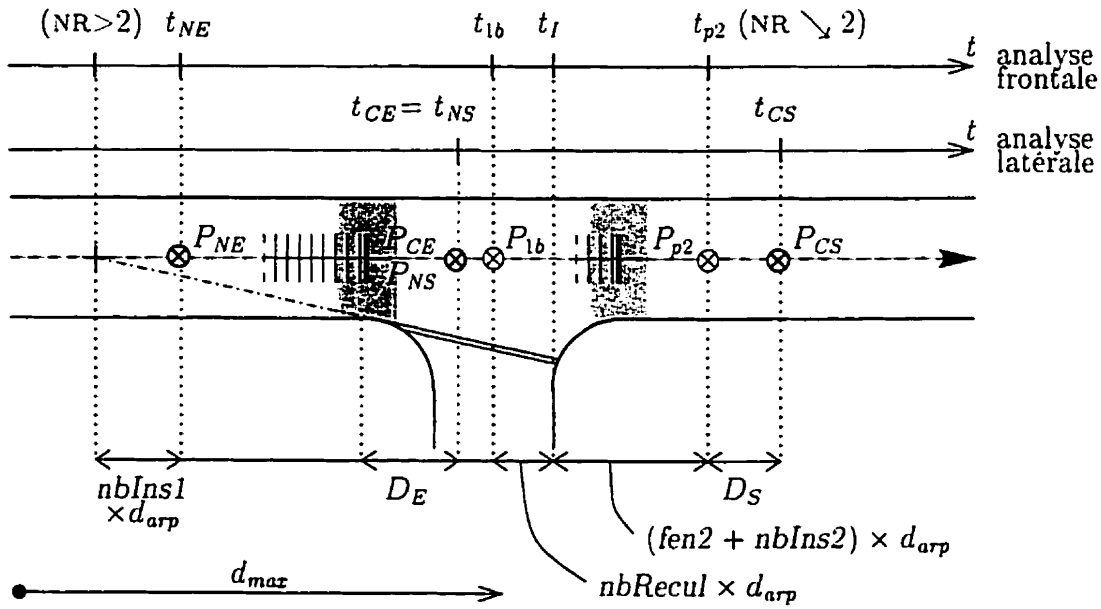


Figure 5.25 : Processus complet de détection des transitions

D'autre part, un critère d'évaluation intéressant de notre conception du détecteur de transitions est la longueur minimale requise des galeries séparant les intersections rencontrées afin que ce module fonctionne correctement. Pour reconnaître l'événement *1Branche*, il faut observer (NB=1) pendant (*fen2*+*nbIns2*) instants après l'instant t_l. D'autre part, l'analyse pour *NotifEn* commence à percevoir (NB>1) dès $(d_{max}/d_{arp}) \cdot \Delta t$ avant l'instant t_l de la prochaine intersection (figure 5.25 ci-dessus). En supposant que l'image binaire exploitée à t_{p2} pour *1Branche* est exploitée aussi pour *NotifEn* de

l'intersection suivante, la distance minimale entre intersections est

$$(fen2 + nbIns2)\, d_{arp} \;+\; d_{max} \;[+\; D_S].$$

Le dernier terme (entre crochets) dans l'expression s'élimine si nous supposons de plus que l'analyse pour *NotifEn* peut commencer simultanément avec celle pour *ConfirmSo*. Toutefois, le terme principal dans l'expression est d_{max}, la portée maximale du senseur frontal.

5.6 Résultats de la détection des transitions

5.6.1 Cas expérimentaux

5.6.1.1 Intersections et trajectoires

Nous testons le module détecteur de transitions dans les mêmes cinq intersections d'essai employées au chapitre précédent pour le détecteur de RLs (*cf.* section 4.5.7) ; la figure 5.26 remontre les cinq intersections. Nous employons cette fois-ci 14 chemins topologiques différents, qui sont indiqués dans le tableau 5.14.

Tableau 5.14 : Les 14 chemins de traversée d'intersections

	I_1	I_7	I_{10}	X_2	X_3
chemins	$G_1 \rightarrow G_2$ $G_1 \rightarrow G_3$ $G_2 \rightarrow G_3$	$G_1 \rightarrow G_2$ $G_3 \rightarrow G_2$	$G_1 \rightarrow G_2$ $G_2 \rightarrow G_3$	$G_1 \rightarrow G_3$ $G_1 \rightarrow G_4$ $G_4 \rightarrow G_2$	$G_1 \rightarrow G_3$ $G_2 \rightarrow G_3$ $G_3 \rightarrow G_1$ $G_3 \rightarrow G_2$

Une fois de plus, chacun de ces chemins engendre 33 cas de test au moyen d'une déviation sinusoïdale d'amplitude, de période et de délai de phase variables appliquée à la trajectoire idéale (rappel du tableau 4.18).

5.6.1.2 Valeurs numériques des paramètres

Le tableau 5.15 énumère les valeurs numériques choisies pour les paramètres de base relatifs au véhicule et au senseur frontal ainsi que les paramètres de base pour les différentes fonctions dans la détection des transitions.

Tableau 5.15 : Valeurs numériques des paramètres de base

paramètre	signification	valeur [unités]
R_{long}	longueur du véhicule	3 mètres
d_{max}	limite de portée du senseur frontal	20 mètres
α_{front}	écartement angulaire du plan laser frontal	180°
N_{ray}	nombre de points dans un profil frontal	150 points
d_{arp}	distance parcourue entre instants de mesure	25 centimètres
G_{larg}	largeur minimale (supposée) d'une galerie	4 mètres
$Nconf$	nombre d'instants pour la détection des RL_k	2 instants

Le tableau 5.16 donne les valeurs choisies pour les paramètres utilisés dans les deux phases de l'analyse frontale et dans l'analyse pour les confirmations. La plupart des valeurs dans ce deuxième tableau sont exprimées en fonction des paramètres de base, surtout d_{arp}et R_{long}. Par exemple, *fen1*, le nombre d'instants dans l'image binaire exploitée par l'analyse pour *NotifEn*, vaut l'équivalent d'une longueur de véhicule R_{long}.

5.6.2 Résultats et analyse

La série de graphiques dans les figures 5.27, 5.28 et 5.29 montre les résultats de la détection des transitions pour les 14 chemins. Pour chaque cas de traversée d'intersection, nous voyons d'une part les points de mesure latéraux (les parois) prises suivant la trajectoire non-déviée et d'autre part les poses trouvées par les différentes analyses : P_{NE} (symboles +), P_{CE}=P_{NS} (symboles o), P_{p2} (symboles ×) et P_{CS} (symboles ◇). Chaque nuage de symboles identiques dans un graphique correspond

Tableau 5.16 : Valeurs des paramètres pour les analyses frontale et latérale

paramètre	signification	valeur
fen1	instants dans l'image binaire pour *NotifEn*	$2\,R_{long}/d_{arp}$
nbIns1	instants que ($\mathrm{NR}_i > 2$) doit être observé	R_{long}/d_{arp}
fen2	instants dans l'image binaire pour *1Branche*	R_{long}/d_{arp}
nbIns2	instants que ($\mathrm{NR}_i = 2$) doit être observé	R_{long}/d_{arp}
nbRecul	instants reculés de t_f à t_{1b}	R_{long}/d_{arp}
$max\Delta pos$	limite de déplacement entre RL_k consécutifs	65 centimètres
$max\Delta\theta$	limite de changement d'orientation entre RL_k	10°
minItersStab	itérations que les RL_k doivent être stables	$R_{long}/(Nconf\, d_{arp})$
$\overline{M_f\,\mathrm{RL}}_{min}$	distance minimale entre les RL_k et M_f	R_{long}
minItersDerr	itérations que les RL_k doivent être derrières	$1{,}0/(Nconf\, d_{arp})$

ainsi aux poses calculées par l'analyse correspondante pour les 33 cas de trajectoire.

Le tableau 5.17 montre les statistiques de la longueur des jonctions d'entrée et de sortie pour les 14 cas de traversée d'intersection. Ces longueurs sont calculées comme les différences $(t_{CE} - t_{NE})(d_{arp}/\Delta t)$ pour la jonction d'entrée et $(t_{CS} - t_{1b})(d_{arp}/\Delta t)$ pour la jonction de sortie. À chaque rangée du tableau, les valeurs statistiques, à savoir le minimum (colonne LJ_{min}), le maximum (colonne LJ_{max}), la moyenne (colonne $\overline{LJ}$) et l'écart-type (colonne σ_{LJ}), sont calculées sur la distribution des 33 cas de trajectoire. La dernière rangée indique les statistiques calculées sur l'ensemble des 14 chemins.

D'après la première série de graphiques, nous observons que dans tous les cas expérimentaux le processus de détection des transitions fonctionne bien selon le déroulement normal illustré à la figure 5.25. Pour ce qui est de l'analyse frontale, le signal *NotifEn* a une bonne stabilité par rapport à la trajectoire (nuage relativement resserré) dans toutes les traversées. L'événement *1Branche* a aussi une bonne stabilité par rapport à la trajectoire, sauf pour les chemins $I_1 : G_1 \rightarrow G_3$ et $X_3 : G_3 \rightarrow G_2$ où l'instant t_{p2} dans certains cas vient nettement plus tôt que la moyenne des 33 cas. Ce

phénomène est très prononcé pour $X_3 : G_3 \rightarrow G_2$: nous remarquons en effet quelques symboles $\times$ loin du nuage P_{p2} dans le graphique ainsi qu'une valeur élevée de σ_{LJ} (2,29 mètres) pour la jonction de sortie. Ceci veut dire que le signal pour *1Branche* est parfois déclenché trop tôt, c'est-à-dire par ($\text{NR}_i = 2$) durant *nbIns2* instants *avant* la fin de l'intersection. Mais le processus pour *ConfirmSo* fonctionne correctement dans ces cas malgré une jonction de sortie qui commence plus tôt que normalement.

Pour l'analyse latérale, nous observons une bonne stabilité en général par rapport à la trajectoire. Cependant, les traversées d'intersection à virage accentué ($\geq$ 90°) sont caractérisées par une plus grande dispersion du nuage P_{CE}. Ceci est particulièrement visible dans le cas des chemins $I_{10} : G_2 \rightarrow G_3$ et $X_3 : G_2 \rightarrow G_3$. Une des cause de cette moins bonne stabilité est la variation de la partie non-visible du mur commun en fonction de la déviation appliquée à la trajectoire ; l'endroit précis où convergent les RL_k, et donc la pose P_{CE}, varient de la même manière. Mais le séquencement des analyses subséquentes (pour *1Branche* et *ConfirmSo*) n'est pas affecté par celui de *NotifEn*. Ces derniers cas de figure font ressortir l'indépendance entre les événements perceptuels relatifs à l'entrée dans l'intersection et ceux relatifs à la sortie de l'intersection : les événements sont rattachés à l'environnement et non au senseur ni à la trajectoire suivie.

D'après les statistiques du tableau 5.17, nous voyons que la longueur de la jonction de sortie est souvent plus stable en fonction de la trajectoire que celle de la jonction d'entrée (colonnes σ_{LJ}). Il y a donc un "couplage plus fort" entre *1Branche* et *ConfirmSo* qu'entre *NotifEn* et *ConfirmEn*. Nous pouvons expliquer ceci par le fait que l'analyse pour *ConfirmEn* va "en rentrant dans l'intersection" alors que celle pour *ConfirmSo* va "en rentrant dans une galerie". Ainsi, pendant la phase de l'analyse latérale où les RL_k atteignent leur zone de convergence, les points de parois rajoutés à la jonction d'entrée (qui sont dans l'intersection) sont davantage susceptibles d'affecter le calcul du RL (et donc l'évaluation des deux critères pour la confirmation de transition) que ceux rajoutés à la jonction de sortie (qui sont dans une galerie).

Numériquement, la longueur de la jonction d'entrée varie entre 5,5 mètres (LJ_{min} pour $X_3 : G_1 \rightarrow G_3$ et $X_3 : G_2 \rightarrow G_3$) et 13,5 mètres ($LJ_{max}$ pour $I_1 : G_1 \rightarrow G_3$). De même, la longueur de la jonction de sortie varie entre 11,5 mètres (la borne inférieure) et 20,5 mètres (LJ_{max} pour $X_3 : G_3 \rightarrow G_1$). La valeur de 11,5 mètres, qui est la plus commune pour la jonction de sortie, correspond à sa valeur théorique, à savoir :

$$(nbRecul + fen2 + nbIns2)\, d_{arp} + D_S \;=\; 3\, R_{long} + \left(\frac{R_{long}}{2\, d_{arp}} - 1\right) 2\, d_{arp}$$

Par ailleurs, la longueur des parois de jonctions fournies au détecteur de RLs est un facteur influençant la qualité du RL calculé. Ceci pourrait s'avérer un problème dans des situations comme avec le chemin $X_3 : G_2 \rightarrow G_3$, où la jonction d'entrée est très courte (5,5 mètres) dans plusieurs cas. En effet, la dispersion du nuage P_{CE} dans le graphique pour ce chemin, ainsi que la valeur élevée de σ_{LJ} (1,40 mètres), sont signes de l'instabilité potentielle du RL final calculé. Cependant, dans de tels cas, le détecteur de RLs peut rallonger la jonction d'entrée en demandant au collecteur de mesures des points de parois mesurés avant l'instant t_{NE}.

5.6.3 Conclusion

Nous sommes satisfaits de la performance du détecteur de transitions en vue des résultats expérimentaux que nous venons de voir. La conception de ce module, qui exploite l'information du télémètre frontal, la détection dynamique du RL et la notion de redondance temporelle, semble donc être viable.

Il est vrai que du point de vue de l'implantation, le bon fonctionnement des différentes analyses sensorielles à l'intérieur de ce module dépend d'un choix judicieux des paramètres utilisés : les fenêtres de temps pour les deux phases de l'analyse frontale et les paramètres pour l'analyse latérale. En particulier, les valeurs choisies pour $max\Delta pos$ et $max\Delta\theta$ ont une influence importante sur l'évaluation du critère de stabilité des RL_k et donc sur le séquencement des signaux de confirmation des

Tableau 5.17 : Statistiques de la longueur des jonctions d'entrée et de sortie

	Jonctions d'entrée [m]				Jonctions de sortie [m]			
chemin	LJ_{min}	LJ_{max}	$\overline{LJ}$	σ_{LJ}	LJ_{min}	LJ_{max}	$\overline{LJ}$	σ_{LJ}
$I_1 : G_1 \rightarrow G_2$	11,00	12,50	11,68	0,35	11,50	11,50	11,50	0,00
$I_1 : G_1 \rightarrow G_3$	11,00	13,50	11,92	0,52	11,50	14,00	11,59	0,44
$I_1 : G_2 \rightarrow G_3$	11,00	13,00	11,65	0,48	11,50	13,00	11,82	0,50
$I_7 : G_1 \rightarrow G_2$	9,50	10,00	9,85	0,23	11,50	11,50	11,50	0,00
$I_7 : G_3 \rightarrow G_2$	9,00	11,00	9,80	0,45	11,50	11,50	11,50	0.00
$I_{10} : G_1 \rightarrow G_2$	12,00	13,00	12,67	0,30	11,50	13,00	11,55	0,26
$I_{10} : G_2 \rightarrow G_3$	10,50	13,00	11,44	0,50	11,50	14,00	11,85	0,63
$X_2 : G_1 \rightarrow G_3$	7,00	9,50	8,67	0,60	11,50	11,50	11,50	0,00
$X_2 : G_1 \rightarrow G_4$	7,00	9,50	8,68	0,60	11,50	12,50	11,70	0,33
$X_2 : G_4 \rightarrow G_2$	8,00	10,50	9,50	0,56	11,50	11,50	11,50	0,00
$X_3 : G_1 \rightarrow G_3$	5,50	7,00	6,11	0,30	11,50	11,50	11,50	0,00
$X_3 : G_2 \rightarrow G_3$	5,50	11,00	7,47	1,40	11,50	11,50	11,50	0,00
$X_3 : G_3 \rightarrow G_1$	8,00	10,00	8,73	0,40	17,50	20,50	18,61	0,63
$X_3 : G_3 \rightarrow G_2$	8,00	10,00	8,73	0,40	11,50	19,00	12,35	2,29
globalement	5,50	13,50	9,78	1,91	11,50	20,50	12,14	1,93

transitions. En fait, les valeurs numériques de $max\Delta pos$ et $max\Delta\theta$ indiquées dans le tableau 5.16 correspondent aux seuils requis pour assurer un fonctionnement correcte de l'analyse latérale dans tous les cas de test. Les facteurs déterminant principalement cette contrainte, c'est-à-dire ayant une influence sur le calcul des RLs, sont d_{arp} et la largeur des galeries dans la mine (5 mètres en moyenne dans notre cas). Dans le cas général, il faut donc déterminer $max\Delta pos$ et $max\Delta\theta$ en fonction de d_{arp} et de la gamme de largeurs de galeries prévue dans l'environnement. Les autres paramètres (ceux exprimés en fonction de R_{long} dans le tableau 5.16) sont surtout dépendants de la valeur de d_{max}[13].

[13]Logiquement, les fenêtres de temps pour l'analyse frontale devraient être exprimées en fonction de d_{max} et non R_{long}. Nous avons employé R_{long} car celui-ci est du même ordre de grandeur que les valeurs voulues de ces paramètres.

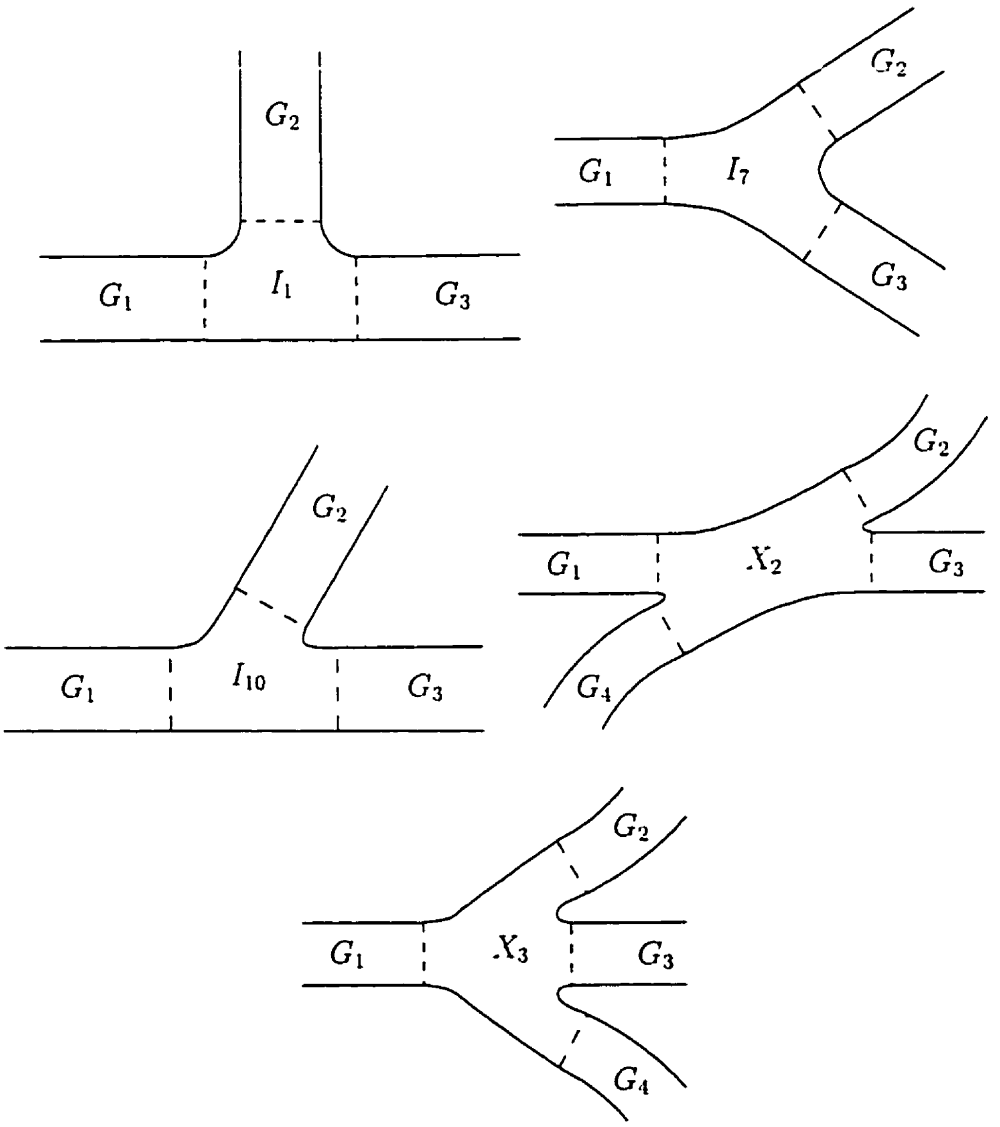
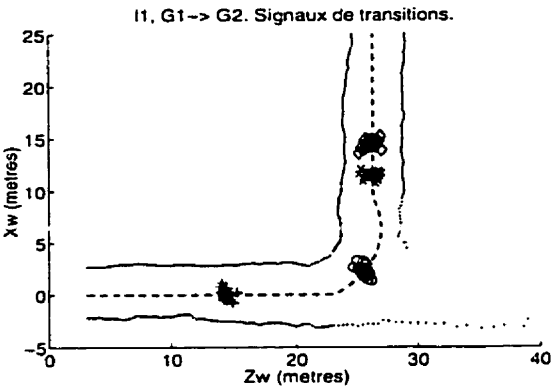
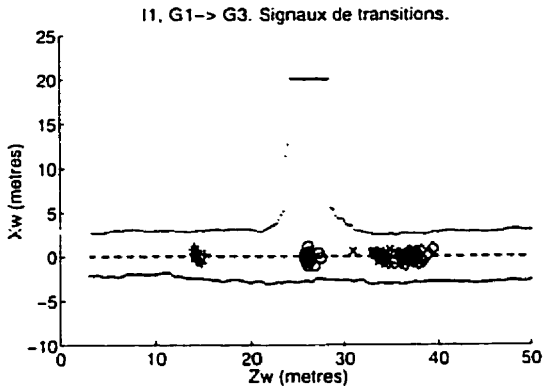


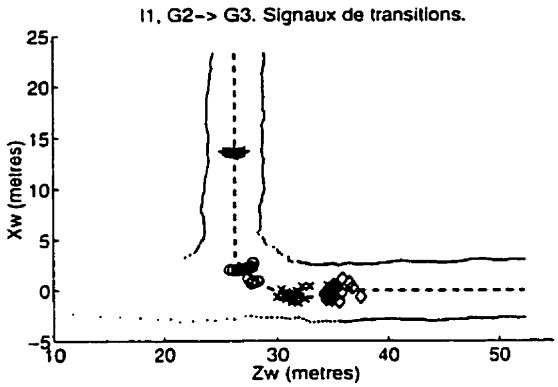
Figure 5.26 : Les cinq intersections d'essai



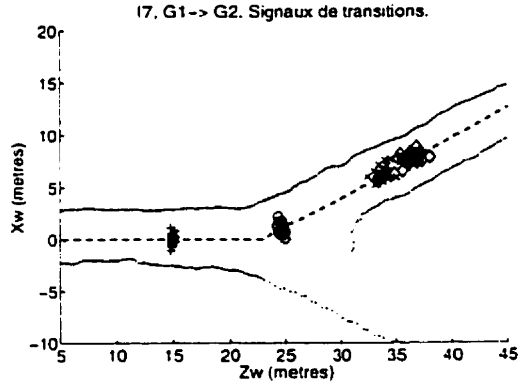
(a) chemin I_1 : $G_1 \rightarrow G_2$.



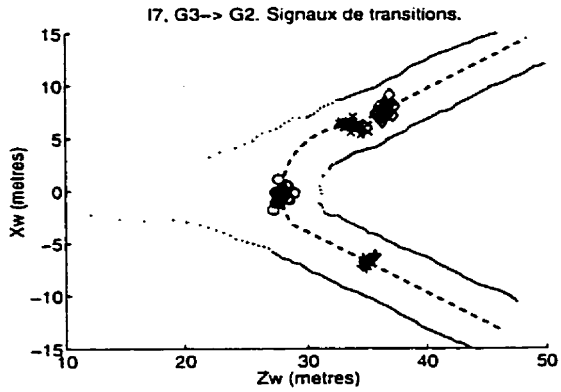
(b) chemin I_1 : $G_1 \rightarrow G_3$.



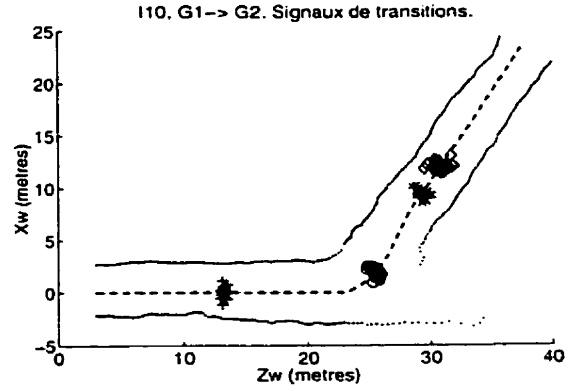
(c) chemin I_1 : $G_2 \rightarrow G_3$.



(d) chemin I_7 : $G_1 \rightarrow G_2$.

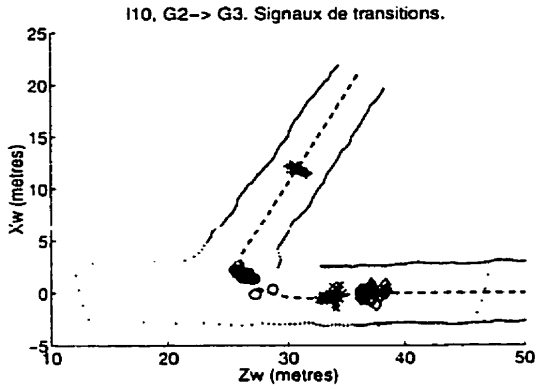


(e) chemin I_7 : $G_3 \rightarrow G_2$.

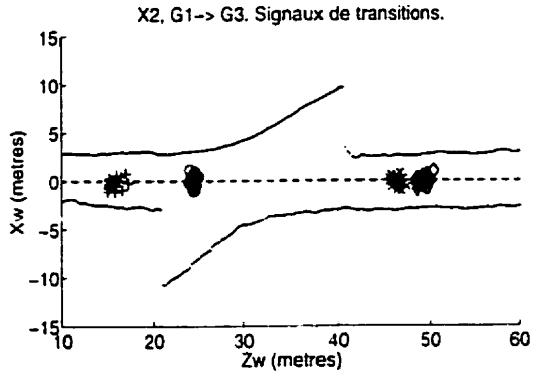


(f) chemin I_{10} : $G_1 \rightarrow G_2$.

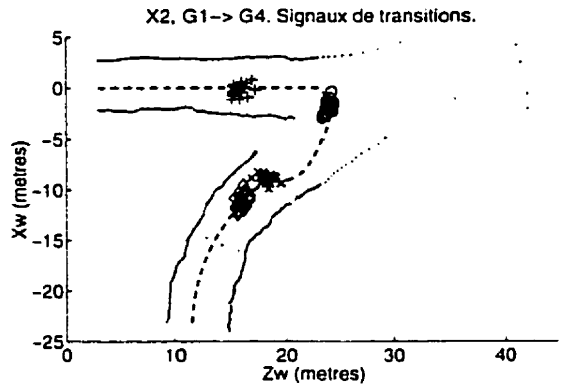
Figure 5.27 : Résultats pour les 14 traversées : poses trouvées



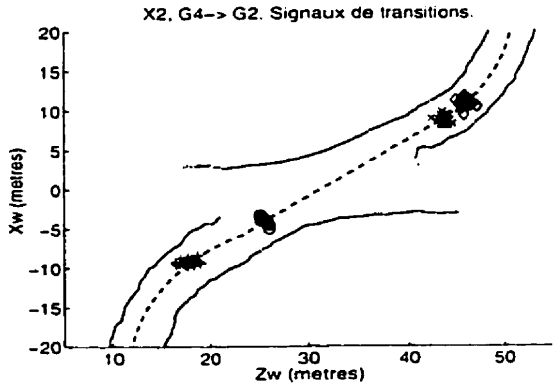
(a) chemin I_{10} : $G_2 \rightarrow G_3$.



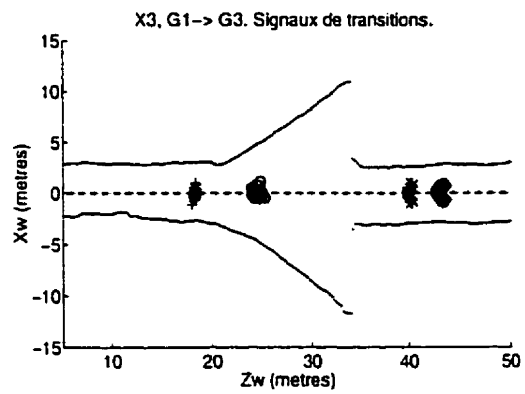
(b) chemin X_2 : $G_1 \rightarrow G_3$.



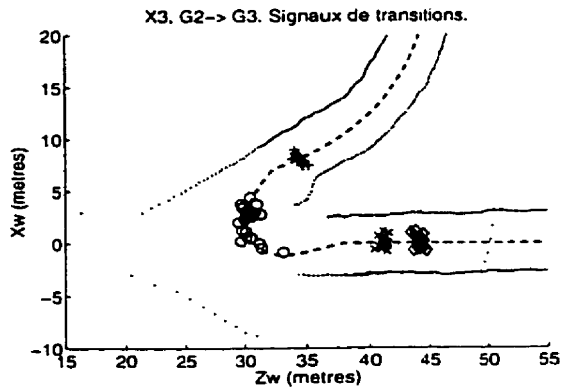
(c) chemin X_2 : $G_1 \rightarrow G_4$.



(d) chemin X_2 : $G_4 \rightarrow G_2$.

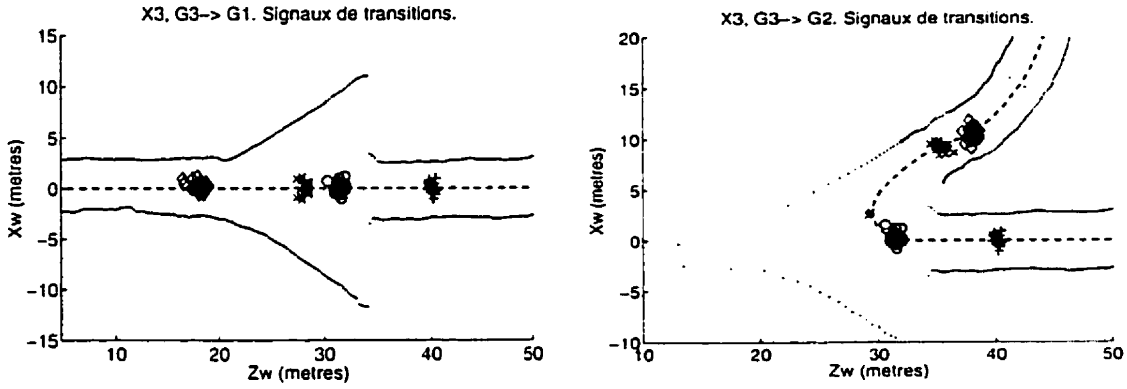


(e) chemin X_3 : $G_1 \rightarrow G_3$.



(f) chemin X_3 : $G_2 \rightarrow G_3$.

Figure 5.28 : Résultats pour les 14 traversées : poses trouvées (suite)



(a) chemin X_3 : $G_3 \rightarrow G_1$.

(b) chemin X_3 : $G_3 \rightarrow G_2$.

Figure 5.29 : Résultats pour les 14 traversées : poses trouvées (fin)

Chapitre 6

Conclusion

Dans cet ouvrage, nous avons présenté l'architecture d'un système d'arpentage automatisé destiné aux mines souterraines. Cette architecture identifie toutes les capacités perceptuelles requises par une plate-forme mobile autonome qui puisse à la fois se déplacer, se localiser et récolter des mesures d'arpentage dans la mine. Dans le cadre de travail ainsi posé, nous avons développé les aspects de la collecte des mesures d'arpentage, d'une part, et de la navigation globale, c'est-à-dire la localisation du robot par rapport au graphe topologique *a priori*, d'autre part.

Au sein de l'activité d'arpentage, nous avons proposé une méthode pour analyser la structure naturelle des parois autour du robot afin de déterminer des points de repères géométriquement stables, les référentiels locaux (RLs), qui sont associés aux sections de parois décrivant les transitions entre éléments topologiques de la mine, c'est-à-dire aux jonctions entre galeries et intersections. Une série de tests dans plusieurs cas de traversée d'intersection, en faisant varier pour chaque cas la trajectoire précise suivie, démontrent que le RL est une caractéristique significative de la structure de l'environnement aux transitions entre éléments topologiques.

Au sein de l'activité de navigation globale, nous avons développé le module de détection des transitions topologiques. Ce module alterne entre l'analyse des mesures frontales et celle des mesures latérales afin de notifier puis confirmer aux autres mo-

dules les changements d'éléments topologiques. L'analyse des mesures frontales est un processus dynamique qui exploite la représentation des profils sous forme d'image qui évolue avec le mouvement du robot. Ce processus cherche à reconnaître plusieurs événements perceptuels qui se produisent dans l'image binaire des profils frontaux. L'analyse des mesures latérales réutilise la notion de RL de manière dynamique afin de localiser dans la structure du milieu parcouru les changements importants correspondant aux événements reconnus par l'analyse frontale. Nos tests sur la détection des transitions démontrent la validité des événements perceptuels que nous avons définis et montrent que ces événements sont rattachés à la structure de l'environnement et sont relativement indépendants de la trajectoire précise suivie et des paramètres des senseurs.

Notre travail apporte plusieurs contributions innovatrices au domaine de la navigation autonome dans des milieux tels que les mines souterraines, formés de corridors partiellement structurés. Tout d'abord, l'architecture de notre système a ceci d'original qu'elle est orientée vers la tâche de l'arpentage des parois tout en séparant cette fonction particulière de celles strictement requises pour la navigation autonome. La décomposition de la tâche de navigation autonome en des modules de localisation globale et de contrôle local du robot est un principe que nous retrouvons dans d'autres architectures destinées aux mines souterraines, comme celles de Banta et al. (1992), Steele et al. (1991) et Hurteau et al. (1992). D'ailleurs, cette conception modulaire et hiérarchisée (par rapport au milieu) des problèmes concourants du déplacement et de la localisation d'une plate-forme autonome caractérise une grande partie des travaux en robotique autonome. Cependant, l'ensemble des problèmes spécifiques identifiés au niveau fonctionnel différencie notre architecture de ces autres travaux.

Notre approche se distingue aussi par sa modélisation de l'environnement ; le modèle se base sur les relations entre la forme des parois, la tâche de navigation et les capacités sensorielles du robot, pour diviser la mine souterraine en sous-régions qualitativement distinctes, à savoir les galeries, intersections et chambres. Remarquons

que le milieu minier souterrain se prête naturellement à une décomposition de cette forme ; de fait, d'autres travaux raisonnent aussi sur l'environnement minier à un niveau plus élevé que seulement la géométrie des parois. Saint-Amant et al. (1988), Kitchener (1986), Steele et al. (1991) et Banta et al. (1992) divisent tous en sections la région de la mine à parcourir, chaque section étant uniforme du point de vue du type de contrôle à appliquer au véhicule. Selon les tâches à accomplir (autres que le déplacement du véhicule) et les modes de contrôle envisagés, cette subdivision peut se baser soit sur des catégories de sous-régions comme dans notre cas (voies de transport, intersections, zones de chargement/déchargement), soit sur d'autres critères de décomposition, ou encore peut exploiter des structures artificielles présentes dans la mine. Mais aucun de ces systèmes ne fait intervenir une représentation explicite, topologique du milieu, ni des règles systématiques pour la décomposition de celui-ci ; leur décomposition est exprimée en termes de déplacements du véhicule (*e.g.* trajet global divisé en segments) plutôt qu'en termes de la structure de l'environnement.

Dans notre cas par contre, la correspondance explicite entre la mine et sa représentation topologique met à notre disposition plusieurs outils pour la résolution des problèmes de départ (navigation et arpentage) qui ne sont pas employés dans les autres travaux mentionnés ci-dessus. Nous avons souligné dès l'introduction l'importance centrale du graphe topologique *a priori*, duquel découlent le chemin global, la stratégie de localisation globale basée sur la détection des transitions et la conception du contrôleur en branchement. En quelque sorte, notre modèle de l'environnement permet de faire le pont entre l'exploration de milieux représentés comme des graphes et les résultats dans ce domaine, d'une part, et la navigation autonome dans les mines souterraines, d'autre part. Aussi, le processus de création de cartes métriques locales dans notre système suivi de la modélisation géométrique des éléments topologiques par Labrosse (1998) est une approche assez différente des techniques classiques d'arpentage et de cartographie des mines.

Par son développement des capacités sensorielles du "robot-arpenteur", ce projet

introduit des idées originales quant à l'analyse des données télémétriques pour aider un robot à se localiser dans un environnement de corridors partiellement structurés. Le concept du RL est certainement nouveau par rapport à ce qui existe dans la littérature. En effet, les autres travaux en milieu minier souterrain que nous avons vus déterminent des points de repère pour la localisation à partir de balayages télémétriques prises à des positions fixes du robot ; à chaque observation, les structures saillantes recherchées (points, segments, coins) sont détectées sur des sous-ensembles de la vue statique des parois (et dans certains cas des structures artificielles) autour du robot. Les RLs, eux, ne sont pas rattachés à des morceaux isolés de parois observés statiquement, mais plutôt aux changements importants dans la structure locale observée par le robot en mouvement. La méthode de calcul du RL assure autant que possible sa stabilité face aux variations de la trajectoire suivie, à l'apparition d'occlusions et à l'irrégularité à une échelle fine de la forme des parois. Grâce à cela, la notion du RL trouve plusieurs applications dans notre système : comme référentiel cartésien pour les cartes métriques locales, comme point de repère exploité par une approche originale (proposée mais non-implantée) pour la relocalisation métrique et comme mesure donnant lieu à un événement perceptuel indiquant que le robot traverse une transition topologique.

L'image binaire des profils frontaux est elle aussi une façon originale de représenter l'environnement local perçu à l'aide du principe télémétrique. C'est une représentation riche mais compacte de l'évolution, le long du parcours, du milieu perçu dans un plan horizontal devant le robot. Elle incorpore un traitement préalable des profils pour la détection des branches et montre implicitement la topologie, et jusqu'à un certain point la géométrie, de l'environnement qui se présente au robot. Même sans la détection préalable des branches, l'idée de laisser accumuler des vues instantanées captées au cours du temps, pour ensuite étudier ces mesures dans leur ensemble plutôt qu'individuellement, est un moyen intéressant d'introduire le principe de redondance temporelle dans le traitement sensoriel. Dans ce projet, l'introduction de

l'image binaire a servi à définir des événements perceptuels pour reconnaître l'arrivée de transitions topologiques devant le robot. Mais cette représentation des profils télémétriques pourrait avoir d'autres applications, par exemple comme information sensorielle pour guider le véhicule dans un algorithme de contrôle local. Nous avons d'ailleurs regardé ce sujet de plus près et proposé quelques critères pour la conception d'un tel système de navigation locale dans Debanné et al. (1997).

Le système d'arpentage automatisé fait plusieurs suppositions sur l'environnement à parcourir qui limitent les zones dans une vraie mine où notre système peut être employé. Au départ, le modèle de mine souterraine, composé de galeries, intersections et chambres, bien qu'il représente correctement les types d'excavations les plus importantes à un niveau souterrain donné dans une mine, constitue néanmoins une simplification du cas réel. En effet, nous ne tenons pas compte de la présence possible de structures locales divergeant du modèle de base, à savoir les obstacles au sol (*e.g.* un éboulement de roches), les renflements dans les galeries qui permettent les croisements de véhicules et les éléments d'infrastructure visibles au robot. Par ailleurs, il a déjà été signalé que notre système n'est pas conçu pour arpenter des zones à excavations denses telles que les zones de chambres et piliers. Cette restriction est quantifiée dans l'implantation du détecteur de transitions : la longueur minimale des galeries pour assurer le bon fonctionnement de ce module, qui est de l'ordre de la portée du senseur frontal. Il y a aussi la question des chambres dans notre modèle, qui désignent des espaces ouverts ayant comme fonctions le rangement des équipements ou l'extraction et le chargement/déchargement du minerai. L'implantation du détecteur de RLs et du détecteur de transitions ne traitent pas le cas des chambres comme distinct du cas des intersections. Mais étant donné les différences de géométrie entre ces deux types d'excavations, nous ne pouvons pas garantir que nos algorithmes aient un comportement correcte dans des chambres de forme quelconque. Enfin, il a été sous-entendu dans tout cet ouvrage que l'arpentage est limité à un seul niveau souterrain dans la mine, puisqu'il n'est pas prévu que le véhicule circule entre les différents

niveaux. Malgré toutes ces limitations, nous considérons que ce projet démontre sur le plan conceptuel la faisabilité de l'arpentage dans une mine souterraine par un robot mobile.

Par souci de limiter l'envergure de ce projet, nous avons été obligés de faire plusieurs simplifications par rapport au sujet de recherche dans sa totalité et d'abréger le développement de plusieurs parties importantes de notre architecture. Par conséquent, notre travail conceptuel peut être poursuivi selon des voies de recherche intéressantes. En premier lieu, plusieurs questions se posent au sujet de la modélisation topologique de la mine. Nous avons supposé un graphe topologique *a priori* qui décrit de manière complète et sans erreurs l'environnement à parcourir ; en même temps, nous avons présenté des définitions structurelles et fonctionnelles des éléments topologiques et avons tenté d'établir des règles de décomposition topologique de la mine. Mais vu la géométrie très variable et irrégulière du milieu et la nature heuristique de ces règles, il faut s'attendre dans le cas général à ce que le graphe topologique d'une mine contiennent des erreurs de modélisation[1]. Nous avons de plus supposé que notre système identifie toujours correctement la topologie de l'environnement. Là encore le cas général risque d'être plus compliqué : le bruit sensoriel et la géométrie d'intersection peuvent tous les deux contribuer à la mauvaise détection des branches à une intersection. Une première voie de recherche consiste donc à développer des stratégies de replanification de chemin et de correction d'erreurs dans le graphe et/ou dans les raisonnements du robot, en présence d'incertitude sur la modélisation et/ou la perception topologique du milieu. Si de plus nous supprimons le graphe topologique initial, alors le problème prend une nouvelle dimension, celle de l'exploration d'un milieu complètement inconnu. C'est maintenant au robot de construire lui-même la carte topologique à mesure qu'il se déplace, à l'aide de règles d'interprétation perceptuelle semblables à celles que nous avons établies. Il s'agit alors d'un contexte de

[1] ou, autrement dit, que la mine ne soit pas perçue en toutes parts par le robot de manière identique au processus de modélisation topologique.

recherche similaire aux travaux de Kuipers et Byun (1987) et Engelson et McDermott (1992), ainsi que d'autres travaux d'exploration employant une carte topologique du milieu, mais dans un type de milieu et avec une configuration de senseurs particuliers à notre problème, d'où l'opportunité d'innover dans ce domaine.

En second lieu, il y a la nécessité d'introduire des modèles plus réalistes du bruit de mesure associé aux capteurs à bord de notre robot, à savoir le senseur odométrique et les senseurs télémétriques, en particulier le télémètre frontal car celui-ci détecte des points plus éloignés que les senseurs latéraux. Pour ce qui est de l'odométrie, nous avons initialement présenté la question de la dégradation de la trajectoire mesurée dans le cas réel, pour ensuite faire abstraction de ce problème afin de simplifier l'élaboration de notre approche. Il faut donc étudier le comportement de notre système dans le cas où l'odométrie est entachée d'erreurs. L'amplitude des erreurs à considérer varie selon que le module en question exploite l'intégration de trajectoire à une échelle locale ou globale. À l'échelle locale, c'est le module de détection des RLs qui intervient. Il faut regarder à la fois les effets du déplacement longitudinal du véhicule et de ses variations d'orientation (*i.e.* la déviation de la trajectoire idéale dans nos tests) sur la trajectoire mesurée à l'échelle de la longueur typique d'une jonction (10 à 12 mètres d'après nos tests). La dégradation de la trajectoire mesurée entraîne des déformations longitudinales et latérales de la forme des parois de jonction mesurées; ces déformations affectent à leur tour le calcul du RL. Il s'agit donc d'étudier le comportement de la détection des RLs en présence d'incertitude odométrique ainsi que les conséquences d'une diminution éventuelle de la stabilité des RLs sur le reste du système.

À une échelle plus globale (donc avec des erreurs plus importantes commises sur l'intégration de trajectoire), il y a d'abord à considérer le module de relocalisation métrique. Nous préconisons comme projet éventuel d'implanter l'approche qui a été proposée dans cet ouvrage et d'évaluer son fonctionnement sous les effets conjugués de l'erreur sur l'estimation initiale de la pose relative (basée sur les RLs) et de la dé-

formation non-rigide de la carte métrique à laquelle les nouvelles mesures doivent être fusionnées. Des modifications devraient être apportées à notre méthode de relocalisation dans le cas où sa performance (c'est-à-dire la convergence de la pose relative vers une solution acceptable) s'avérait inadéquate. Un travail complémentaire à envisager porte sur la déformation de la trajectoire mesurée, et donc des mesures d'arpentage, à l'échelle de plusieurs éléments topologiques. Nombre d'approches existantes pour la correction d'erreurs odométriques réparties sur le trajet d'un robot mobile (*e.g.* Brooks, 1985; Lu et Milios, 1995) peuvent être appliquées à la tâche de corriger les déformations non-rigides des cartes métriques locales et la non-fermeture géométrique des boucles dans la topologie de la mine (phénomènes déjà exposés auparavant).

Quant au senseur télémétrique frontal, nous avons développé l'analyse des profils frontaux pour la notification des transitions en l'absence de bruit de capture. Cependant, comme nous l'avons déjà mentionné, le bruit de capture dans les profils télémétriques fournis par un senseur réel affectera forcément le processus de détection des branches et donc la qualité des images binaires produites. En fait, nous avons examiné cette question lors du développement du détecteur de transitions, en mettant nos efforts sur l'élaboration d'un modèle réaliste de l'erreur de mesure à rajouter aux points de profils calculés (sans erreur) dans la mine virtuelle. La première composante de cette erreur est le bruit sur la distance de chaque point (*cf.* Blais et al., 1988; Lang et al., 1993, pour l'étalonnage de la caméra BIRIS). La deuxième composante est reliée à l'affaiblissement du signal laser de retour en fonction de l'orientation locale de la paroi ; ceci est modélisé par un seuil sur la puissance du signal en dessous duquel le point de profil est considéré comme non-détecté. C'est surtout la deuxième composante de l'erreur qui affecte notre système, car elle crée des "trous" dans le profil qui sont traités comme des branches (LVs). Le résultat pour l'image binaire est que des "fausses traces" viennent se rajouter à la vraie trace des profils, leur nombre et leurs dimensions augmentant avec la valeur fixée du seuil. De cette manière, il devient plus difficile d'extraire de l'image binaire les événements perceptuels significatifs. Comme

nous n'avons pas achevé cette étude, elle n'a pas été documentée formellement dans cet ouvrage. Mais il serait bien sûr profitable de poursuivre ce sujet, notamment en améliorant l'analyse frontale pour la rendre robuste aux perturbations de l'image binaire.

Un dernier axe de recherche que nous jugeons nécessaire pour compléter la conception de notre système est celui de la navigation locale du robot. Il faut prévoir dans le déplacement autonome du véhicule deux modes principaux de contrôle, l'un pour les galeries et l'autre pour les régions ouvertes dans la mine (*i.e.* les intersections et les chambres dans notre modèle). C'est d'ailleurs pourquoi l'activité de navigation locale dans notre architecture est divisée en deux modules, le contrôleur de suivi de galerie et le contrôleur de branchement. Pour réaliser le suivi de galerie, nous envisageons l'approche développée par Juneau et al. (1993) car elle est prévue spécifiquement pour les galeries souterraines et pourrait donc être implantée directement au sein de notre système. Dans cette optique, il faudrait alors établir un deuxième mode de contrôle pour effectuer les branchements. Il existe toutefois dans la littérature de nombreuses approches pour le contrôle des déplacements d'un robot basé sur l'information sensorielle, plusieurs d'entre elles ayant l'avantage d'être suffisamment générales pour se prêter à la fois à la navigation dans les galeries et au contrôle en branchement. Nous pensons entre autres au contrôle réactif (Soldo, 1990; Payton, 1986; Arkin, 1990), à la méthode des champs de potentiel (Krogh et Thorpe, 1986; Krogh, 1984) et à d'autres approches (*e.g.* Kamon, Rivlin et Rimon, 1996; Lumelsky et Stepanov, 1987). Bien entendu, l'application d'une de ces approches implique un travail supplémentaire pour l'adapter au type de milieu et à la configuration sensorielle particuliers à notre contexte.

Ainsi, nous avons identifié trois domaines, la modélisation topologique de la mine, les effets du bruit sensoriel et le contrôle local du véhicule, qui constituent tous trois des voies de recherche fort intéressantes et qui forment, avec le développement présenté dans cet ouvrage, la conception complète d'un système automatisé d'arpentage et

de navigation autonome destiné aux mines souterraines. Quoique notre travail soit limité par rapport au sujet de réaliser complètement un tel système (sujet qui, nous tenons à le rappeler, est très vaste, posant de nombreux problèmes conceptuels et techniques), nous croyons néanmoins que le présent travail constitue un pas important vers la réalisation physique d'un tel système et qu'il contribue des idées valables aux domaines de l'automatisation minière et de la robotique mobile.

Annexe A

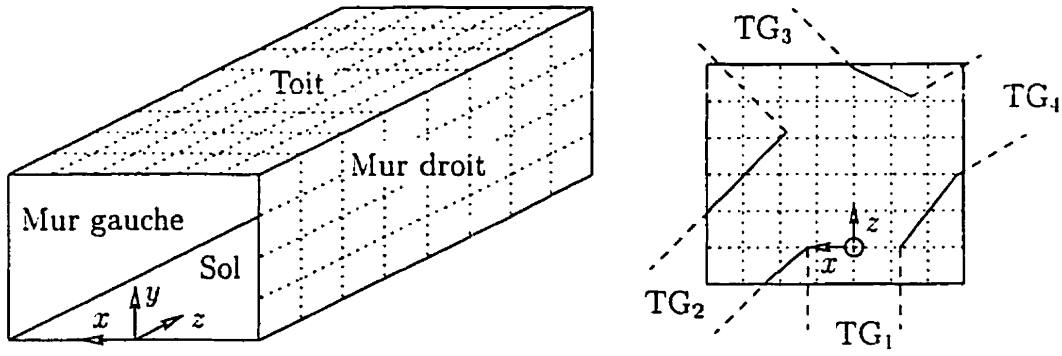
Banc expérimental de navigation en mine virtuelle

A.1 Génération de la mine virtuelle

Nous appelons *mine virtuelle* la version infographique d'une mine souterraine, conçue de manière à reproduire des données télémétriques et visuelles réalistes lorsqu'un véhicule (virtuel lui aussi) s'y déplace. Par rapport au modèle GNOMine, la mine virtuelle réunit les attributs de la géométrie des parois, vu précédemment, et de la *rugosité* des parois. Ce deuxième attribut sert à caractériser différents types de relief associés aux surfaces rocheuses dans une mine. Cet attribut est représenté par trois paramètres : la densité des aspérités sur la paroi, la dimension fractale et la déviation moyenne des aspérités.

Chaque objet géométrique de la mine virtuelle est un ensemble de surfaces dans l'espace tridimensionnel, généré dans son propre référentiel à partir des paramètres numériques associé à cet objet dans le modèle. Pour générer un tronçon de galerie, nous partons d'une grille plane de points, de longueur égale à celle de l'axe du tronçon et de largeur le double de la somme de la hauteur et la largeur du tronçon. La grille est pliée le long de quatre arêtes pour obtenir les quatre parois d'un tube de section

rectangulaire (figure A.1(a)). Ensuite, nous dévions le tube droit longitudinalement pour qu'il suive l'équation de l'axe du tronçon.



(a) Tronçon de galerie.

(b) Tronçon de galerie.

Figure A.1 : Création des objets géométriques de la mine

Pour une intersection, il faut créer les murs, le toit et le sol. Un mur d'intersection est une surface reliant à leurs extrémités les parois verticales de deux galeries. Nous créons une grille plane de points, dont les coordonnées sont recalculées pour suivre la courbe 2D au sol du mur (spécifiée dans le modèle de la mine). Celle-ci est une spline cubique tangente en ses deux extrémités aux axes des galeries.

Le toit (respectivement sol) est une surface dont le contour est donné par les arêtes supérieures (respectivement inférieures) des galeries qui se raccordent à l'intersection et des murs entre les galeries. D'abord, nous déterminons les dimensions d'une grille plane parallèle au plan xz qui recouvre l'intersection (figure A.1(b)). Ensuite, le contour de la surface du toit (sol) est déterminé en transformant tous les points d'arêtes supérieures (inférieures) des galeries et murs vers le référentiel de l'intersection. La région de la grille à l'intérieur du contour est déterminée à l'aide d'une algorithme de croissance de région.

Nous utilisons un algorithme de génération de fractales, présenté dans Peitgen et Saupe (1988), pour obtenir le relief des surfaces. L'algorithme est contrôlé par les trois

paramètres de l'attribut rugosité. Premièrement, une surface donnée est discrétisée avec une résolution spatiale de points correspondant à la valeur de densité des aspérités. La dimension fractale influence le calcul d'un patron fractal bi-dimensionnel qui déformera la surface initiale. L'amplitude moyenne de la déformation de la surface initiale est contrôlée par le paramètre de déviation moyenne, en normalisant le patron fractal par celui-ci avant de l'appliquer aux points.

Plutôt que de modifier la position des points de la surface discrétisée, nous pourrions produire l'apparence de relief en appliquant le patron fractal à un attribut visuel de la surface, par exemple le niveau d'intensité de gris. Cependant, l'information sensorielle qu'un robot doit extraire de l'environnement peut être structurale (*e.g.* profils télémétriques) autant que visuelle (*e.g.* images de caméra). La rugosité d'une surface doit donc se manifester par un véritable relief et non pas uniquement par une texture.

Lorsque tous les objets de la mine (tronçons de galeries, murs, toits et sols d'intersections et chambres) ont étés générés, il faut les disposer dans l'espace afin de réaliser la mine virtuelle. À ce niveau intervient une librairie infographique qui définit des classes d'objets pour les formes 3D ainsi que d'autres classes d'objets pour le rendu graphique (caméras, illumination, transformations géométriques, etc.). Une base de donnée, appelée *arbre de scène*, est construite à partir de ces objets. Pour disposer correctement les objets de la scène, on insère dans l'arbre de scène des transformations géométriques qui proviennent de deux sources :

- la translation et rotation engendrées entre les deux bouts d'un tronçon de galerie ;
- la translation et rotation entre deux extrémités de galeries dans une intersection (ou chambre).

La position de ces transformations dans l'arbre de scène correspond aux endroits de raccordements entre objets géométriques.

A.2 Déplacement du véhicule

Dans l'environnement simulé le robot est représenté par une caméra, un phare et un ensemble de télémètres, tous rigidement liés au repère local du véhicule. La position et l'orientation (la pose) du véhicule dans la mine sont déterminées par une transformation géométrique entre le repère global de la scène et celui rattaché au point M_r du robot (figure A.2). La valeur instantanée de cette transformation pourra

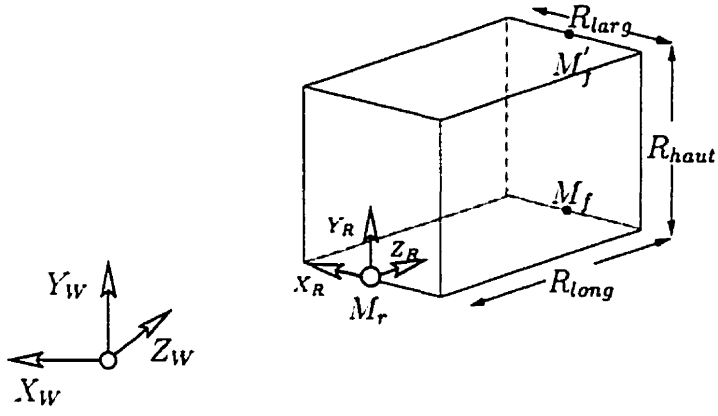


Figure A.2 : Déplacement du robot dans la mine virtuelle

provenir d'un module de contrôle local du robot qui tiendra compte des modèles cinématiques et dynamiques d'un vrai véhicule.

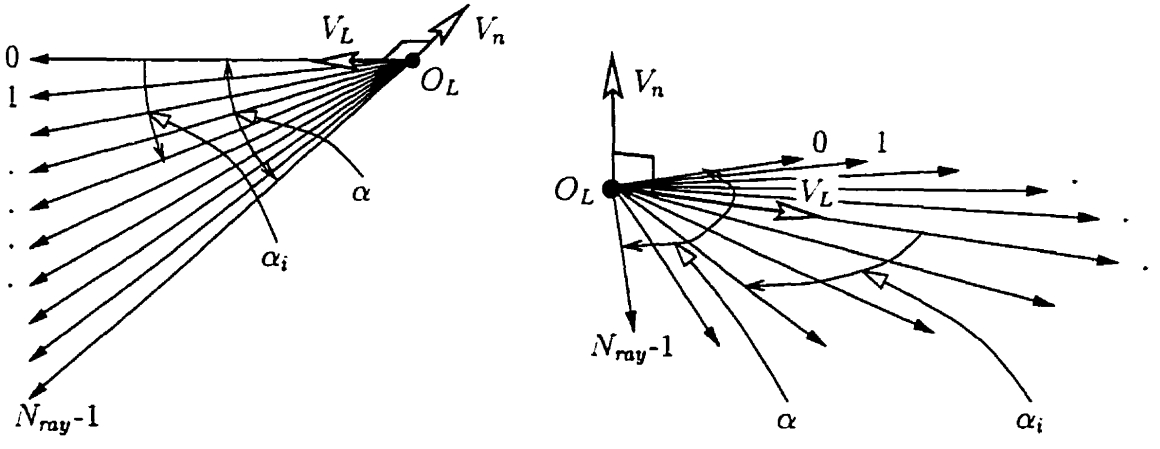
En l'absence de capacités de navigation locale du robot, la trajectoire sera calculée au préalable et spécifiée comme une série de valeurs de pose globale du robot. Le chemin suivi par le robot est idéalisé : dans les tronçons de galerie, il suit exactement l'axe du tronçon ; dans les intersections, le chemin est une courbe en spline cubique joignant les extrémités de deux axes de galerie. Sur chaque morceau du chemin, nous prenons des points à intervalle constante d_{arp} sur la courbe en fonction de la vitesse linéaire V_R du robot (supposée constante) et de l'intervalle de mesure des données télémétriques Δt :

$$d_{arp} = V_R \cdot \Delta t$$

Pour simuler le contrôle de véhicule pour le suivi de chemin, nous pouvons introduire des déviations de la trajectoire par rapport aux chemin idéal. Dans les essais expérimentaux des chapitres 4 et 5, les trajectoires idéales sont déviées par une sinusoïde d'amplitude, de période et de délai de phase variables.

A.3 Calcul des profils télémétriques

Les senseurs télémétriques du robot virtuel sont en fait une version simplifiée du principe de télémètre à lumière structurée. A partir d'un point O_L rigidement lié au véhicule, nous projetons une série de rayons coplanaires à intervalles angulaires uniformes (figure A.3).



(a) Projection des plans laser latéraux. (b) Projection des plans laser frontal et du toit.

Figure A.3 : Projection des rayons laser

Chaque rayon induit une demi-droite dans l'espace, obtenue par rotation du vecteur de direction principale $\vec{V_L}$ par un angle α_i, $i = 0 \ldots N_{ray}$autour de $\vec{V_n}$. Le nombre de rayons projetés, N_{ray}, et l'écartement angulaire α du plan laser sont des paramètres variables de la fonction de télémétrie.

L'intersection des demi-droites de projection avec l'environnement immédiat du robot est un ensemble de points dans l'espace qui forment le *profil* télémétrique. Initialement, chaque point était calculé au moyen d'une fonction de la librairie infographique qui recherche dans toutes les surfaces de la scène le point d'intersection le plus proche le long de la demi-droite. Cependant, cette approche s'est avérée trop lente à cause du nombre de fois que le tri des surfaces devait être effectuée pour obtenir le profil. La méthode actuelle consiste à calculer une seule fois l'intersection de *tout le plan laser* avec la scène et d'en extraire les points de profil dans les directions des rayons. De plus, nous réduisons le temps de recherche en ne considérant que les objets de la mine virtuelle qui intersectent une "boîte de visibilité" autour du véhicule. Ces deux modifications ont accéléré le calcul des profils de manière considérable.

Du fait qu'un télémètre réel est limité en portée, nous imposons aussi une limite de portée à nos senseurs simulés. Soit d_{max} la distance maximale de détection. Pour un plan laser donné, nous évaluons la distance entre chaque point de profil P_{profil} et l'origine de projection. Si :

$$dist(O_L, P_{profil}) > d_{max} \tag{A.1}$$

alors ce point sera rejeté, c'est-à-dire considéré comme non-détecté.

A.4 Interface de visualisation

Il s'est avéré utile de développer une interface graphique pour visualiser le processus de capture de données sensorielles du robot qui se déplace dans la mine. L'outil nous procure une vision instantanée de l'évolution du robot à chaque position dans sa trajectoire. Le logiciel utilise les ressources de la librairie infographique. Le schéma de la figure A.4 illustre l'organisation de notre logiciel.

Le logiciel gère en parallèle deux scènes : l'une pour visualiser la mine du point de vue du robot et l'autre pour avoir une vue globale de la mine et du robot qui

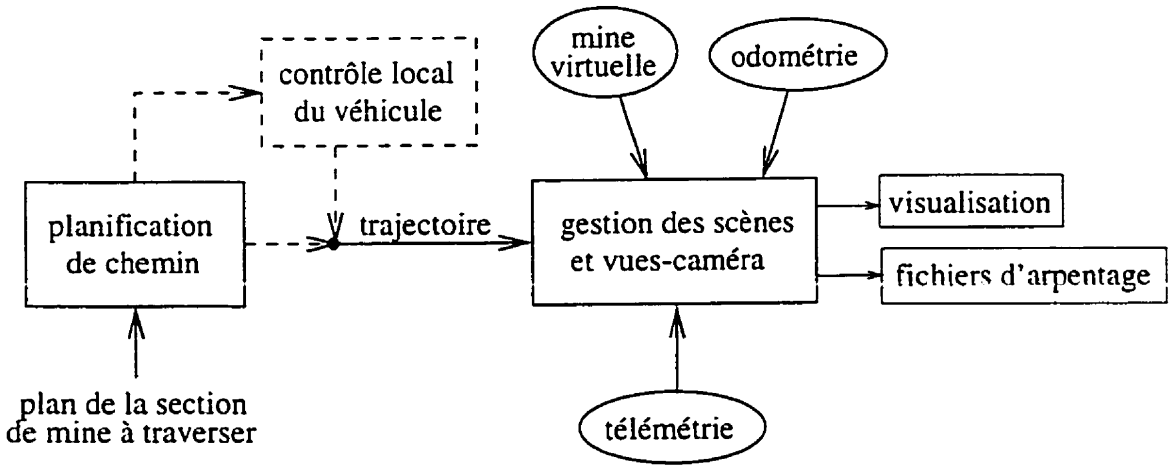


Figure A.4 : Logiciel de gestion de la navigation en mine virtuelle

s'y déplace. Les deux scènes partagent une matrice de transformation qui prend successivement les valeurs de la trajectoire suivie. De plus, l'interface comporte deux fenêtres d'affichage des profils télémétriques. Chacune de ces fenêtres contient un graphique où sont tracés les profils dans des coordonnées locales au véhicule. Un profil est représenté par une courbe constituée de segments reliant deux à deux les points consécutifs du profil. Dans l'exemple donné à la figure A.4, les profils affichés correspondent à la configuration de senseurs choisie pour l'exploration/arpentage: en haut, trois profils latéraux et en bas, le profil dans un demi-plan horizontal devant le véhicule.

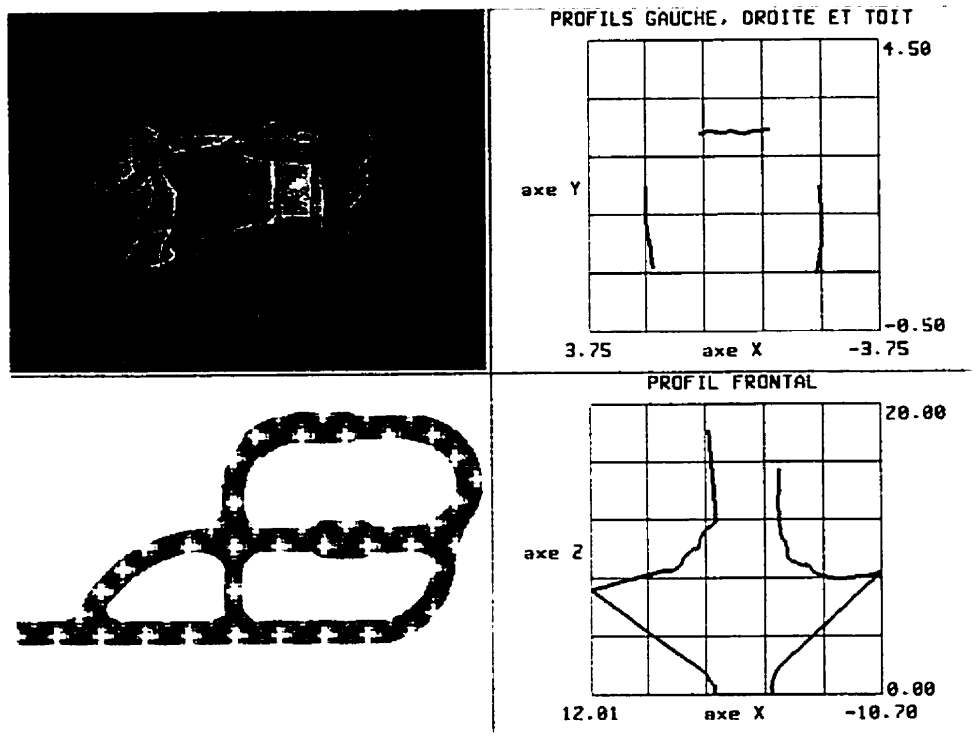


Figure A.5 : Visualisation de la navigation en mine virtuelle

Annexe B

Simulateur de l'arpentage d'une mine souterraine

B.1 Présentation générale

Dans le corps de l'ouvrage nous avons présenté *séparément* les activités principales de la navigation globale et de l'arpentage. Dans ce contexte, nous avons implanté et testé deux modules particuliers du système, le détecteur de RLs et le détecteur de transitions. Cependant, nous avons mené le travail de développement une étape plus loin : une simulation du système d'arpentage automatisé *dans son ensemble*, faisant appel à l'intégration des modules individuels déjà implantés.

Par rapport au système complet décrit à la section 3.4, la simulation réalise l'interaction entre un sous-ensemble des modules tel que le montre le schéma de la figure B.1. Remarquons premièrement que l'activité de la navigation locale est remplacée par l'interface de navigation en mine virtuelle décrite à la section A.4, qui elle fait intervenir une trajectoire pré-calculée du robot. Cette interface fournit au reste du système des fichiers de mesures télémétriques et odométriques. Par ailleurs, le module relocalisateur métrique est exclus de l'activité d'arpentage car nous n'introduisons pas de bruit de mesure dans l'odométrie du véhicule.

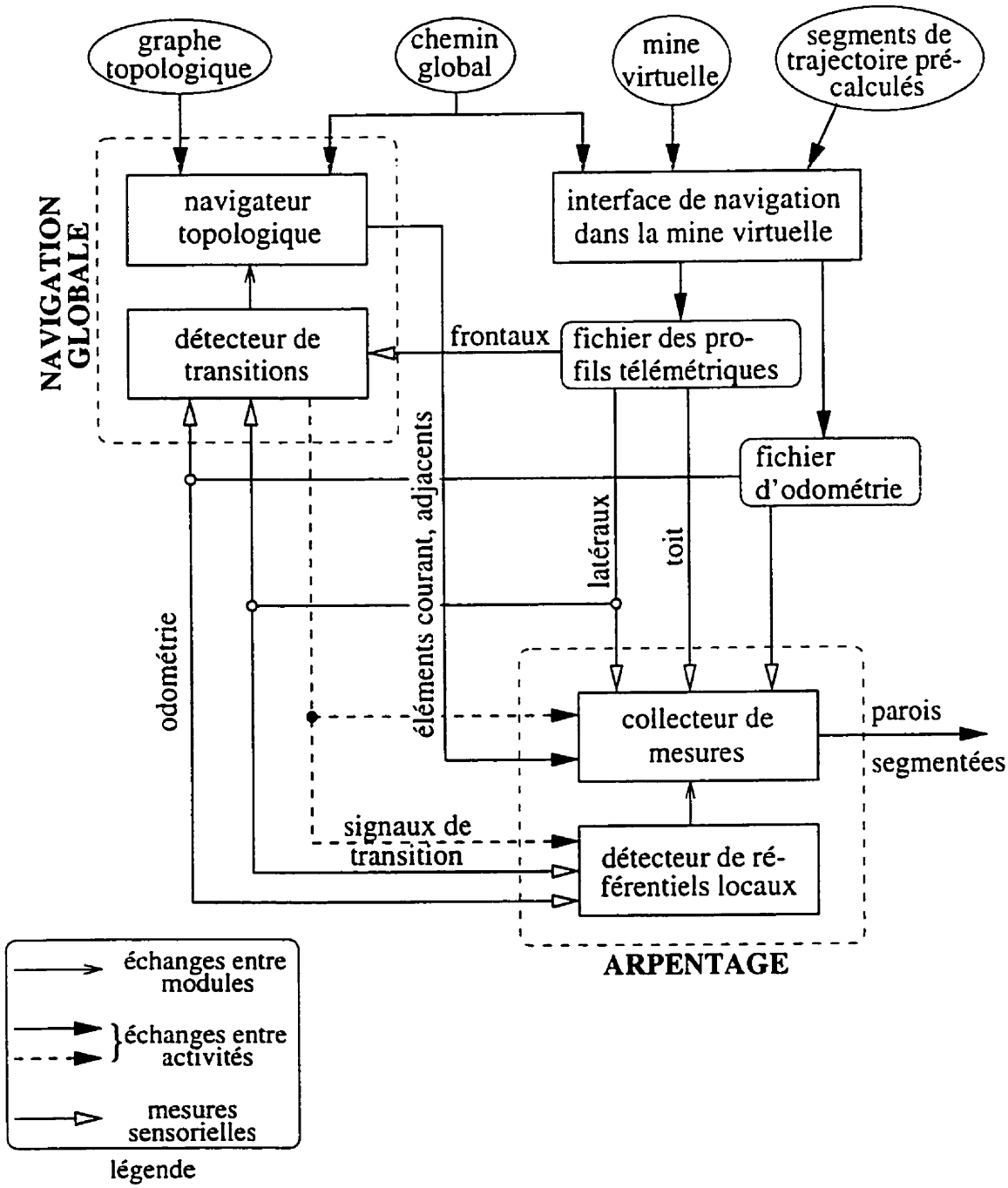


Figure B.1 : Système tel que réalisé par le simulateur

Notre simulateur fonctionne en quatre phases :

1. sélection du chemin global à suivre ;
2. navigation et collecte de mesures dans la mine virtuelle ;
3. détection des transitions de structure ;
4. segmentation des parois en galeries et intersections.

Normalement, dans un système réel, les processus 2 à 4 se dérouleraient simultanément. Dans la simulation, ils ont lieu séquentiellement à cause des contraintes de l'implantation informatique du système. En effet, le simulateur est réalisé non comme un seul programme faisant tous les calculs, mais plutôt comme une interface usager principale (exploitant le logiciel Matlab) qui appelle un autre programme (implanté avec la librairie infographique OpenGL) s'occupant des phases 1 et 2, puis gère en Matlab les phases 3 et 4. Nous nous référons ci-après à la partie Matlab du simulateur par le nom `MineArpent` et à la partie OpenGL par le nom `MineGL`. Notons que le fait d'implanter le système de cette manière facilite pour l'utilisateur la visualisation des différents processus et résultats impliqués dans l'arpentage automatisé. Dans la suite de l'annexe, nous décrivons les quatre phases du simulateur et illustrons son utilisation au moyen d'images en couleur captées sur écran d'ordinateur.

B.2 Phase 1 : sélection du chemin global

La première étape dans l'arpentage de la mine est de choisir le chemin global à suivre par le robot. La mine entière est affichée dans une fenêtre (vue d'oiseau) et il s'agit pour l'usager de sélectionner dans l'ordre à l'écran une série de régions contiguës de la mine ; le chemin global alterne forcément entre galeries et intersections. Le programme `MineGL` connaît le graphe topologique de la mine ainsi que la correspondance entre éléments géométriques de la mine virtuelle (*cf.* section A.1) et éléments topologiques.

La figure B.2 montre le graphe topologique de la mine virtuelle exploitée par le simulateur ; les galeries et intersections y sont étiquetées. Dans la figure B.3 nous voyons l'état de MineGL lorsqu'un chemin global composé de 6 galeries et 5 intersections a été sélectionné. Le chemin illustré est le suivant :

$$G_1 \rightarrow I_1 \rightarrow G_8 \rightarrow I_6 \rightarrow G_{20} \rightarrow I_{11} \rightarrow G_{22} \rightarrow I_{12} \rightarrow G_{25} \rightarrow I_{15} \rightarrow G_{32}$$

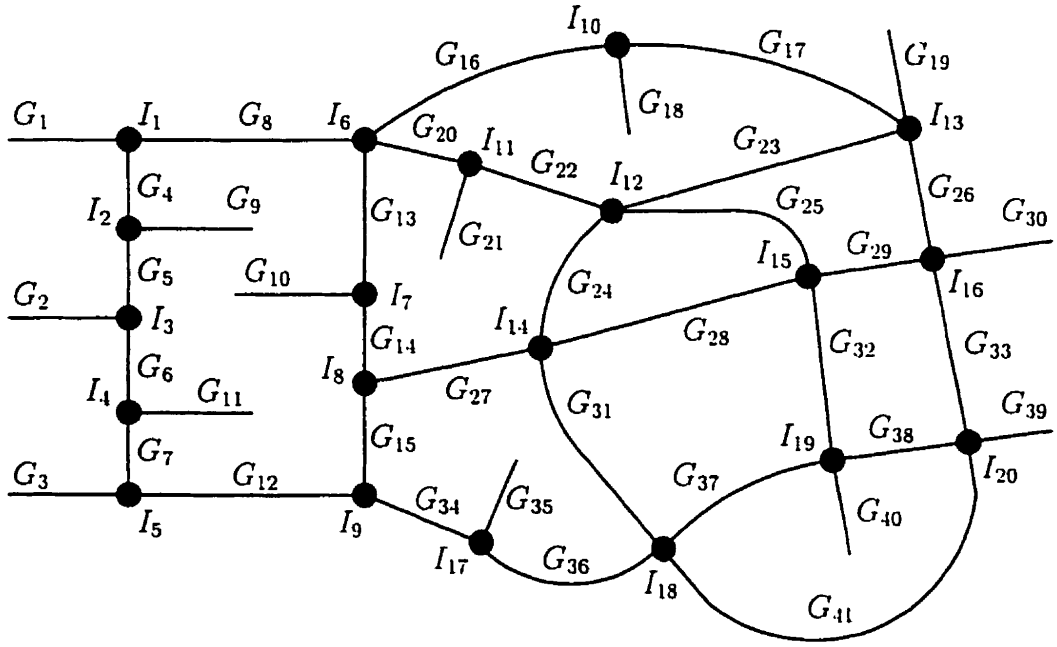


Figure B.2 : Topologie de la mine virtuelle utilisée par le simulateur

B.3 Phase 2 : navigation dans la mine virtuelle

Lorsque l'usager indique que le chemin global est complet, MineGL passe au mode de navigation et collecte des profils télémétriques. L'usager a alors devant lui l'interface de visualisation décrite à la section A.4, maintenant dotée de deux nouvelles fonctions. La première de ces fonctions consiste à afficher, dans le quadrant B (voir la figure B.4),

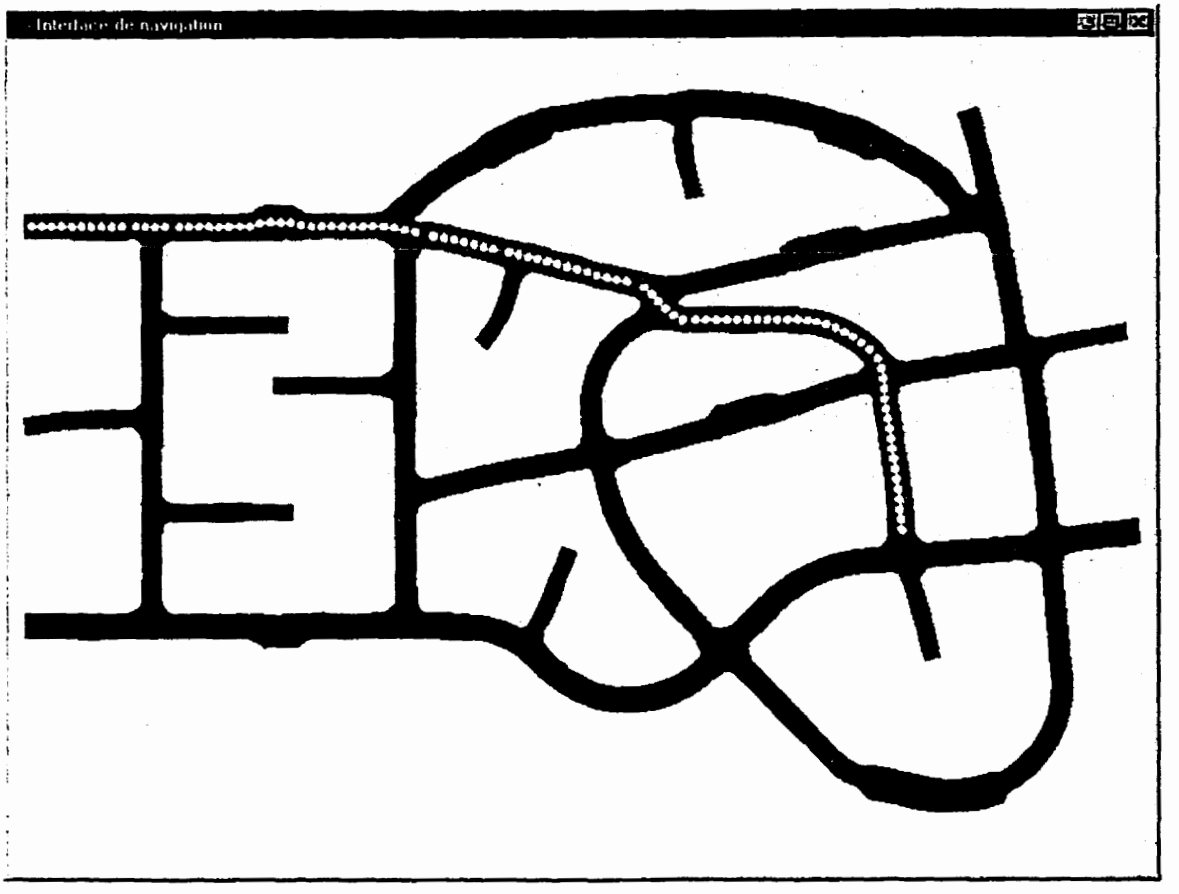


Figure B.3 : Sélection interactive du chemin global

les *positions théoriques*[1] du robot lors des événements importants dans la détection des transitions. En nous référant au développement de la section 5.4, ces événements sont *NotifEn*, *ConfirmEn*, *1Branche* et *ConfirmSo*. Nous nous basons sur le déroulement normal de la traversée d'une intersection, décrit dans la section 5.5.6, pour calculer les poses successives du robot P_{NE}, P_{CE}, P_{1b}, P_{p2} et P_{CS} associées à chaque intersection dans le chemin global. Nous affichons aussi les RLs théoriques associés aux jonctions d'entrée et sortie. Ceux-ci se situent là où les galeries se rattachent géométriquement aux intersections dans la mine virtuelle. La deuxième nouvelle fonction est reliée

[1]La véritable analyse des mesures pour la détection des transitions n'est faite qu'à la phase suivante, par le programme `MineArpent`.

à l'affichage des profils latéraux dans le quadrant C : l'usager peut alterner entre l'affichage en 2D des profils instantanés et l'affichage en 3D de l'accumulation des profils.

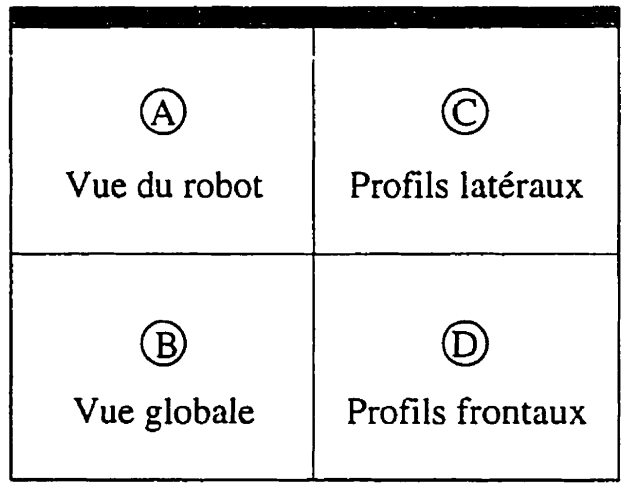


Figure B.4 : Schéma de l'interface de navigation avec ses 4 quadrants

Étant donné que les profils prennent un certain temps à calculer, nous générons au préalable les fichiers de profils correspondant à des "segments de chemin" traversant chaque galerie (dans les deux sens) et chaque intersection (en reliant chaque paire de galeries, dans les deux sens)[2]. Une fois le chemin global choisi, `MineGL` doit seulement récupérer les fichiers appropriés puis lire et afficher les profils. Ces fichiers sont concaténés pour former les fichiers de mesures passés à la phase 3. `MineGL` remplit aussi le rôle du module navigateur topologique : il passe à la phase 3 la description topologique de toutes les intersections dans la mine ainsi que celle de tous les branchements effectués dans le chemin global (deux fichiers supplémentaires).

La figure B.5 montre l'interface de navigation lorsque le robot vient de tra-

[2]En outre, nous générons plusieurs (en pratique, trois) trajectoires pour chaque segment de chemin en faisant varier la déviation sinusoïdale appliquée à la trajectoire idéale. Une fois le chemin global établi, une trajectoire est choisie aléatoirement parmi celles disponibles pour chaque segment de chemin. Ceci permet d'évaluer dans la simulation la stabilité du calcul des RLs par rapport aux variations de trajectoire.

verser l'intersection I_{11} dans le chemin de parcours sélectionné précédemment. Nous apercevons dans le quadrant B les marqueurs représentant la détection des transitions : en rouge, les événements reliés à l'analyse frontale (poses P_{NE} et P_{p2}), et en vert, les événements reliés à l'analyse latérale (poses P_{CE} et P_{CS} ainsi que les RLs associés aux jonctions). Dans le quadrant C, nous voyons l'accumulation en 3D des profils latéraux.

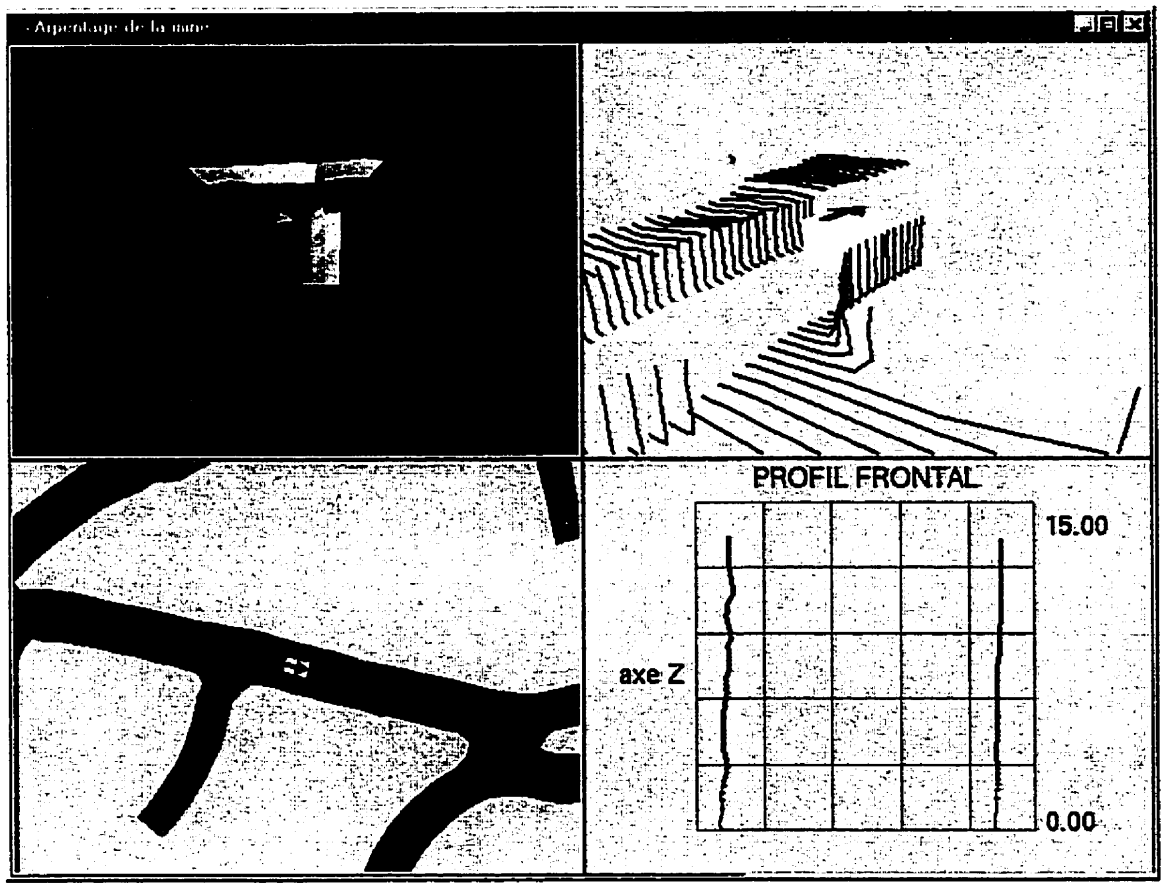


Figure B.5 : Navigation dans la mine virtuelle, affichage des profils et des marqueurs d'événements

B.4 Phase 3 : détection des transitions de structure

Les résultats expérimentaux de la section 5.6 ont démontré la performance du détecteur de transitions pour la traversée d'intersections *isolées*. Le programme `MineArpent` quant à lui, implante ce module dans le cas où le robot traverse de manière *continu* plusieurs intersections. Ceci requiert une fonction de supervision qui gère l'alternance entre phases d'analyse frontale et latérale. Nous considérons que le système a à tout moment un *état* qui prend l'une des quatre valeurs : *AnalyseFrontEn*, *AnalyseLatEn*, *AnalyseFrontSo* ou *AnalyseLatSo*. L'état initial du système dépend du premier élément topologique dans le chemin global. Si c'est une galerie, l'état initial est *AnalyseFrontEn*, et si c'est une intersection, *AnalyseFrontSo*.

De plus, la fonction de supervision doit traiter le cas particulier d'un élargissement important dans une galerie; celui-ci peut provoquer l'apparition d'occlusions dans l'image binaire frontale pendant suffisamment longtemps pour déclencher le signal *NotifEn*. Nous avons donc établi deux critères, l'un basé sur la largeur de l'espace navigable mesurée après *NotifEn*, l'autre sur le nombre de branches devant le robot, afin de reconnaître qu'il s'agit d'un élargissement et non d'une intersection. Le système retourne alors à l'état *AnalyseFrontEn*.

Lorsqu'une intersection est complètement traversée (signal *ConfirmSo*), `MineArpent` calcule les RLs *définitifs* associés aux jonctions d'entrée et sortie. Ceux-ci sont différents des derniers RLs trouvés aux états *AnalyseLat{En/So}* car le calcul exploite maintenant les parois entières mesurées depuis *NotifEn* jusqu'à *ConfirmSo* et exploite la notion de mur commun (*cf.* section 4.5.1). Les RLs définitifs sont stockés dans une structure globale décrivant toutes les intersections; ils servent ensuite à identifier les parois d'intersections (phase 4).

L'image de la figure B.6 montre l'interface usager graphique (IUG) de `MineArpent`. L'IUG comporte trois régions réservées à des graphiques Matlab : à gauche, un graphique principal où sont affichés l'accumulation des points de parois, le robot et sa trajectoire et les marqueurs des événements et des RLs calculés; à droite, deux

graphiques auxiliaires pour visualiser l'image binaire frontale (en haut) aux états *AnalyseFront{En/So}* et la détection des RLs (en bas) aux états *AnalyseLat{En/So}*. L'usager contrôle le déroulement de la simulation au moyen de plusieurs boutons au bas de l'IUG. À droite, l'usager sélectionne la phase courante : phases 1 et 2 (`MineGL`), phase 4, ou quitter le programme. À gauche, l'usager contrôle l'exécution de la phase 3 : lancer la trajectoire, halte, arrêter. Enfin, la barre-outil en haut de l'IUG, prédéfinie en Matlab, fournit trois boutons qui nous permettent de modifier l'échelle d'un graphique et de manipuler par rotation en 3D son point de vue d'affichage. Notons qu'il est possible de cumuler les mesures (et les résultats d'analyse) de plusieurs chemins de parcours ou encore d'effacer toutes les données accumulées.

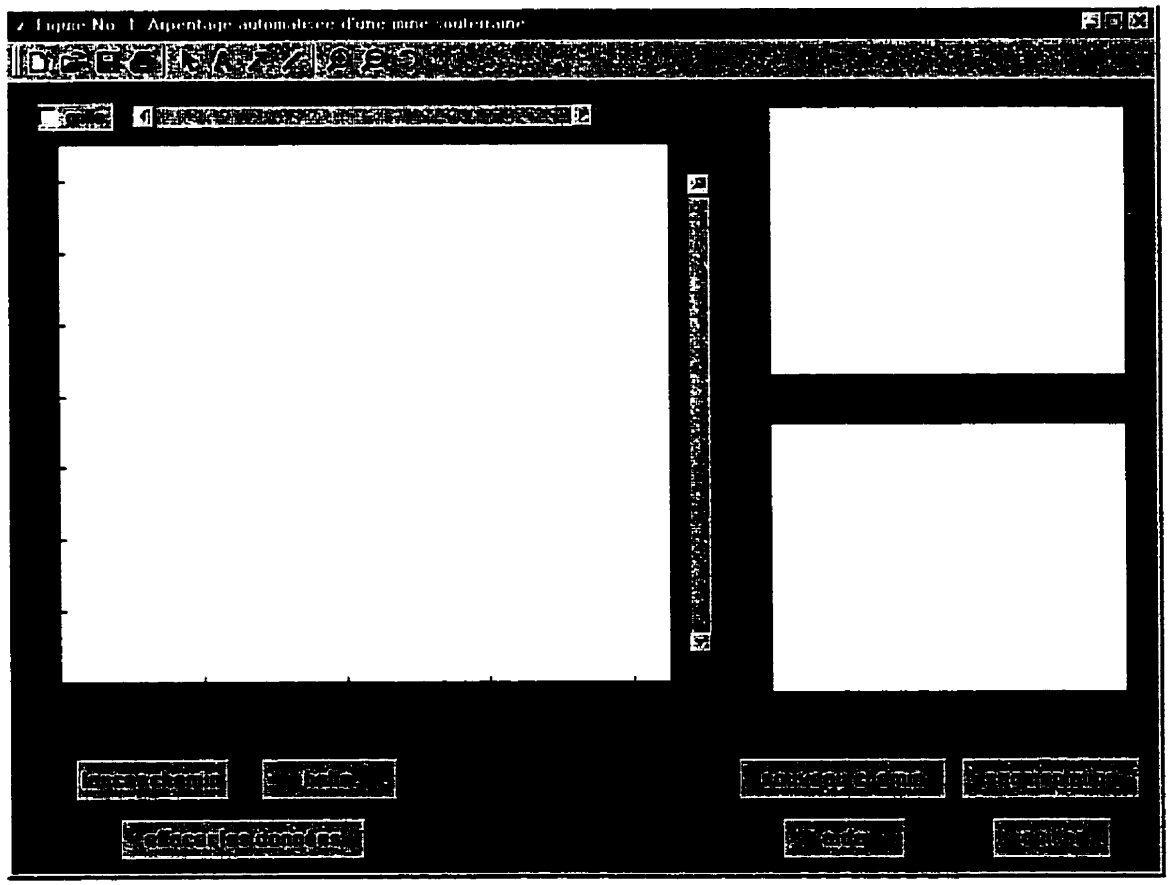


Figure B.6 : Interface usager graphique du programme `MineArpent`

Dans les figures B.7 et B.8 nous voyons l'IUG de `MineArpent` pendant l'exécution de la phase 3. Dans la première image, le robot est au même point que dans la figure B.5 dans le chemin global sélectionné précédemment. L'état courant du système est *AnalyseFrontEn* et de fait l'image binaire est visible. Les marqueurs des événements et des RLs dans le graphique principal suivent la même convention que dans `MineGL`. Dans la deuxième image, le robot est à la sortie de l'intersection I_{12} dans un nouveau chemin de parcours. L'état courant cette fois-ci est *AnalyseLatSo* et le calcul des RLs définitifs est visible (courbes modélisées, segments initiaux et finaux de la minimisation). Comme tous les RLs de l'intersection I_{12} sont maintenant déterminés, les points de parois appartenant à I_{12} sont identifiés et affichés dans une couleur distincte (*cf.* phase 4).

B.5 Phase 4 : segmentation des parois mesurées

Dans l'optique d'éventuellement fournir les données d'arpentage à un processus de modélisation géométrique de la mine, il est important de pouvoir pré-segmenter l'ensemble des mesures en morceaux décrivant séparément chaque galerie et intersection (*cf.* section 3.1.1). Notre but dans la simulation n'est pas d'implanter toute la fonctionnalité du module collecteur de mesure telle que décrite dans la section 4.2. Nous voulons seulement séparer les mesures appartenant aux intersections de celles appartenant aux galeries, puis visualiser cette segmentation.

Notre méthode s'appuie sur l'information topologique fournie par la phase 2 sur les intersections de la mine et les branchements effectués. Une structure globale (mentionné ci-haut) maintient une description de chaque intersection : son degré, les étiquettes des galeries rattachées et ses RLs définitifs à mesure que ceux-ci sont calculés. Lorsqu'une intersection est entièrement déterminée (c'est-à-dire, quand toutes les jonctions ont leurs RLs définitifs), alors l'intersection peut être segmentée. Nous définissons (dans le plan du sol) la région *intérieure* de l'intersection comme étant

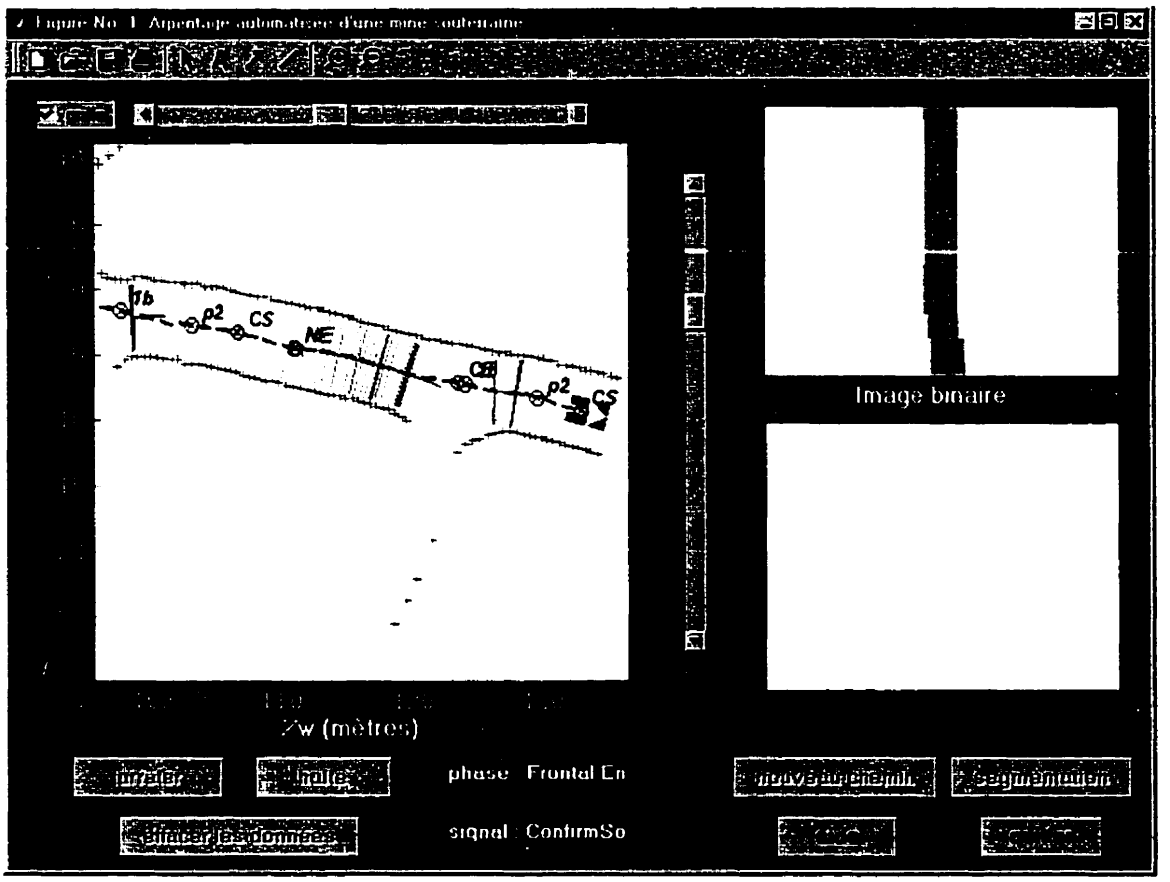


Figure B.7 : Détection des transitions à l'état *AnalyseFrontEn*

l'intersection des demi-plans définis par les droites passant par tous les RLs et par le barycentre des origines des RLs. Les points de parois appartenant à l'intersection sont ceux qui se trouvent dans la région intérieure.

Les images des figures B.9 et B.10 illustrent la segmentation des parois en galeries et intersections. Dans cet exemple, le robot a parcouru deux chemins à travers la mine : celui de la section B.2 et un deuxième qui est :

$$I_{19} \rightarrow G_{37} \rightarrow I_{18} \rightarrow G_{31} \rightarrow I_{14} \rightarrow G_{24} \rightarrow I_{12} \rightarrow \qquad \text{(B.1)}$$
$$G_{23} \rightarrow I_{13} \rightarrow G_{17} \rightarrow I_{10} \rightarrow G_{16} \rightarrow I_{6} \rightarrow G_{13}$$

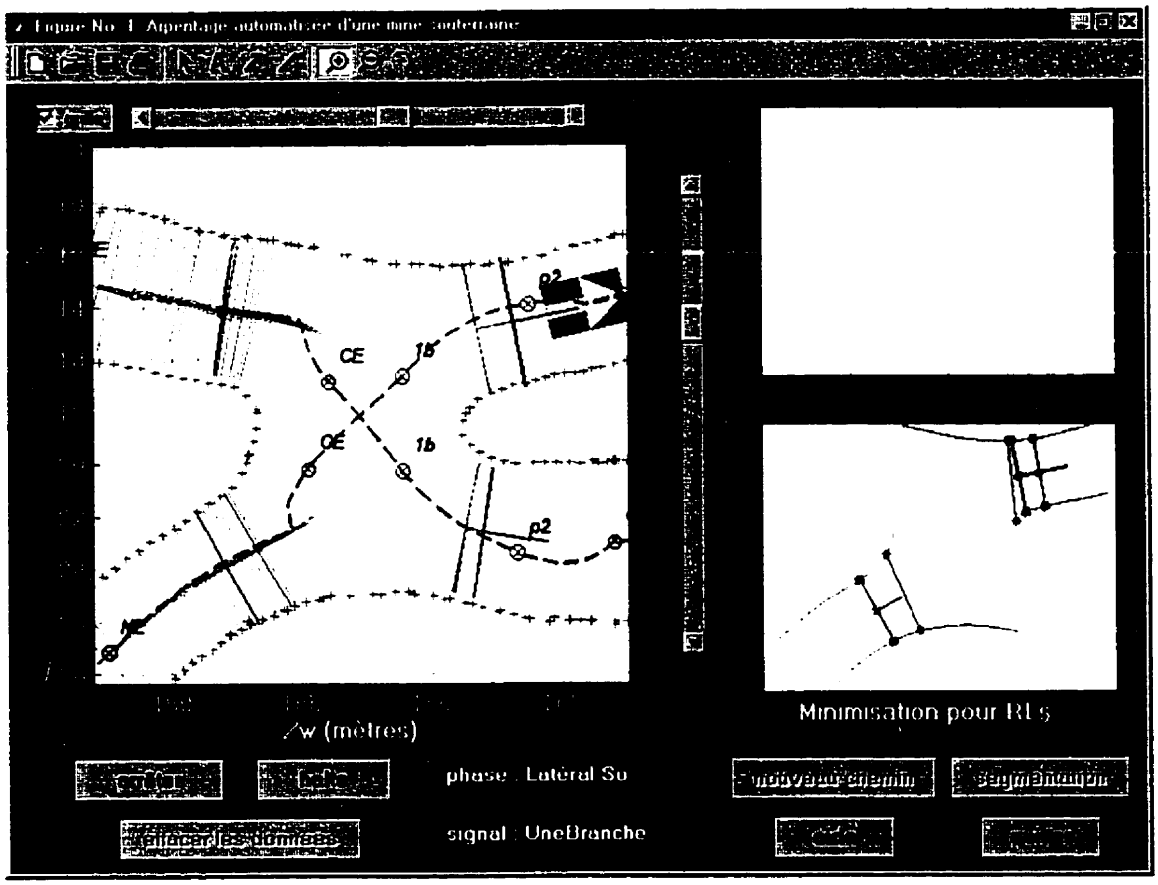


Figure B.8 : Détection des transitions à l'état *AnalyseLatSo*

Dans la figure B.9 nous avons la vue 2D (dans le plan du sol) de tous les points de parois et leur segmentation. Les intersections I_6 et I_{12} sont entièrement déterminées et sont affichées dans une couleur distincte. Par contre, les intersections I_1, I_{10}, I_{11}, I_{13}, I_{14}, I_{15}, I_{18} et I_{19}, elles, ne sont que partiellement déterminées. L'usager peut aussi voir la segmentation en 3D (figure B.10) ; dans cette image, le point de vue 3D est rapproché de I_6 et I_{12}. Ici, les galeries sont représentées par l'accumulation des mesures instantanées ($\text{Dlat}_{g,d}$, hauteur du toit) transformées par les poses du véhicule ; les intersections sont représentées par des surfaces approximant les murs, le toit et le sol.

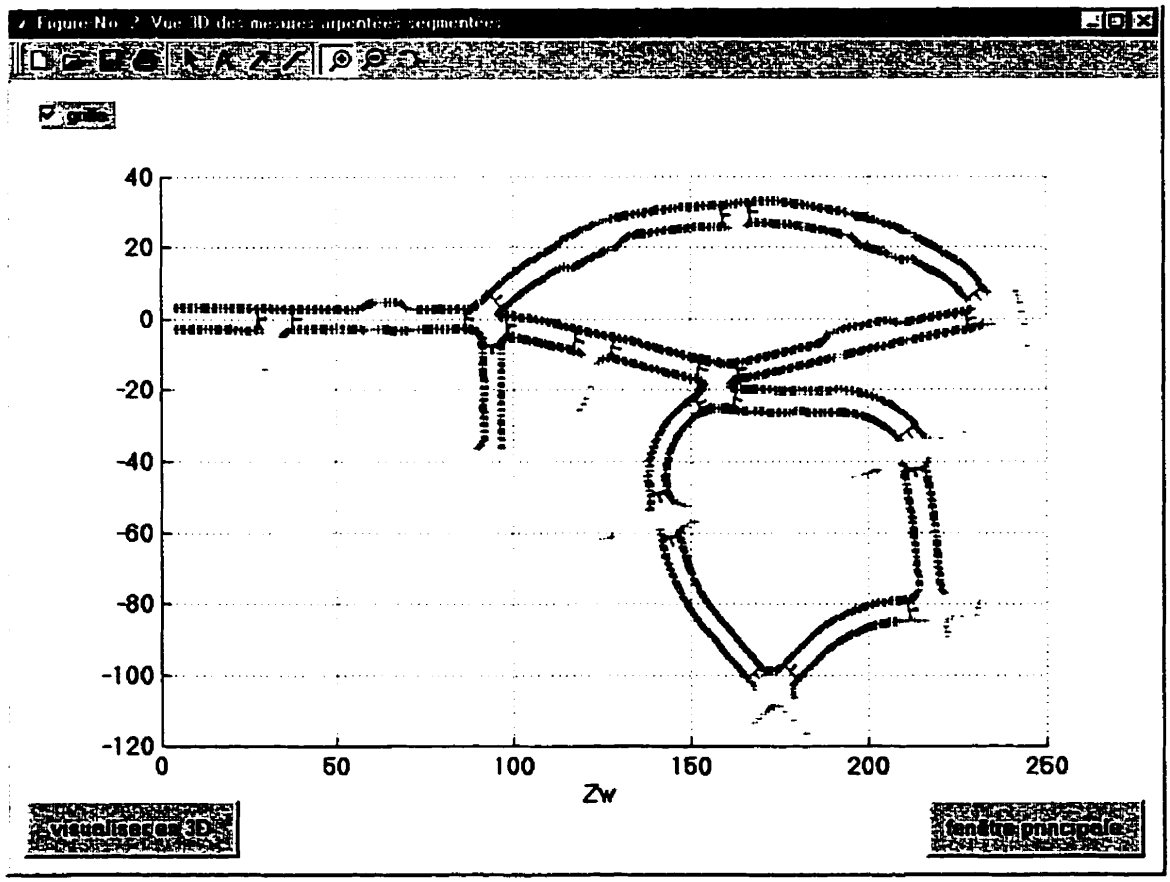


Figure B.9 : Vue 2D de la segmentation des parois en galeries et intersections

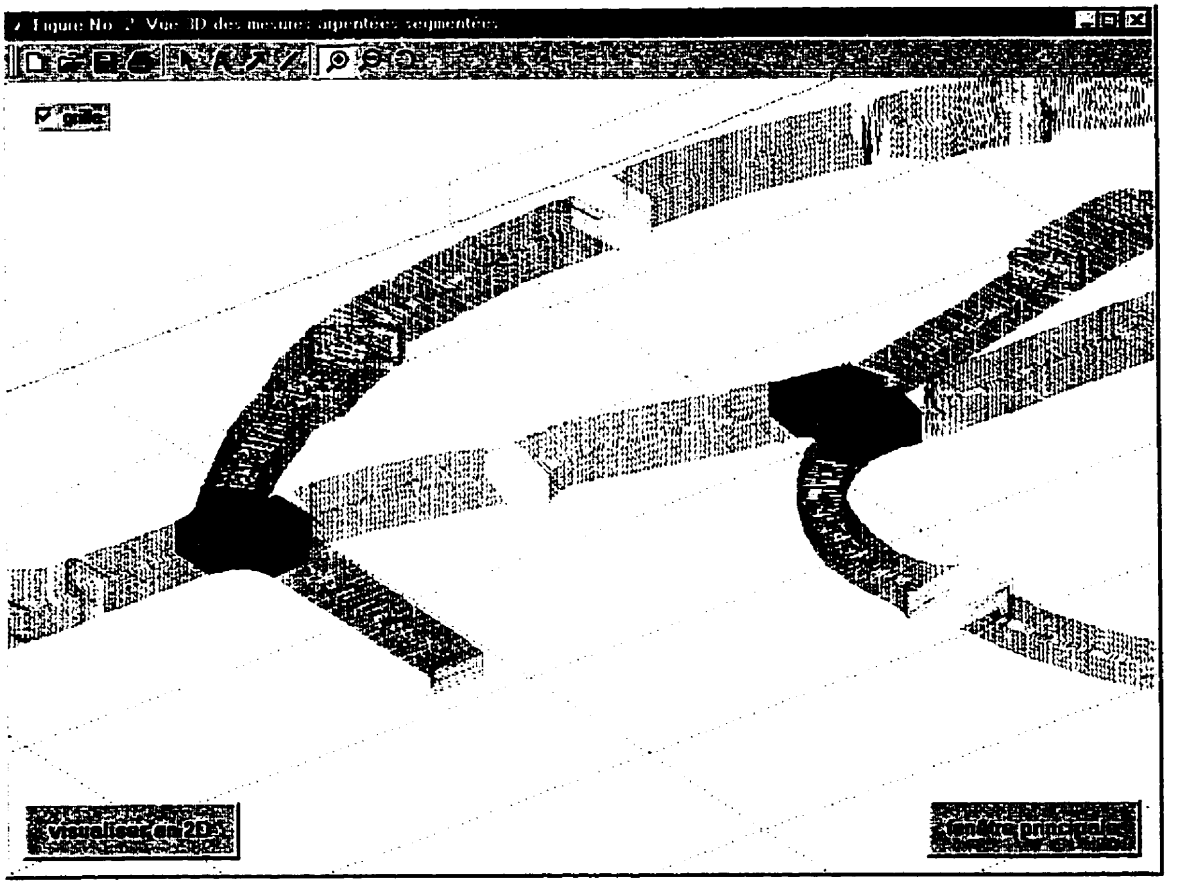


Figure B.10 : Vue 3D de la segmentation des parois en galeries et intersections

Bibliographie

ANDERSON, D. L. (1990), Laser tracking and tram control of a continuous mining machine, rapport technique RI 9319, U.S. Bureau of Mines, Pittsburgh, PA.

ARKIN, R. C. (1990), Integrating behavioral, perceptual, and world knowledge in reactive navigation, *Robotics and Autonomous Systems* **6**, 105–122.

BANTA, L., NUTTER, R. S. et XIA, Y. (1992), Mode-based navigation for autonomous mine vehicles, *IEEE Transactions on Industry Applications* **28**(1), 181–185.

BANTA, L. et RAWSON, K. D. (1994), Sensor fusion for mining robots, *IEEE Transactions on Industry Applications* **30**(5), 1321–1325.

BARSHAN, B. et DURRANT-WHYTE, H. F. (1995), Inertial navigation systems for mobile robots, *IEEE Transactions on Robotics and Automation* **11**(3), 328–342.

BÉRANGER, V. (1996), Construction et reconnaissance de vues panoramiques pour la localisation dans un environnement inconnu, Master's thesis, Ecole Polytechnique de Montréal.

BLAIS, F. et RIOUX, M. (1986), BIRIS: a simple 3-D sensor, dans *Proceedings of the SPIE: Optics, Illumination and Image Sensing for Machine Vision*, Vol. 728, Cambridge, MA, pp. 235–242.

BLAIS, F., RIOUX, M. et DOMEY, J. (1991), Optical range image acquisition for the

navigation of a mobile robot, dans *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, pp. 2574–2580.

BLAIS, F., RIOUX, M., DOMEY, J. et BERALDIN, J.-A. (1988), A very compact real time 3-D range sensor for mobile robot applications, dans *Proceedings of the SPIE: Mobile Robots III*, Vol. 1007, Cambridge, MA, pp. 330–338.

BORENSTEIN, J. et KOREN, Y. (1991*a*), Histogramic in-motion mapping for mobile robot obstacle avoidance, *IEEE Transactions on Robotics and Automation* **7**(4), 535–539.

BORENSTEIN, J. et KOREN, Y. (1991*b*), The vector field histogram - fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation* **7**(3), 278–288.

BORGHI, G. et BRUGALI, D. (1995), Autonomous map learning for a multi-sensor mobile robot using diktiometric representation and negotiation mechanism, dans *Proceedings of the International Conference on Advanced Robotics*, Spain.

BORTHWICK, S. et DURRANT-WHYTE, H. (1994), Simultaneous localisation and map building for autonomous guided vehicles, dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Munich, Germany, pp. 761–768.

BORTHWICK, S., STEVENS, M. et DURRANT-WHYTE, H. (1993), Position estimation and tracking using optical range data, dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Yokohama, Japan, pp. 2172–2177.

BROOKS, R. A. (1985), Visual map making for a mobile robot, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, MO, pp. 824–829.

BULATA, H. et DEVY, M. (1994), Perception of a mobile robot for navigation in

structured scenes, dans *Proceedings of ORIA '94*, Marseille, France, pp. 211–219.

CHATILA, R. et LAUMOND, J.-P. (1985), Position referencing and consistent world modeling for mobile robots, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, MO, pp. 138–145.

COX, I. J. (1991), Blanche — an experiment in guidance and navigation of an autonomous robot vehicle, *IEEE Transactions on Robotics and Automation* **7**(2), 193–204.

CROWLEY, J. L. (1985), Navigation for an intelligent mobile robot, *IEEE Journal of Robotics and Automation* **RA-1**(1), 31–41.

DEBANNÉ, P. (1997), Génération de données sensorielles dans un milieu de navigation simulé, rapport technique GRPR-RT-9706, Groupe de Recherche en Perception et Robotique, École Polytechnique, Montréal, Québec.

DEBANNÉ, P., HERVÉ, J.-Y. et COHEN, P. (1997), Global self-localization of a robot in underground mines, dans *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Orlando, FL, pp. 4400–4405.

DUDEK, G., FREEDMAN, P. et HADJRES, S. (1996), Mapping in unknown graph-like worlds, *Journal of Robotic Systems* **13**(8), 539–559.

DUDEK, G., JENKIN, M., MILIOS, E. et WILKES, D. (1993), Map validation and self-location in a graph-like world, dans *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1648–1653.

DUDEK, G., JENKINS, M., MILIOS, E. et WILKES, D. (1991), Robotic exploration as graph construction, *IEEE Transactions on Robotics and Automation* **7**(6), 859–865.

ELFES, A. (1987), Sonar-based real-world mapping and navigation, *IEEE Journal of Robotics and Automation* **RA-3**(3), 249–265.

ENGELSON, S. P. et MCDERMOTT, D. V. (1992), Error correction in mobile robot map learning, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, pp. 2555–2560.

FORSBERG, J., LARSSON, U. et WERNERSSON, Å. (1995), Mobile robot navigation using the range-weighted Hough transform, *IEEE Robotics and Automation Magazine* pp. 19–26.

GONZALEZ, J., STENTZ, A. et OLLERO, A. (1992), An iconic position estimator for a 2D laser rangefinder, dans *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, pp. 2646–2651.

GREINER, R. et ISUKAPALLI, R. (1996), Learning to select useful landmarks, *IEEE Transactions on Systems, Man and Cybernetics* **26**(3), 437–449.

HURTEAU, R., SAINT-AMANT, M., LAPERRIÈRE, Y. et CHEVRETTE, G. (1992), Optical guidance system for underground mine vehicles, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, pp. 639–644.

JUNEAU, L., HURTEAU, R., FREEDMAN, P. et CHEVRETTE, G. (1993), Using laser range data to model tunnel curvature for the automatic guidance of a mining vehicle, dans *Proceedings of the 2nd IEEE Conference on Control Applications*, pp. 643–648.

KAMON, I., RIVLIN, E. et RIMON, E. (1996), A new range-sensor based globally convergent navigation algorithm for mobile robots, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, pp. 429–435.

KITCHENER, L. C. (1986), 70 ton capacity, automated, underground trolley truck project, dans *Symposium: Applications of Automation in Mining Present and Future*, Sudbury, Ontario.

KROGH, B. H. (1984), A generalized potential field approach to obstacle avoidance

control, dans *Robotics Research: The Next Five Years and Beyond, SME Conference Proceedings*, Bethlehem, PA.

KROGH, B. H. et THORPE, C. E. (1986), Integrated path planning and dynamic steering control for autonomous vehicles, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 1664–1669.

KUC, R. et VIARD, V. B. (1991), A physically based navigation strategy for sonar-guided vehicles, *The International Journal of Robotics Research* **10**(2), 75–87.

KUIPERS, B. J. et BYUN, Y. T. (1987), A qualitative approach to robot exploration and map-learning, dans *Proceedings of the AAAI Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pp. 390–404.

KUIPERS, B. J. et LEVITT, T. S. (1988), Navigation and mapping in large-scale space, *AI Magazine* pp. 25–42.

LABROSSE, F. (1998), Modélisation générique d'objets et application aux mines, Thèse de doctorat, École Polytechnique de Montréal.

LABROSSE, F., DEBANNÉ, P., HERVÉ, J.-Y. et COHEN, P. (1998), Virtual environments for planning underground mining operations, dans *Proceedings of the Centennial Meeting of the Canadian Institute of Mining, Metallurgy and Petroleum/XVIth Congress of the Council of Mining and Metallurgical Institutions*, Montréal, Canada.

LABROSSE, F., HERVÉ, J.-Y. et COHEN, P. (1996), GNOMine : An application of the GNOME model to the representation of mines, rapport technique GRPR-RT-9611, Groupe de Recherche en Perception et Robotique, École Polytechnique, Montréal, Québec.

LANG, S., KORBA, L., BLAIS, F. et LECAVALIER, M. (1993), Characterization and testing of the BIRIS range sensor, dans *Proceedings of the IMTC Conference*, pp. 459–464.

LAPERRIÈRE, Y. (1991), Etude d'un système de guidage optique pour un véhicule minier articulé, Master's thesis, Ecole Polytechnique de Montréal.

LASIRIS, I. (n.d.), *Laser Diode Structured Light Products*.

LEONARD, J. J. et DURRANT-WHYTE, H. F. (1992), *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publishers, Boston, MA.

LEONARD, J. J., DURRANT-WHYTE, H. F. et COX, I. J. (1992), Dynamic map building for an autonomous mobile robot, *The International Journal of Robotics Research* **11**(4), 286–298.

LEVITT, T. S. et LAWTON, D. T. (1990), Qualitative navigation for mobile robots, *Artificial Intelligence* **44**, 305–360.

LU, F. et MILIOS, E. (1995), Optimal global pose estimation for consistent sensor data registration, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, Nagoya, Japan, pp. 93–100.

LUMELSKY, V. J. et STEPANOV, A. A. (1987), Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape, *Algorithmica* pp. 403–430.

MACKENZIE, P. et DUDEK, G. (1994), Precise positioning using model-based maps, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, pp. 1615–1621.

MATTHIES, L. et ELFES, A. (1988), Integration of sonar and stereo range data using a grid-based representation, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, Philadelphia, PA, pp. 727–733.

MIN, B.-K., CHO, D. W., LEE, S.-J. et PARK, Y.-P. (1996), Exploration of a mobile robot based on sonar probability mapping, *Journal of Dynamic Systems, Measurement, and Control* **118**, 150–157.

MORAVEC, H. P. (1988), Sensor fusion in certainty grids for mobile robots, *AI Magazine* **9**(2), 61–74.

MORAVEC, H. P. et ELFES, A. (1985), High resolution maps from wide angle sonar, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, D.C., pp. 116–121.

MOUTARLIER, P. et CHATILA, R. (1989), Stochastic multisensory data fusion for mobile robot location and environment modeling, dans *Proceedings of the 5th International Symposium on Robotics Research*, Tokyo, Japan, pp. 85–94.

ORIOLO, G., VENDITTELLI, M. et ULIVI, G. (1995), On-line map building and navigation for autonomous mobile robots, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2900–2906.

PAYTON, D. W. (1986), An architecture for reflexive autonomous vehicle control, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1838–1845.

PEITGEN, H.-O. et SAUPE, D., éditeurs (1988), *The Science of Fractal Images*. Springer-Verlag, Berlin, Germany.

PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A. et VETTERLING, W. T. (1988), *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, England.

RENCKEN, W. (1993), Concurrent localisation and map building for mobile robots using ultrasonic sensors, dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Yokohama, Japan, pp. 2192–2197.

RENCKEN, W. (1994), Autonomous sonar navigation in indoor, unknown and unstructured environments, dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Munich, Germany, pp. 431–438.

SAINT-AMANT, M., PICHÉ, A. et HURTEAU, R. (1988), Evaluation of automatic

guidance systems and their applicability to load haul and dump (LHD) vehicles, dans *Proceedings of the 3rd Canadian Symposium on Mining Automation*, Montréal, Québec, pp. 177–186.

SAMMARCO, J. J. (1988), On-board navigation for a continuous mining machine, dans *Proceedings of the 3rd Canadian Symposium on Mining Automation*, Montréal, Québec, pp. 159–163.

SAMMARCO, J. J. (1990), Heading control for a continuous mining machine, dans *Proceedings of the 4th Canadian Symposium on Mining Automation*, pp. 37–45.

SCHNAKENBERG, G. H. (1988), U.S. Bureau of Mines coal mining automation research, dans *Proceedings of the 3rd Canadian Symposium on Mining Automation*, Montréal, Québec, pp. 145–157.

SGOUROS, N., PAPAKONSTANTINOU, G. et TSANAKAS, P. (1996), Localized qualitative navigation for indoor environments, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, pp. 921–926.

SHAFFER, G. et STENTZ, A. (1992), A robotic system for underground coal mining, dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, pp. 633–638.

SHAFFER, G., STENTZ, A., WHITTAKER, W. et FITZPATRICK, K. (1992), Position estimation for underground mine equipment, *IEEE Transactions on Industry Applications* **28**(5), 1131–1140.

SIMMONS, R. et KOENIG, S. (1995), Probabilistic robot navigation in partially observable environments, dans *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1080–1087.

SOLDO, M. H. (1990), Reactive and preplanned control in a mobile robot, dans

Proceedings of DARPA Image Understanding Workshop, Pittsburgh, PA, pp. 792–796.

STEELE, J., GANESH, C. et KLEVE, A. (1993), Control and scale model simulation of sensor-guided LHD mining machines, *IEEE Transactions on Industry Applications* **29**(6), 1232–1238.

STEELE, J. P. H., KING, R. H. et STRICKLAND, W. H. (1991), Modeling and sensor-based control of an autonomous mining machine, dans *Proceedings of the International Symposium on Mine Mechanization and Automation*, Vol. 1, pp. 6–55–6–67.

STENTZ, A. et HEBERT, M. (1995), A complete navigation system for goal acquisition in unknown environments, dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 425–432.

STUCK, E. R., MANZ, A., GREEN, D. A. et ELGAZZAR, S. (1994), Map updating and path planning for real-time mobile robot navigation, dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Munich, Germany, pp. 753–760.

WILLIAMS, W. R. (1983), *Mine Mapping and Layout*, Prentice-Hall.

YAMAUCHI, B. et BEER, R. (1996), Spatial learning for navigation in dynamic environments, *IEEE Transactions on Systems, Man and Cybernetics* **26**(3), 496–505.