

Université de Technologie de Compiègne
Centre de Robotique, d'Electrotechnique et d'Automatique

THESE

présentée par

Cyril DROCOURT

pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE

Discipline : Robotique

Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels

Soutenue le 22 février 2002 devant le jury composé de :

M.	D. MEIZEL	Professeur à l'Université de Technologie de Compiègne	Président
M.	R. CHATILA	Directeur de recherche CNRS au LAAS, Toulouse	Rapporteur
M.	P. RIVES	Directeur de recherche INRIA à Sophia Antipolis	Rapporteur
Mme.	V. CHERFAOUI	Maître de Conférences à l'Université de Technologie de Compiègne	Examineur
M.	E. BRASSART	Maître de Conférences à l'Université de Picardie Jules Verne	Examineur
M.	L. DELAHOUCHE	Maître de Conférences à l'Université de Picardie Jules Verne	Directeur de thèse

Aux vrais amis qui sont toujours présents, même dans les moments difficiles.

Remerciements

Le travail accompli dans le cadre de cette thèse a été effectué au Centre de Robotique, d'Electrotechnique et d'Automatique (CREA) de l'Université de Picardie Jules Verne. Je remercie les deux directeurs consécutifs du CREA, Monsieur Gérard André Capolino et Monsieur Mustapha Mouaddib, pour m'avoir accueilli au sein de cette structure.

J'exprime toute ma gratitude à Monsieur Claude Pégard, ancien chef du département Informatique de l'Institut Universitaire de Technologie d'Amiens pour m'avoir permis d'enseigner au sein de ce département durant ces années de thèse.

Je remercie tous les membres de mon jury, Monsieur Dominique Meizel pour l'honneur qu'il me fait de le présider, Monsieur Raja Chatila et Monsieur Patrick Rives pour l'intérêt qu'ils ont porté à mon travail en acceptant d'être rapporteurs, et à Madame Véronique Berge-Cherfaoui pour l'estime qu'elle m'a fait de participer à ce jury en tant qu'examinateur.

Je remercie sincèrement Monsieur Eric Brassart, chef du département informatique d'Amiens, pour avoir accepté de participer à ce jury de thèse en tant qu'examinateur, malgré un emploi du temps chargé, et aussi pour les grandes qualités qu'il possède et qu'il a su conserver : dévouement, honnêteté et simplicité.

Je souhaite surtout remercier Laurent Delahoche, mon directeur de thèse, pour m'avoir encadré durant ces trois dernières années. Il a su guider efficacement mes travaux de recherche tout en me laissant une liberté très appréciable. Je tiens à m'excuser auprès de Maud, sa compagne, pour les nombreuses soirées et vacances qu'il a su me consacrer, mais aussi auprès de sa planche à voile, qu'il a négligé par ma faute ces dernières années.

Je ne peux citer toutes les personnes qui ont contribué à l'élaboration de cette thèse ou simplement à l'ambiance qu'ils ont apportée ces dernières années, mais l'amitié de Cyril et Arnaud, leur bonne humeur et leurs disponibilités ont rendu ce travail agréable à réaliser au quotidien. Je remercie également Bruno pour sa sincérité et son enthousiasme, Christophe et Cédric pour les liens privilégiés que j'ai entretenus avec eux, et en règle générale, l'ensemble des personnes que j'ai pu côtoyer au sein de l'IUT d'Amiens.

Enfin mes dernières pensées vont à l'ensemble des personnes de ma famille qui m'ont toujours encouragé à poursuivre cette voie et soutenus dans les moments difficiles, à mon frère Laurent et plus particulièrement à ma mère.

Table des matières

Introduction générale	1
Chapitre 1 – Etat de l'art.....	5
1.1. INTRODUCTION	6
1.2. LES SYSTEMES DE PERCEPTION	6
1.2.1 <i>Les capteurs proprioceptifs</i>	7
1.2.1.1. Les capteurs de déplacement	8
1.2.1.2. Les capteurs d'attitude	8
1.2.2 <i>Les capteurs extéroceptifs</i>	9
1.2.2.1. Les capteurs télémétriques	10
1.2.2.2. Les systèmes de vision.....	11
1.3. LES METHODES DE LOCALISATION	19
1.3.1 <i>Localisation a l'estime</i>	20
1.3.2 <i>Localisation absolue</i>	21
1.3.2.1. Repères artificiels.....	21
1.3.2.2. Repères naturels	23
1.3.3 <i>Méthodes hybrides</i>	27
1.3.3.1. Le recalage dynamique	27
1.3.3.2. La coopération multi-capteur	28
1.4. LOCALISATION ET MODELISATION SIMULTANEE DE L'ENVIRONNEMENT	29
1.4.1 <i>Les approches métriques</i>	30
1.4.1.1. Les méthodes de modélisation géométriques	30
1.4.1.2. Les méthodes de modélisation basées sur des grilles d'occupation	31
1.4.2 <i>Les approches topologiques</i>	33
1.4.3 <i>Conclusion sur les méthodes de modélisation</i>	35
1.5. CONCLUSION.....	35

Chapitre 2 – Coopération entre deux systèmes de vision omnidirectionnelle pour la localisation	37
2.1. INTRODUCTION	38
2.2. LE SYSTEME DE PERCEPTION OMNIDIRECTIONNEL STEREOSCOPIQUE	38
2.3. LA SEGMENTATION RADIALE	41
2.4. APPARIEMENT DES SECTEURS	44
2.4.1 <i>La théorie des croyances de Dempster-Shafer</i>	45
2.4.2 <i>Calcul des critères d'appariement</i>	48
2.4.3 <i>Fusion des critères d'appariement</i>	50
2.4.4 <i>Algorithme global d'appariement des secteurs</i>	51
2.5. CALCUL DES POINTS DE L'ENVIRONNEMENT DANS LE MODELE SENSORIEL	53
2.6. ALGORITHMES DE LOCALISATION	53
2.6.1 <i>Les primitives considérées</i>	53
2.6.2 <i>Algorithmes de mise en correspondance</i>	55
2.7. RESULTATS EXPERIMENTAUX	60
2.8. CONCLUSION.....	66

Chapitre 3 – Mise à jour de la carte de l'environnement au sens des moindres carrés	67
3.1. INTRODUCTION	68
3.2. METHODE DE DECISION PAR FUSION D'INFORMATIONS	69
3.2.1 <i>Détermination de la croyance accordée à chaque association</i>	69
3.2.2 <i>Généralisation de l'opérateur de combinaison de Dempster-Shafer</i>	73
3.2.3 <i>Algorithme de décision</i>	74
3.3. MISE A JOUR INCREMENTALE DE LA CARTE DE L'ENVIRONNEMENT	76
3.3.1 <i>Rappel sur les M.C.R.</i>	77
3.3.2 <i>Mise à jour des coordonnées d'un amer</i>	78
3.3.3 <i>Phase d'initialisation de l'estimation de position d'une balise</i>	80
3.3.4 <i>Traitement des liens entre amers verticaux</i>	81
3.4. TRAITEMENTS POST-MODELISATION	84
3.4.1 <i>Maintien de la cohérence par filtrage</i>	84
3.4.2 <i>Gestion des liens dans le cas d'une suppression d'un point de la carte</i>	85
3.5. RESULTATS EXPERIMENTAUX	86
3.5.1 <i>Tests en environnement fermé et structuré sur de petits trajets</i>	86
3.5.2 <i>Résultats en environnement de type couloir sur des trajets importants</i>	89
3.6. CONCLUSION.....	94

Chapitre 4 – Localisation d’un robot mobile par inversion ensembliste	95
4.1. INTRODUCTION	96
4.2. RAPPELS SUR LA THEORIE ENSEMBLISTE	97
4.2.1 <i>Définition d’un problème d’inversion ensembliste</i>	97
4.2.2 <i>Synthèse sur l’analyse par intervalles</i>	98
4.2.2.1. Opérations sur les intervalles	98
4.2.2.2. Fonctions d’inclusion.....	99
4.2.3 <i>L’algorithme SIVIA</i>	101
4.3. LOCALISATION D’UN ROBOT MOBILE PAR L’UTILISATION DE SIVIA	105
4.3.1 <i>Définition du problème</i>	105
4.3.2 <i>Caractérisation du modèle d’erreur</i>	106
4.3.2.1. Premier modèle d’erreur pour la localisation.....	106
4.3.2.2. Deuxième modèle d’erreur pour la localisation.....	108
4.3.3 <i>Algorithme de localisation</i>	110
4.3.4 <i>Comment utiliser l’information odométrique ?</i>	112
4.4. RESULTATS EXPERIMENTAUX	114
4.4.1 <i>Résultats expérimentaux utilisant le premier modèle d’erreur</i>	114
4.4.2 <i>Résultats expérimentaux utilisant le second modèle d’erreur</i>	116
4.4.3 <i>Comparaison des deux modèles d’erreur</i>	117
4.5. CONCLUSION.....	118

Chapitre 5 – Une approche ensembliste au problème de mise à jour incrémentale de la carte	119
5.1. INTRODUCTION	120
5.2. REPRESENTATION DES AMERS DANS LA CARTE DE L’ENVIRONNEMENT	121
5.3. ALGORITHME DE LOCALISATION MODIFIE	124
5.4. METHODE DE DECISION POUR LA MISE EN CORRESPONDANCE.....	125
5.4.1 <i>Détermination de la croyance accordée à chaque association</i>	125
5.4.2 <i>Algorithme de décision</i>	130
5.5. MISE A JOUR INCREMENTALE DE LA CARTE DE L’ENVIRONNEMENT	131
5.6. RESULTATS EXPERIMENTAUX	133
5.6.1 <i>Tests en environnement structuré de type hall</i>	133
5.6.2 <i>Résultats en environnement de type couloir sur des trajets importants</i>	136
5.7. CONCLUSION.....	142

Conclusion générale et perspectives	143
--	------------

Références bibliographiques	147
--	------------

Bibliographie de l’auteur.....	154
---------------------------------------	------------

Introduction générale

L'homme a toujours cherché à concevoir des entités à l'image de l'être humain sur lesquelles ils pourraient exercer son autorité, exécutant ainsi ses moindres désirs. Les premières approches remontent à l'époque médiévale, où des automates étaient placés dans les églises afin d'impressionner les paysans, fascinés par ces "êtres suprêmes". Les automates sont ensuite devenus des jouets populaires au XVIII^e siècle sous forme miniature. La littérature a également fourni bons nombres d'ouvrages concernant des êtres artificiel. On peut citer Mary Shelley, qui en 1818 écrivit Frankenstein, créature à l'apparence humaine dont les parties de son corps tenaient ensemble fixés par des boulons et des écrous. En 1921, Karel Capek introduisit le terme "robot" dans sa pièce de théâtre "R.U.R." (Rossum's Universal Robot), du tchèque "robota" signifiant travail forcé, corvée. Il raconte l'histoire d'un savant appelé Rossum, ayant réussi à mettre au point des créatures semblables physiquement à des être humains, que son fils exploita au sein de son entreprise. Le terme "robotique" fut lui amené par l'écrivain Isaac Asimov, qui proposa les trois lois de la robotique suivantes :

- Loi 1 : Un robot ne peut blesser un être humain ni, par son inaction, permettre qu'un humain soit blessé.
- Loi 2 : Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la Première Loi.
- Loi 3 : Un robot doit protéger sa propre existence aussi longtemps qu'une telle protection n'est pas en contradiction avec la Première et/ou la Deuxième Loi.

On retrouve finalement parmi ces représentations, à la fois la volonté et la fascination d'avoir à sa disposition des robots exécutant ses désirs, mais aussi la peur de les créer de façon trop fidèle à l'être humain. En effet, qu'advierait-il en cas de révolte ?

Un tournant fut franchi dans les années 60, où les robots industrielles firent leur apparition essentiellement dans l'industrie automobile. Ces derniers n'ont plus l'apparence humaine comme les avaient imaginé leurs créateurs, mais ont l'allure de bras mécaniques, exécutant aveuglément les mêmes mouvements. Ce premier type fut appelé robots de 1^{ère} génération. La 2^{ème} génération quant à elle introduisit les fonctions de perception et de régulation issus de l'automatisme et permettant les corrections de mouvement par rapport à l'évolution des paramètres de l'environnement. Finalement, la 3^{ème} génération concerne les robots capables d'interagir avec l'environnement, de percevoir et de modéliser l'ensemble de son espace d'évolution et de prendre des décisions.

Avec l'évolution des technologies et les nombreux travaux de recherche réalisés, des progrès ont été réalisés sur la conception de robots mobiles. La partie mécanique est maintenant une étape presque parfaitement maîtrisée. Les différentes architectures mécaniques existantes permettent d'envisager tous les types de déplacement dans l'espace. Le plus simple est bien évidemment l'utilisation de roues, qui à moindre coût peut assurer un déplacement du robot sur un sol plat. Toutefois, l'obstination de l'homme à copier la nature l'a incité à développer d'autres solutions comme les robots à pattes, les robots rampants, ... L'un des buts ultimes consistant à réaliser une machine pouvant se déplacer comme l'être humain est maintenant une chose tout à fait envisageable. En effet, bon nombre de grands constructeurs les ont déjà intégrés à leurs catalogues, comme Sony avec le SD3-3X, un robot

bipède de 50 cm de hauteur, ainsi que Fujitsu qui commercialise le HOAP-1 avec son kit de développement sous Linux. Honda quant à lui a fabriqué Asimo un robot de taille humaine, comme Isamu de l'Université de Tokyo. Tout ceci pourrait presque nous faire penser que George Lucas ne s'était pas trompé en imaginant le robot C-3PO dans ses films Star Wars.

Toutefois, dans la domaine de "l'intelligence artificielle", les progrès sont moindres. En 1950, Turing définit un test, composé de la manière suivante : une personne communique au moyen d'un clavier avec un ordinateur sans savoir que s'en est un. Si la personne ne peut pas déterminer si elle parle à un humain ou à une machine, l'ordinateur est considéré intelligent. Il existe maintenant un grand nombre de programmes capable de simuler des discussions par l'analyse de mots clefs. D'autres exemples prouvent que certains raisonnements artificiels ne sont pas une preuve d'intelligence. Par exemple en 1997, l'ordinateur Deep Blue d'IBM réussit à battre le champion du monde des échecs Garry Kasparov. Finalement, l'intelligence se définit essentiellement comme la capacité d'apprentissage, comme le souligne Francisco J. Varela : "*...il devenait plus clair que l'intelligence la plus profonde et la plus fondamentale est celle du bébé qui acquiert le langage...*". Même si la problématique reste plus modeste, la capacité d'un robot à intégrer des informations sur l'environnement relève donc d'une certaine forme d'apprentissage.

La navigation d'un robot mobile en milieu conditionné, c'est à dire un milieu intérieur parfaitement connu, dans lequel on a disposé ou pas des balises permettant au robot de se repérer, reste quelque chose de réalisable. Néanmoins, la partie "intelligente" du système est loin d'être maîtrisée. En effet, lors de la mission Mars Pathfinder de 1996, la choix de la NASA s'est porté sur la conception de Sojourner, un robot piloté par un processeur Intel 80C85 vieux de plus de dix ans. Le robot était donc piloté presque entièrement de la terre se trouvant pourtant à des millions de kilomètres. Ce choix montre bien les difficultés que représente la réalisation d'un robot autonome capable d'évoluer en milieu inconnu. Pourtant bon nombre de chercheurs se sont intéressés à la possibilité de donner à un robot les moyens de percevoir un environnement complètement inconnu, motivés par de vastes domaines d'applications. On peut notamment penser aux applications d'aides aux personnes handicapées, comme les fauteuils roulants intelligents, le domaine automobile où on souhaiterait pouvoir réaliser des véhicules n'ayant plus besoin de chauffeurs. Mais comment reproduire cette analyse de l'environnement que notre cerveau effectue en temps réel et de façon naturelle ?

Les progrès en matière de capteurs ont été significatifs ces dernières années en terme de précision, même si cela est souvent au détriment de la quantité de donnée qu'il faut manipuler, qui reste toutefois loin de celle des "capteurs" humains. Par exemple la rétine contient environ 125 millions de capteurs, contre 25 millions pour un film photographique, et un nombre encore bien inférieur pour une caméra CCD. De plus, les informations fournies par ces divers capteurs peuvent être de type tout à fait différent : distance avec un objet, vitesse, déplacement relatif, ... La diversité des informations fournies par ces capteurs a nécessité un développement de méthodes de fusion de données permettant d'obtenir une information sensorielle globale. Finalement, on peut s'apercevoir que la conception d'un robot et la perception de l'environnement sont déjà quasiment maîtrisés : Le robot peut se déplacer et percevoir des informations sur le monde l'entourant pour se localiser. Toutefois, une dernière étape est encore nécessaire pour libérer le robot d'une quelconque connaissance préalable : un apprentissage de son milieu d'évolution en fonction des informations sensorielles.

Cadre, objectifs et contributions

Ces travaux de thèse ont été réalisés au sein du C.R.E.A. (Centre de Robotique Electrotechnique et Automatique) d'Amiens, et plus particulièrement au sein de l'équipe "Perception en Robotique" animé par Eric Brassart. Cette équipe est composée de trois axes principaux :

- ✓ Localisation et modélisation de l'environnement
- ✓ Reconnaissance d'objets
- ✓ Coopération multi-robots

Le premier axe est fédérateur de nombreux travaux en robotique : Localisation d'un robot [BRA95] [MAR99] [PEG99], Coopération multi-capteurs [DEL97a] [CLE99], calibration [CAU00]. L'une des particularités de la majorité de ces travaux réside dans l'utilisation du capteur SYCLOP (SYstème Conique pour la LOCALisation et la Perception), qui est un capteur de vision omnidirectionnelle constitué d'un miroir conique sous lequel est placé une caméra CCD. Ce système permet l'obtention d'une image sur 360° en une seule acquisition. Ce capteur a essentiellement été utilisé comme système de localisation absolu puis, grâce à la coopération avec des capteurs télémétriques, il a permis la mise à jour de la carte de l'environnement.

C'est dans la continuité de ces travaux que s'inscrit ce mémoire. Les objectifs sont dans un premier temps de construire un modèle sensoriel basé sur l'utilisation conjointe de deux capteurs omnidirectionnels. Dans un deuxième temps nous avons tenté d'utiliser ces informations pour mettre en oeuvre une méthode de modélisation incrémentale de l'environnement sans connaissance a priori.

Organisation du mémoire

Dans un premier temps, nous proposons un état de l'art concernant tous les points devant être abordés pour l'intégration d'un système de navigation sur un robot. Ce chapitre commence par une étude des différents capteurs disponibles sur le marché. Nous abordons ensuite les méthodes de localisations permettant d'estimer de façon robuste la configuration d'un robot. Enfin, nous terminons par une synthèse sur les méthodes de modélisation de l'environnement .

Le deuxième chapitre est consacré aux traitements des données sensorielles issues de la coopération de deux systèmes de vision omnidirectionnelles. Après avoir brièvement présenté le capteur utilisé, nous proposons une méthode d'extraction de droites radiales basée sur l'utilisation du gradient. Ce traitement est essentiel pour une mise en correspondance des secteurs sur les deux images. L'appariement de ces secteurs est obtenu par l'utilisation de critères discriminants, basés sur le niveau de gris, et fusionnés par les règles de combinaison de Dempster-Shafer. Cette étape nous permettra de calculer les coordonnées des points observés dans le repère du robot par triangulation. Finalement, deux méthodes de localisation, basées sur la mise en correspondance de points seront détaillées et expérimentés sur plusieurs acquisitions.

La qualité de la phase de localisation nous permettra de proposer une méthode de modélisation incrémentale de la carte de l'environnement sans connaissance a priori. Nous parlerons alors de paradigme de localisation et modélisation simultané. Après avoir rappelé les bases de la généralisation des règles de Dempster-Shafer développées par D. Gruyer, nous introduisons une méthode de décision pour intégrer et fusionner des données au sein d'une

carte. Cette règle est basée sur les deux angles d'observations de l'amer. Ensuite, nous exposerons la méthode de génération de carte choisie, utilisant les Moindres Carrés Récursifs, qui nous permettra non seulement d'insérer de nouveaux points dans la carte, mais aussi de mettre à jour ceux déjà existants à partir d'observations. Les résultats expérimentaux effectués dans deux environnements différents, poseront le problème de la quantification de l'imprécision dans l'intégralité du processus de localisation/modélisation.

Dans le quatrième chapitre, à partir des constatations faites sur les résultats du chapitre précédent, nous présentons la théorie ensembliste, qui nous permettra d'obtenir une estimation de la configuration du robot de façon garantie. Après une étude de différents modèles d'erreurs associés à l'observation d'un amer, nous proposons une méthode de localisation utilisant l'algorithme SIVIA développé par L. Jaulin. Nous présentons également des résultats expérimentaux sur deux séries d'acquisitions et dans deux environnements différents. Nous constaterons finalement la validité de la quantification de l'imprécision de la localisation sous forme de sous pavage.

Dans le dernier chapitre, nous étendons l'approche détaillée dans le chapitre au problème de la modélisation des amers verticaux de l'environnement sous forme de sous pavages. L'algorithme de localisation devra être légèrement modifié pour prendre en compte cette particularité. Nous présentons ensuite la méthode de décision utilisée, basée sur la surface de recouvrement entre sous pavages observés et sous pavages de la carte. Cette étape nous obligera à calculer l'image d'un sous pavage en reprenant le concept de l'algorithme ImageSP introduit par M. Kieffer. Nous proposons ensuite un méthode de mise à jour de la carte utilisant des intersections entre pavés, permettant d'optimiser la quantification de l'imprécision sur les amers. Les résultats expérimentaux présentés mettront en avant la validité de cette approche, intégrant l'imprécision à toutes les étapes, tant d'un point de vue robustesse que d'un point de vue qualitatif vis à vis de la réalité terrain.

Nous concluons ce mémoire par un récapitulatif de l'ensemble des solutions proposées tout au long des chapitres, et sur l'analyse qualitative de l'ensemble des résultats obtenus. Finalement nous exposerons les perspectives concernant l'amélioration de ce travail, mais aussi concernant les extensions futur pouvant être données à notre paradigme de modélisation et localisation simultanée.

Chapitre 1

Etat de l'art

Sommaire partiel

1.1.	INTRODUCTION	6
1.2.	LES SYSTEMES DE PERCEPTION	6
1.2.1	<i>Les capteurs proprioceptifs</i>	7
1.2.1.1.	Les capteurs de déplacement	8
1.2.1.2.	Les capteurs d'attitude	8
1.2.2	<i>Les capteurs extéroceptifs</i>	9
1.2.2.1.	Les capteurs télémétriques.....	10
1.2.2.2.	Les systèmes de vision	11
1.3.	LES METHODES DE LOCALISATION	19
1.3.1	<i>Localisation a l'estime</i>	20
1.3.2	<i>Localisation absolue</i>	21
1.3.2.1.	Repères artificiels	21
1.3.2.2.	Repères naturels.....	23
1.3.3	<i>Méthodes hybrides</i>	27
1.3.3.1.	Le recalage dynamique	27
1.3.3.2.	La coopération multi-capteur.....	28
1.4.	LOCALISATION ET MODELISATION SIMULTANEE DE L'ENVIRONNEMENT.....	29
1.4.1	<i>Les approches métriques</i>	30
1.4.1.1.	Les méthodes de modélisation géométriques	30
1.4.1.2.	Les méthodes de modélisation basées sur des grilles d'occupation	31
1.4.2	<i>Les approches topologiques</i>	33
1.4.3	<i>Conclusion sur les méthodes de modélisation</i>	35
1.5.	CONCLUSION	35

1.1. Introduction

La décomposition du problème de la mobilité pour les robots mobiles autonomes amène à définir une architecture classique en robotique, organisée suivant un fonctionnement séquentiel « perception / décision / action ». La prise de décision nécessite donc un conditionnement le plus robuste possible des données sensorielles issues du système de perception. L'étape de décision peut revêtir plusieurs formes suivant la tâche considérée. Pour la tâche de génération incrémentale de carte, elle consistera essentiellement à intégrer de nouvelles primitives ou à fusionner des observations avec des primitives cartographiques déjà existantes. L'étape de localisation se révélera cruciale dans le processus précédent, et se révèle indissociable de la phase de modélisation. C'est à partir de la connaissance de son environnement et des buts qu'il doit atteindre qu'un robot mobile pourra élaborer des lois de commande et pourra prendre des décisions d'action adéquates.

Le paradigme de localisation et de modélisation simultanée de l'environnement que nous souhaitons mettre en œuvre nécessitera donc de s'intéresser aux problématiques de perception, de localisation et bien sûr de génération incrémentale de carte. Le plan de ce chapitre sera donc calqué sur cette organisation et aura pour finalité de situer nos travaux par rapport à un existant dans ces domaines.

Nous commencerons donc par détailler les principaux systèmes de perception utilisés en robotique mobile en nous focalisant sur les systèmes de vision omnidirectionnelle. Ensuite, nous présenterons un panorama non exhaustif des principales méthodes et travaux de localisation. Enfin, nous ferons une synthèse sur les différentes méthodes de modélisation de l'environnement avec une classification qui sera fonction du type de représentation choisi. Nous concluons ce chapitre en situant nos travaux par rapport aux problématiques mises en évidence dans cet état de l'art.

1.2. Les systèmes de perception

La fonction perception consiste globalement à saisir un certain nombre d'informations sensorielles dans le but d'acquérir une connaissance et une compréhension du milieu d'évolution. Comme nous l'avons précisé précédemment, elle est le préalable indispensable aux étapes suivantes qui sont généralement pour un robot mobile les étapes de localisation et de mise à jour de carte de l'environnement (Figure 1-1).

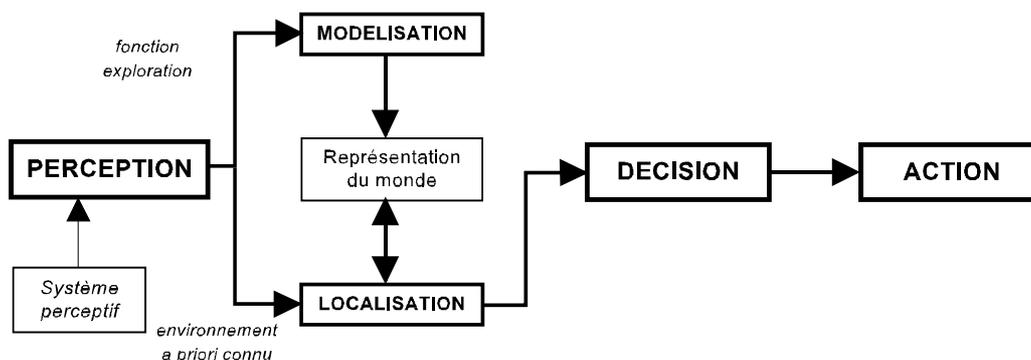


Figure 1-1 : Chaîne fonctionnelle d'un système de navigation.

Plusieurs constats peuvent être faits sur cet organe essentielle de la chaîne fonctionnelle de la navigation. Le premier consiste à noter que le choix d'un système de perception est souvent dépendant du milieu d'évolution du robot mobile ainsi que des fonctionnalités mises en œuvre sur le robot pour qu'il puisse remplir sa mission.

Le second se révèle être générique et consiste à affirmer qu'un système de perception constitué d'un unique capteur sera rarement suffisant pour percevoir correctement l'environnement. Il en va de même pour l'être humain qui, pour appréhender son environnement, sollicite plusieurs organes perceptifs. Le système de perception d'un robot mobile intégrera le plus souvent plusieurs capteurs qui seront de types complémentaires pour un enrichissement des informations sensorielles, ou de types redondants pour répondre au problème de fonctionnement en mode dégradé. Dans ce cadre, des méthodes de fusion de données seront généralement employées pour conditionner ces informations sensorielles.

Enfin le troisième constat qui pourra être dressé est celui du coût de l'intégration de capteurs sur le véhicule autonome. La précision désirée et une fréquence d'acquisition élevée seront autant de facteurs qui augmenteront le coût d'un capteur. Il s'agit donc là d'une contrainte qui pèsera inévitablement sur le choix d'un système de perception.

La classification des capteurs est généralement faite par rapport à deux familles :

- ✓ les capteurs proprioceptifs qui fournissent des informations propres au comportement interne du robot, c'est-à-dire sur son état à un instant donné,
- ✓ les capteurs extéroceptifs qui fournissent des informations sur le monde extérieur au robot.

1.2.1 Les capteurs proprioceptifs

Les capteurs proprioceptifs fournissent, par intégration, des informations élémentaires sur les paramètres cinématiques du robot. Les informations sensorielles gérées dans ce cadre sont généralement des vitesses, des accélérations, des angles de giration, des angles d'attitude.

On peut regrouper les capteurs proprioceptifs en deux familles [FRA90] :

- ✓ les capteurs de déplacement qui comprennent les odomètres, les accéléromètres, les radars Doppler, les mesureurs optiques. Cette catégorie permet de mesurer des déplacements élémentaires, des variations de vitesse ou d'accélération sur des trajectoires rectilignes ou curvilignes.
- ✓ les capteurs d'attitude, qui mesurent deux types de données : les angles de cap et les angles de roulis et de tangage. Ils sont principalement constitués par les gyroscopes et les gyromètres, les capteurs inertiels composites, les inclinomètres, les magnétomètres. Ces capteurs sont en majorité de type inertiels.

1.2.1.1. Les capteurs de déplacement

Les odomètres

Les odomètres permettent de fournir une quantification des déplacements curvilignes du robot en mesurant la rotation de ses roues. Le calcul de la position relative du robot est réalisée par intégration des rotations élémentaires des roues.

Les odomètres sont généralement composés de codeurs incrémentaux qui permettent de mesurer les angles de rotation avec une précision qui dépend de la résolution du codeur. L'information de déplacement nécessitera la connaissance du diamètre des roues, de l'entraxe des roues, de la structure mécanique et cinématique du véhicule.

Ce capteur est fortement utilisé en robotique mobile puisqu'il présente l'avantage d'être simple à mettre en œuvre et surtout d'être peu coûteux.

Les accéléromètres

L'accéléromètre est un capteur qui mesure l'accélération linéaire en un point donné. En pratique, la mesure de l'accélération est effectuée à l'aide d'une masse d'épreuve M , de masse m , reliée à un boîtier du capteur. Le principe de ce capteur est de mesurer l'effort massique non gravitationnel qu'on doit appliquer à M pour le maintenir en place dans le boîtier lorsqu'une accélération est appliquée au boîtier. Le calcul du déplacement élémentaire du robot est obtenu par double intégration de ces informations. Cette double intégration conduit généralement à des accumulations importantes d'erreurs. Ce capteur est plus coûteux que des odomètres.

Le radar Doppler

Le radar Doppler fournit une estimation instantanée de la vitesse linéaire d'une plate-forme mobile par rapport à un objet de la scène en se basant sur l'effet Doppler-Fizeau. Le principe est de diriger un faisceau électromagnétique de fréquence F vers le sol. Le faisceau reçu après rediffusion sur le sol présente un décalage de fréquence ΔF proportionnel à la vitesse V du déplacement relatif du véhicule par rapport au sol. L'intégration de la mesure de vitesse fournira une estimation du déplacement du mobile. Ce capteur est le plus souvent utilisé en milieu extérieur. Il présente le gros avantage d'être insensible aux glissements des roues ou des chenilles du mobile. En revanche, sa précision se dégrade rapidement avec la diminution de la vitesse. De plus il est généralement lourd à mettre en œuvre, tant du point de vue électronique que du point de vue du traitement des informations reçues.

1.2.1.2. Les capteurs d'attitude

Les capteurs d'attitude permettent d'estimer les paramètres intrinsèques du robot que sont les angles de cap, de roulis et de tangage. Ces capteurs sont principalement de type inertiel. Ces capteurs ont pour point commun d'être généralement coûteux et sensibles au bruit, d'où une intégration moins fréquente dans les systèmes embarqués que les odomètres.

Le gyroscope, gyromètre et gyrocompas

Les gyroscopes permettent de mesurer une variation angulaire. Ils sont intéressants en robotique mobile parce qu'ils peuvent compenser les défauts des odomètres. Une erreur d'orientation odométrique peut entraîner une erreur de position cumulative qui peut être diminuée voire compensée par l'utilisation conjointe de gyroscopes [BOR96]. Les gyroscopes très précis sont trop onéreux pour être utilisés en robotique mobile. Cependant, les gyroscopes à fibre optique, connus pour leur grande précision, ont vu leur prix chuter et sont donc devenus une solution attractive pour la navigation en robotique mobile.

Le gyromètre est un capteur qui permet de mesurer une vitesse angulaire. Il existe plusieurs types de gyromètres : les premiers à avoir fait leur apparition furent mécaniques, aujourd'hui, on utilise surtout des gyromètres laser ou des gyromètres optiques [VAG93].

Le gyrocompas est un capteur qui permet de mesurer le cap. Il est composé d'un gyroscope et d'un compas magnétique. Le gyrocompas conserve le nord magnétique durant tout le déplacement du véhicule, après l'avoir initialement déterminé de façon autonome.

Le magnétomètre ou compas magnétique

Le magnétomètre qui est aussi appelé compas magnétique mesure la direction du champ magnétique terrestre pour déduire l'orientation du robot.

Parmi toutes les technologies adoptées pour ce type de capteur, la mieux adaptée pour la robotique mobile est celle dite à vanne de flux. Elle a l'avantage de consommer peu d'énergie, de n'avoir aucune pièce mobile, d'être résistante aux chocs et aux vibrations et d'être peu coûteuse [BOR97]. Toutefois, les mesures sont perturbées par l'environnement magnétique du robot (comme par exemple les lignes d'énergie ou les structures en acier). Ceci rend difficile l'utilisation de ce capteur en milieu intérieur. Néanmoins, ce capteur a par exemple été intégré avec succès dans un système de localisation basé sur le Filtrage de Kalman Étendu [ARN93].

La caractéristique principale de ce capteur est sa précision moyenne qui, sur un long trajet, est relativement bonne [FRA90].

1.2.2 Les capteurs extéroceptifs

Les capteurs extéroceptifs permettent de percevoir le milieu d'évolution du robot. Ils sont généralement le complément indispensable aux capteurs présentés précédemment. Des méthodes de fusion de données seront alors utilisées pour conditionner et traiter les informations sensorielles de natures différents. Deux familles de capteurs extéroceptifs embarqués peuvent être identifiées : les capteurs télémétriques et les systèmes de vision.

1.2.2.1. Les capteurs télémétriques

Les capteurs à ultrasons

Les capteurs à ultrasons utilisent l'air comme milieu de propagation. La méthode de mesure consiste à exciter une céramique piézo-électrique à l'aide de quelques impulsions de fréquence égale à la fréquence de la pastille. Connaissant la vitesse de propagation de l'onde acoustique dans un milieu donné, la mesure du temps de vol de l'onde permet d'obtenir la distance d'un objet par rapport au capteur.

Les capteurs ultrasonores les plus couramment utilisés sont les capteurs Polaroid qui émettent une onde ultrasonore à 44 kHz et qui permettent de mesurer des distances comprises entre trente centimètres et dix mètres.

Les avantages majeurs de ces capteurs sont leur faible prix de revient et leur simplicité d'implantation. En revanche de nombreux inconvénients induisent leur utilisation de moins en moins fréquente en robotique mobile :

- ✓ Une très faible directivité qui est liée au cône d'émission de l'onde dont l'angle d'ouverture est important. Ceci implique une erreur généralement importante qui entache la mesure (Figure 1-2).

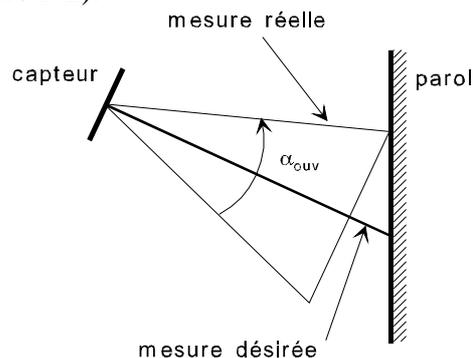


Figure 1-2 : cône d'émission d'un faisceau ultrasonore.

- ✓ Un angle d'incidence relativement faible qui n'excède généralement pas 30 à 40° suivant le matériau de la paroi.
- ✓ Une forte sensibilité aux conditions d'utilisation, telles que la température, l'humidité et les échos parasites. Ces paramètres influent directement sur la précision de mesure.
- ✓ Une forte influence aux problèmes de réflexions multiples. Ce phénomène se produit lorsque l'onde ultrasonore heurte plusieurs parois avant de revenir sur le capteur.

Les capteurs laser

Le capteur laser est basé sur l'émission d'une onde électromagnétique structurée. Ce capteur permet donc d'obtenir un faisceau d'ondes très concentré, contrairement aux capteurs ultrasonores. Les systèmes laser possèdent de nombreux avantages qui en font des capteurs souvent utilisés dans les applications de robotique mobile :

- ✓ la résolution angulaire : des faisceaux angulairement très fins peuvent être obtenus avec des lentilles d'émission de petite taille, du fait de la courte longueur d'onde émise.
- ✓ la réponse optique des cibles : les longueurs d'ondes courtes permettent d'obtenir un écho pour des angles d'incidence du faisceau sur la cible allant jusqu'à 75°, et améliore de ce fait la probabilité de détection.
- ✓ la précision sur la mesure de distance par rapport à la portée, pour les télémètres, est importante. Une précision de l'ordre du dixième de millimètre peut être obtenue sur des distances de l'ordre de 30 mètres.

L'inconvénient majeur du laser se situe au niveau de l'électronique de mesure qui doit être capable, étant donnée la vitesse de propagation des ondes électromagnétiques, de mesurer des intervalles de l'ordre de la pS pour des précisions de distance de l'ordre du millimètre. Ceci implique un matériel coûteux et complexe.

En robotique les systèmes laser sont utilisés principalement de trois manières différentes :

- ✓ en télémètre : ils sont alors utilisés, tout comme les capteurs à ultrason, pour des mesures de distance.
- ✓ en goniomètre : ils sont utilisés pour des mesures d'angles [HOL91].
- ✓ en source de lumière structurée : ils sont généralement utilisés pour la modélisation 3D de l'environnement. La source laser sera dans ce cas toujours associée à une caméra et fournira des informations sur la profondeur [KEM93].

Un exemple d'intégration de ce capteur laser sur un robot est la plate-forme mobile HILARE II du LAAS [FIL94].

1.2.2.2. Les systèmes de vision

Les systèmes de vision en robotique sont basés sur l'utilisation d'une caméra CCD. L'arrivée des capteurs CCD (Charge Coupled Device), en 1975, a été déterminante dans l'évolution de la vision : la rapidité d'acquisition, la robustesse et la miniaturisation sont autant d'avantages qui ont facilité leur intégration. Les systèmes de vision sont très performants en termes de portée, précision et quantité d'informations exploitables. Ils sont de plus les seuls capables de restituer une image sensorielle de l'environnement la plus proche de celle perçue par l'être humain.

En revanche, l'inconvénient majeur de tels systèmes de perception se situe au niveau de la gestion du flux important de données exploitables : traiter une image demeure une opération délicate et surtout coûteuse en temps de calcul.

Utilisé seul, un caméra CCD ne peut fournir qu'une information 2D. Les techniques qui vont permettre d'obtenir des informations 3D à partir d'un tel capteur sont généralement liées à l'adjonction d'un autre capteur. Dans ce cadre nous pouvons identifier les techniques suivantes :

- ✓ la stéréovision
- ✓ les techniques de lumières structurées

On peut ajouter aux deux catégories précédentes celle consistant à associer des systèmes de réflexion de type miroir au capteur CCD : il s'agit des techniques de vision omnidirectionnelle. Précisons que cette dernière sera bien sûr utilisée pour obtenir un modèle sensorielle panoramique de l'environnement.

Nous présentons dans la suite de cette étude les trois techniques précédente insistant plus particulièrement sur la vision omnidirectionnelle qui concerne directement ces travaux de thèse.

La stéréovision

La stéréovision consiste à observer une même scène avec deux caméras qui sont éloignées l'une de l'autre et dont on connaît la distance qui les séparent. Connaissant la géométrie exacte du système stéréoscopique la première étape de reconstruction 3D consiste à mettre en correspondance les deux images. Cette phase réside dans la détermination de couples de points observés dans les deux images, ou dans l'appariement de points d'intérêt. L'information 3D pourra alors être fournie par triangulation.

La stéréovision est basée sur le même principe de reconstitution de la profondeur que la vision chez l'être humain.

La Figure 1-3 décrit plus précisément le principe de la stéréovision. Tout point M visible depuis les deux caméras se projette en P_1 sur le plan image de la première caméra et en P_2 sur le plan image de la deuxième caméra. Inversement, connaissant deux points en correspondance P_1 et P_2 , ainsi que la distance D séparant les centres optiques O_1 et O_2 des deux caméras, les coordonnées du point P sont données par l'intersection de (O_1P_1) et (O_2P_2) .

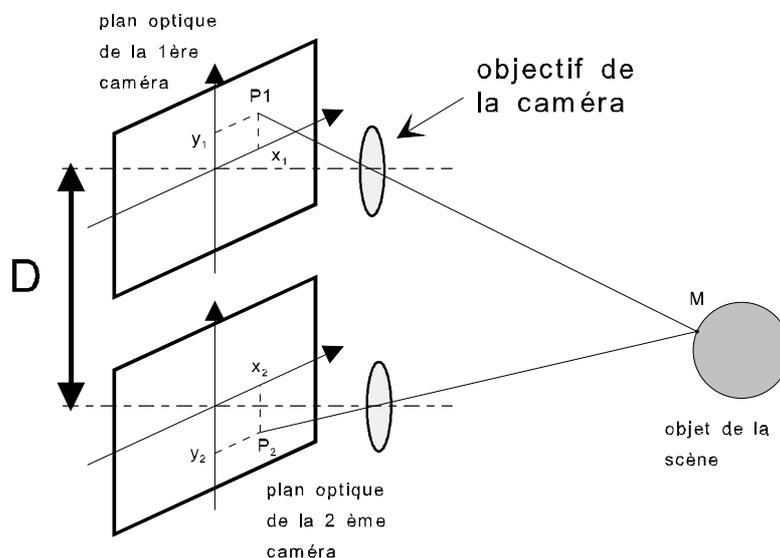


Figure 1-3 : principe de la stéréovision.

La stéréovision apparaît comme un des moyens de perception les plus performants en robotique. Toute la problématique de la stéréovision réside dans la robustesse de la phase de mise en correspondance des informations : elle est souvent liée à de nombreuses ambiguïtés mais aussi à des temps de calcul très importants (aspect fortement combinatoire de l'appariement).

Le principe de stéréovision a par exemple été utilisé par Olson pour le très médiatique robot *Rocky7* destiné à être envoyé sur la planète Mars [OLS98]. Le robot ne pouvant posséder une carte préétablie de son environnement extra-terrestre d'évolution, celle-ci est construite en utilisant la stéréovision. *Rocky7* se sert ensuite de cette carte globale pour appairer la carte locale, obtenue elle aussi par stéréovision, en utilisant une mesure de similarité basée sur la distance de Hausdorff. La stéréovision est utilisée dans de nombreuses autres applications robotique [SAI86] [TAK95] [KRO95].

Goshtasby et Grover dans [GOS93], et plus récemment Gluckman et Nayar dans [GLU99] ont placé deux miroirs plans pour calculer la profondeur d'une scène. L'utilisation d'une caméra et de deux miroirs plans est équivalente à la stéréoscopie avec deux caméras. Elle présente en outre l'avantage que les données stéréoscopiques proviennent de la même caméra. Une approche similaire, basée sur l'utilisation de deux miroirs plans, a été proposée dans [SOU96].

Précisons à ce niveau que cette technique pourra bien sûr être appliquée à la vision omnidirectionnelle que nous détaillerons dans un des paragraphes suivant.

Les capteurs à triangulation active

L'obtention de la 3D est gérée comme pour la stéréovision, mais on remplace une caméra par un laser qui sera chargé de projeter un motif de lumière structurée. Le principe est d'éclairer l'environnement avec une lumière cohérente et de l'observer avec une caméra sous un certain parallaxe. Connaissant la position de la source lumineuse et celle de la caméra, l'information peut être obtenue par triangulation.

Ces capteurs sont utilisés pour obtenir des images de profondeur, mais également pour l'évitement d'obstacles [OHY94][FOF97], pour la numérisation d'objet 3D [PAP97].

Plusieurs patrons de lumière structurée peuvent être utilisés :

- ✓ *Capteur à simple triangulation* : les capteurs à simple triangulation acquièrent la distance d'un seul point lumineux à la fois. Une application célèbre est celle du véhicule d'exploration Pathfinder destiné à être envoyé sur la planète Mars [MAT95a].
- ✓ *Les capteurs par plan de lumière* : la source de lumière projetée n'est plus un point, mais un plan de lumière. Un système de localisation basé sur ce principe a été développé dans [JOU01].
- ✓ *Capteurs à plusieurs plans de lumière* : il s'agit dans ce cas de projeter un motif tel que des lignes ou de petits disques sur la scène observée (Figure 1-4) [MUN87].

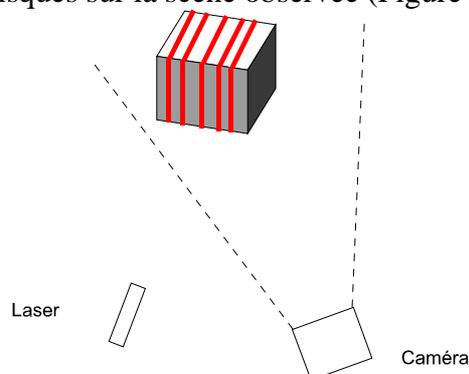


Figure 1-4 : triangulation par plusieurs plans laser.

La technique de triangulation active est plus robuste que la stéréovision car elle permet de faire l'économie de l'étape de mise en correspondance. En revanche la précision n'est pas uniforme sur l'ensemble de l'image : plus un objet est éloigné, plus l'erreur de mesure est importante [FOF97].

Les systèmes de vision omnidirectionnelle

Ces systèmes de perception associent une caméra CCD et un élément permettant d'obtenir une vue sur 360 degrés de l'environnement. A ce titre ils sont de plus en plus utilisés en robotique mobile au détriment de la vision monoculaire. Suivant l'élément ou les éléments ajoutés à une caméra CCD, on pourra distinguer quatre techniques pour obtenir une image omnidirectionnelle.

□ **Génération d'images multiples par utilisation de plusieurs caméras**

Cette première technique consiste à utiliser plusieurs caméras couvrant un champ de vision égal à 360 degrés : par exemple quatre caméras séparées par des angles de 90 degrés.

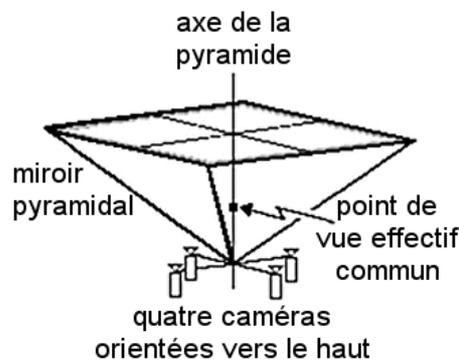


Figure 1-5 : Système Fullview[®] des laboratoires Bell.

Dans ce cadre, nous pouvons citer le système développé au laboratoire de Bell. AT & T qui intègre quatre caméras conventionnelles et quatre miroirs triangulaires [NAL96]. On dispose donc de quatre images pour pouvoir calculer une image panoramique. Le résultat est une image panoramique à haute résolution (3800 par 480 pixels). Ce capteur possédant un seul point de vue effectif, il permet de calculer n'importe quelle projection planaire pour obtenir des images purement perspectives.

Un autre exemple de système est celui présenté dans [KAW98] : le capteur génère une image omnidirectionnelle grâce à un système composé de six caméras et d'un miroir hexagonal.

Le problème de cette technique réside d'une part dans les angles morts, c'est à dire les parties de l'environnement qui ne sont pas vues à un instant donné, et d'autre part dans les zones de l'environnement qui sont sur deux images à la fois. Dans ce dernier cas, il faut ajouter un traitement des intersections qui augmente le temps de calcul.

□ **Génération d'images multiples par rotation d'une caméra**

Cette deuxième technique consiste à faire pivoter une caméra autour d'un axe [SAR89] [ISH92]. Cette méthode permet d'avoir une prise de l'environnement avec une seule caméra, mais il faut effectuer une rotation complète avant d'obtenir une vue sur 360 degrés.

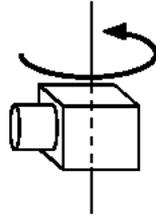


Figure 1-6 : caméra rotative.

La technique de mosaïque peut-être utilisée pour représenter une séquence vidéo, pour construire des panoramas lors d'un déplacement de la caméra [ZHE90] et ainsi mémoriser le déplacement, ou bien pour la génération d'image omnidirectionnelle.

De tels systèmes génèrent un volume de données qui nécessite des temps de traitement très important. En outre, les aspects mécaniques, la gestion du flou dû au mouvement de la caméra font que ces méthodes sont complexes à mettre en œuvre. Une alternative possible est l'utilisation d'une caméra linéaire comme le fait Barth dans [BAR96]. Toutefois, quelle que soit la solution envisagée, l'acquisition d'une scène panoramique avec une caméra matricielle ou linéaire reste une méthode lourde à mettre en œuvre.

□ Utilisation de lentille spéciale

Cette troisième méthode consiste à ajouter un ensemble optique sur la caméra CCD afin de dévier les rayons lumineux, comme par exemple l'œil de poisson (fish-eye) [CAO86].

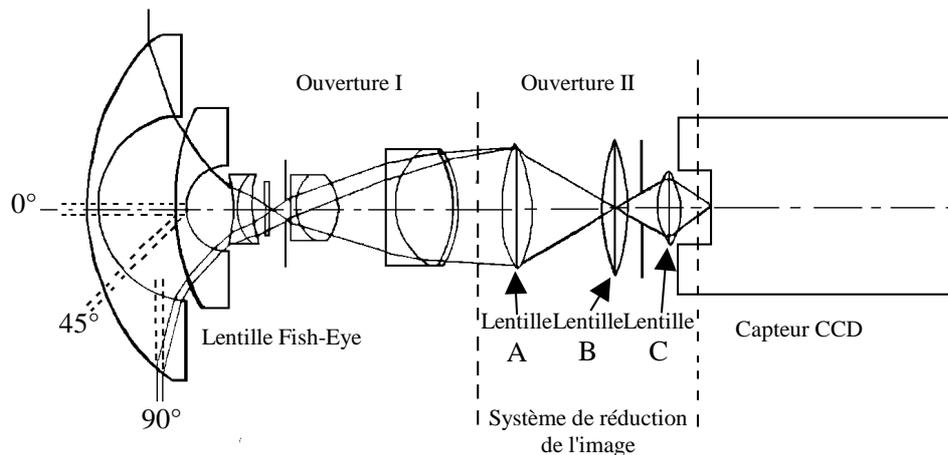


Figure 1-7 : Schéma du système optique du capteur omnidirectionnel utilisé par Cao.

Une caméra Fish-eye, qui utilise un objectif Fish-eye au lieu d'un objectif conventionnel, peut acquérir une vue hémisphérique en temps réel. Morita a proposé une méthode stéréoscopique de l'analyse du mouvement basée sur cette technique [MOR89]. Les chercheurs de l'université de Cincinnati ont appliqué une caméra Fish-eye au contrôle de position d'un robot mobile en utilisant des cibles [CAO86][RON86] et en suivant des lignes [ELK94][MAT95b].

L'analyse de ces images est difficile car les éléments utiles se situent généralement sur la périphérie, là où la résolution est faible. De plus, il est difficile de générer une image complètement perspective car les objectifs Fish-eye n'ont pas un centre de projection unique. L'image obtenue comporte beaucoup de distorsion et des méthodes de calibration pour de tels systèmes optiques sont nécessaires. Toutefois, l'ajout de mécanismes optiques complique le modèle mathématique et augmente la distorsion déjà importante pour un système de vision.

□ Utilisation de miroirs convexes

La dernière méthode consiste à placer un miroir réfléchissant face à la caméra pour obtenir une image qui est la projection omnidirectionnelle 2D de l'environnement. Ces capteurs omnidirectionnels sont appelés capteurs catadioptriques. Ils présentent l'avantage de fournir une vue globale de l'environnement en une seule acquisition. Autre avantage : ils n'ont aucune partie mobile et donc consomment peu d'énergie (pas de mécanisme de rotation et d'orientation). Par contre, le redressement d'image pose problème. Rappelons que la catadioptrie est la théorie des systèmes optiques comportant une succession d'éléments réfléchissants et réfractants (dioptries : lentilles...). Dans les systèmes étudiés, il s'agit d'un miroir de révolution convexe associé à une caméra CCD. La classification que nous avons adoptée pour la présentation de ces systèmes reprend les grandes lignes de [PER96][MAY97][BAK98]. Nous verrons à travers cette description qu'un point important qui caractérise un système de vision omnidirectionnelle est la possibilité de gérer un unique point de vue (centre de projection). Précisons que cette contrainte est importante, car vérifiée, elle permet la reconstruction d'images perspectives géométriquement correctes par la fonction plénoptique calculée au centre de projection effectif [ADE91].

Les miroirs utilisés pour la vision omnidirectionnelle sont de plusieurs types :

- ✓ miroir conique
- ✓ miroir sphérique
- ✓ miroir hyperboloïdal
- ✓ miroir paraboïde

➤ Miroir conique

Dès 1990, Yagi a utilisé le système de perception basé sur un miroir conique COPIS (COnic Projection Image Sensor) [YAG94] (Figure 1-8). Les angles de gisement des balises naturelles observées sont extraits à partir de droites radiales. Une prédiction des angles d'azimut est effectuée lors du déplacement du robot par odométrie [YAG95]. La carte de l'environnement étant a priori connue, l'algorithme de localisation consiste à mettre en correspondance cette prédiction avec les angles des droites radiales. La prédiction ainsi effectuée permet de réduire le temps de traitement lié à l'appariement. Le calcul de la configuration est ensuite réalisé à l'aide de la méthode des moindres carrés. Les possibilités du système ont ensuite été étendues pour effectuer de l'évitement d'obstacles et de la reconstruction d'environnement [YAG94] [YAG95].

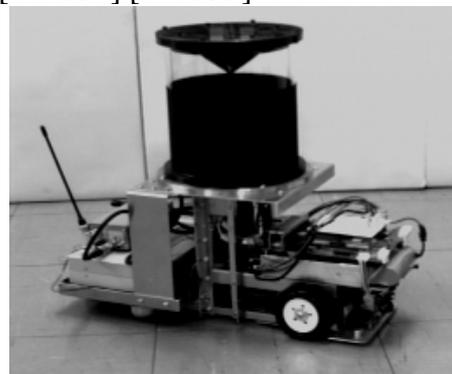
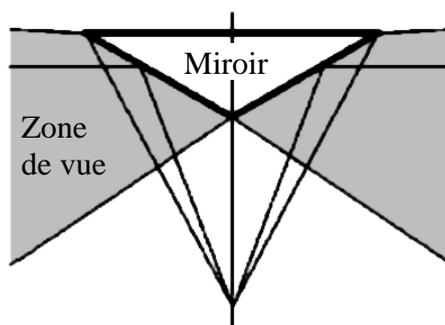


Figure 1-8 : Champ de vision du miroir conique et le système COPIS.

A peu près parallèlement, notre laboratoire a également développé un système de perception panoramique baptisé SYCLOP (SYstème Conique pour la LOcalisation et la Perception) [PEG96]. Ce capteur sera décrit plus en détail dans le chapitre suivant.

Laurent Delahoche, dans le cadre de ses travaux de thèse, a utilisé ce capteur en association avec un télémètre laser pour localiser un robot de manière absolue en utilisant uniquement les amers naturels de l'environnement [DEL97a]. Ces travaux ont été étendu à la localisation dynamique par filtrage de Kalman Etendu [DEL97b] et à la modélisation incrémentale de l'environnement [DEL98]. Bruno Marhic a utilisé ce capteur seul pour estimer la position d'un robot mobile à l'aide de invariants [MAR99].

Toujours par rapport à un aspect national, nous pouvons citer les travaux réalisés à l'université de Besançon, qui ont eu pour finalité de développer un système de navigation, STAN (Système de transport à apprentissage neuronal), basé sur la vision omnidirectionnelle stéréoscopique [BID95]. Le capteur est composé d'une caméra CCD et de deux miroirs coniques. Un système de déflexion permet d'obtenir une superposition des deux zones exploitables (anneaux) de mesures sur le plan image de la caméra.

Les travaux présentés dans cette thèse utilise un tel système de vision pour résoudre le problème de la construction incrémentale de l'environnement avec localisation simultanée.

Un inconvénient sérieux du système de vision conique est qu'il n'a pas un point focal unique (un centre de point de vue). Par conséquent, il est impossible de produire une image perspective sans déformation à partir d'une image omnidirectionnelle.

➤ Miroir sphérique

Une image obtenue à partir d'un miroir sphérique a une résolution qui est bonne dans la région centrale mais la résolution est faible dans la région périphérique. Il s'agit là d'une caractéristique commune avec les objectifs Fish-eye. La zone de vision de ce type de capteur est la plus large parmi tous ceux utilisant des miroirs convexes [BOG95] (Figure 1-9).

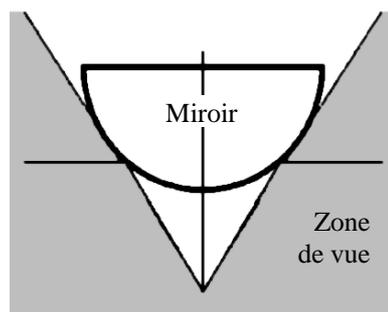


Figure 1-9 : Champ de vision du miroir sphérique.

Hong [HON91] a utilisé un capteur de vision omnidirectionnelle composé d'un miroir sphérique. Ce capteur est utilisé à des fins de navigation. L'idée est de détecter des changements d'apparence d'amers caractéristiques de l'horizon.

Un miroir sphérique est également utilisé par Winters [WIN99] pour déterminer la position du robot. Le robot possède une base de données d'images omnidirectionnelles qui est comparée à l'image qu'il vient d'acquérir.

Enfin, nous pouvons citer les travaux de Matsui [MAT00] qui utilise ce type de capteur monté à l'avant du robot. L'image obtenue est dépliée et une mesure de corrélation est effectuée sur une base d'images de référence pour en déduire la localisation du robot.

➤ **Miroir hyperboloïdal**

Il s'agit d'un miroir dont le profil est une hyperbole. Yamazawa et Yagi ont utilisé les premiers un capteur de ce type : le capteur HyperOmni Vision [YAM93] (Figure 1-10).

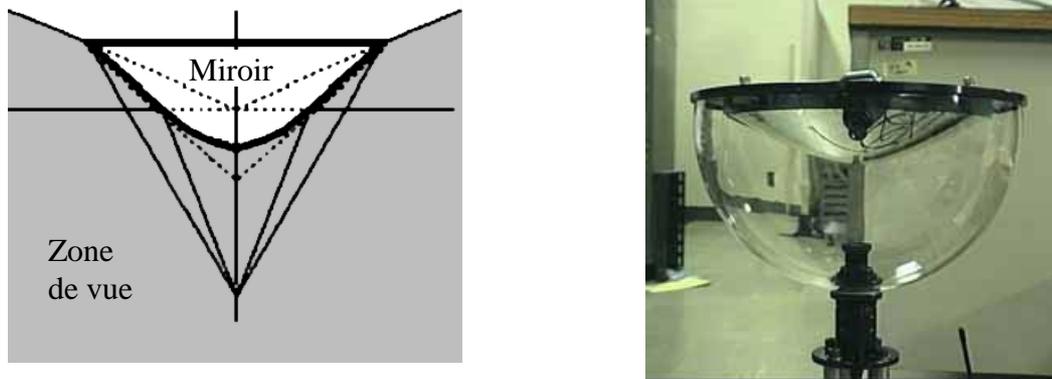


Figure 1-10 : Un prototype du capteur HyperOmni Vision.

Le miroir hyperboloïde a deux points focaux. Si le centre optique de la caméra est placé sur l'un de ces deux points focaux, il est possible de générer une image plane à partir d'une image omnidirectionnelle. Cette génération d'images perspectives ou d'images panoramiques permet au robot d'utiliser des méthodes de vision existantes pour la navigation et la manipulation.

Ishiguro, de l'université de Kyoto, a conçu une famille de capteurs omnidirectionnels, à faible coût et relativement compacts, utilisant des miroirs sphériques et hyperboliques [ISH98]. Les réflexions internes causées par le support en verre cylindrique sont supprimées par l'adjonction d'un "dard" central noir mat.

Enfin, nous pouvons citer les travaux de Svoboda qui estime la vitesse d'un robot à l'aide d'un miroir hyperboloïdal [SVO98b].

➤ **Miroir paraboloid**

L'utilisation du miroir paraboloid en robotique n'est pas généralisé. Une des rares études intégrant ce type de miroir concerne les travaux de Péri et Nayar [NAY97] qui ont réalisé un système de visioconférence nommé OmniVideo.

➤ **La stéréovision omnidirectionnelle**

La stéréovision omnidirectionnelle permet d'obtenir une information de profondeur sur un champs de vision de 360 degrés. Il s'agit d'une technique très intéressante d'un point de vue richesse et robuste de l'information sensorielle. Cette approche est directement liée à la méthodologie proposée dans ce mémoire.

Ishiguro et Tsuji ont proposé un système stéréo omnidirectionnel qui utilise deux vues panoramiques [ISH90] [ISH92]. Les deux images panoramiques sont obtenues à l'aide de deux caméras en rotation. Les axes optiques de chaque caméra passent par l'axe de rotation. La profondeur d'un objet est estimée à partir de ses projections dans les deux images prises à deux positions différentes. Ce système est extrêmement lourd d'un point de vue temps de calcul ce qui rend son utilisation impossible pour des applications en temps réel. Murray a généralisé cette approche pour le calcul de la profondeur et du mouvement dans des séquences d'images [MUR95]. Kang et Szeliski font une reconstruction tridimensionnelle de la scène à l'aide d'images panoramiques [KAN97].

Sur le plan national, Benosman, Maniere et Devars ont proposé un autre type de système panoramique en utilisant deux caméras linéaires [BEN96]. Les deux caméras sont positionnées l'une au dessus de l'autre mais cette fois, leur centre optique est confondu avec l'axe de rotation. Un point situé sur une ligne verticale d'une des deux images panoramiques se trouve sur la même ligne verticale dans l'autre image. Pour cette raison, ils ramènent le problème de mise en correspondance 2D à un problème à une dimension.

L'utilisation de plusieurs miroirs convexes s'est aussi fortement généralisée. Nene et Nayar ont développé une série de capteurs avec une caméra et deux miroirs [NEN98]. Ils ont montré que la conception de système stéréoscopique à partir de deux miroirs et d'une caméra peut être faite à partir de n'importe quelle combinaison caméra-miroir en préservant la propriété de point de vue unique. Une expérimentation avec d'une part une combinaison caméra perspective - miroirs plans et d'autre part une combinaison caméra orthographique - miroir parabolique est présentée.

La stéréovision basée sur l'utilisation conjointe de deux lentilles de type fish-eye est proposé par Shah et Aggarwal dans [SHA97]. L'image « stéréo fish-eye » permet au robot de se localiser dans un environnement structuré intérieur. L'estimations de position du robot est basée sur le paramétrage 3D des lignes horizontales de l'environnement.

Un point important en stéréoscopie omnidirectionnelle est la possibilité de généraliser la géométrie épipolaire classiquement utilisée en vision binoculaire. La contrainte épipolaire permet de restreindre le problème de recherche dans le plan à une recherche sur une droite. Plusieurs auteurs ont étudié la géométrie épipolaire pour des systèmes stéréoscopiques omnidirectionnels [SVO98a][SMA01]. Dans [MOR00] les auteurs proposent une projection cylindrique à géométrie épipolaire. Dans [SVO98b] Svoboda propose quant à lui une estimation du mouvement basée sur ce principe.

Enfin, sur le plan national, nous pouvons citer le projet STAN de l'université de Besançon [BID95], déjà discutés précédemment. Ce système de vision stéréoscopique permet par apprentissage neuronal de localiser un robot dans un milieu de type industriel.

1.3. Les méthodes de localisation

Nous avons précédemment positionné le problématique de localisation comme étant un préalable incontournable de celle de la génération de carte incrémentale de l'environnement. La suite logique de la synthèse faite sur les capteurs portera donc sur les différentes méthodes de localisation. Par analogie avec la classification des capteurs, nous pourrions distinguer deux types de méthodes de localisation :

- ✓ Les méthodes de localisation relatives, basées sur l'utilisation des capteurs proprioceptifs,
- ✓ Les méthodes de localisation absolues, basées sur l'utilisation de capteurs extéroceptifs,
- ✓ Les méthodes dites « hybrides » qui sont basées sur l'utilisation conjointe des deux types de capteur.

1.3.1 Localisation à l'estime

La navigation à l'estime, ou navigation relative, consiste à évaluer la position, l'orientation, et éventuellement la vitesse du robot mobile par intégration des informations fournies par des capteurs dits proprioceptifs. L'intégration se fait par rapport au point de départ du robot. Ces données peuvent être des informations de déplacement (odomètre), de vitesse (vélocimétrie) ou d'accélération (accéléromètre). Ces systèmes permettent d'obtenir un flux relativement important au niveau de l'obtention des estimations de position. Cette caractéristique intéressante a favorisé l'utilisation de ces systèmes de localisation en robotique mobile, ainsi que dans des secteurs de pointe tels que les domaines spatial et militaire : fusées, missiles, avions, sous-marins... Il n'y a que peu de méthodes basées sur l'utilisation unique de capteurs proprioceptifs. Ainsi dans la suite du paragraphe, nous ne fournissons qu'une explication succincte de ces techniques.

L'odométrie permet de déterminer la position et le cap (x,y,θ) d'un véhicule par intégration de ses déplacements élémentaires, et ce, par rapport à un repère lié à sa configuration initiale. L'algorithme de localisation est basé sur le comptage des impulsions générées par des codeurs durant une période d'échantillonnage connue. Connaissant le rayon des roues et la distance entre chaque roue, il est possible d'exprimer de manière récursive la position et le cap du robot. Les avantages de l'odométrie résident dans sa simplicité de mise en œuvre et dans son faible coût : ces caractéristiques en font un système de localisation couramment utilisé en robotique mobile. En outre, l'odométrie permet d'obtenir une estimation de la position et du cap à une cadence relativement élevée. L'inconvénient est une précision très médiocre sur des distances importantes, ce à cause des erreurs cumulatives. Une étude intéressante d'intégration poussée de l'odométrie a été réalisée par Borenstein dans [BOR95]. Il y propose une méthode originale IPEC (Internal Position Error Correction), permettant d'obtenir une estimation odométrique fiable sans avoir recours à un système de localisation absolue ou à un système de navigation inertielle.

La vélocimétrie est une méthode qui consiste à mesurer directement la vitesse du véhicule et à l'intégrer pour obtenir le déplacement. La vitesse de rotation instantanée, dans la majorité des cas, est obtenue avec des gyromètres. La vitesse linéaire est quant à elle généralement obtenue par l'utilisation d'un radar Doppler dirigé vers le sol. Dans ce cadre nous pouvons citer ici les travaux de H. Makela [MAK91] qui localise à l'estime un véhicule circulant sur des chemins forestiers à l'aide d'un radar Doppler.

Ce type de navigation utilise les accéléromètres, les gyroscopes et les compas magnétiques. La localisation relative d'un robot est déterminée à partir d'informations inertielles acquises au cours de son mouvement. Les méthodes de localisation inertielles utilisent les informations d'accélération suivant des axes prédéterminés ou alors l'effet gyroscopique par rapport à des axes de rotations prédéfinis. Le calcul de la position est effectué par double intégration de ces informations. Le robot VEGA du LIRMM utilise pour se localiser un odomètre et un système inertielle composé de six capteurs : un accéléromètre 3 axes, 4 gyromètres piézo-électriques et un gyromètre à fibre optique [VAG93].

1.3.2 Localisation absolue

La localisation absolue est une technique qui permet à un robot de se repérer directement dans son milieu d'évolution, que ce soit en environnement extérieur (mer, espace, terre), ou en environnement intérieur (ateliers, immeubles, centrales nucléaires...). Ces méthodes de localisation sont basées sur l'utilisation de capteurs extéroceptifs.

Pour répondre à la problématique qu'est la localisation d'un robot dans son environnement, deux types de stratégies sont utilisables :

- ✓ la première consiste à utiliser des points de repère naturels,
- ✓ la deuxième à utiliser des points de repère artificiels.

Il est à noter que quelque soit le cas de figure, la localisation absolue nécessite toujours une représentation de l'environnement. Le robot possède donc «une banque de données» regroupant les caractéristiques des références externes qui est appelée carte de l'environnement.

1.3.2.1. Repères artificiels

Les repères artificielles sont des balises caractéristiques qui sont ajoutées au milieu d'évolution du robot et dont les positions sont connues. L'inconvénient de ce type de techniques réside essentiellement dans son manque de souplesse et dans sa lourdeur d'utilisation. En effet un domaine d'évolution vaste nécessitera un investissement lourd en équipement. En outre tout changement de configuration de l'environnement impliquera une remise en cause du réseau de balises. En revanche cette technique a le gros avantage d'être précise, robuste et surtout de satisfaire la contrainte temps réelle. Précisons à ce niveau que cette approche ne pourra répondre à la problématique de modélisation incrémentale et localisation simultanée.

Les balises artificielles peuvent être de deux types :

- ✓ Actives : elles émettent des signaux,
- ✓ Passives : elles ne peuvent pas émettre.

Balises actives et GPS

Les balises actives sont généralement interactives avec le système de perception de l'engin mobile puisque ce sont elles qui émettent l'onde captée par l'équipement de mesure. Les deux principaux types d'organes de transmission utilisés sont les émetteurs de sources lumineuses (infrarouge par exemple) et les antennes émettrices hyperfréquences. Ces ondes peuvent être émises en continu ou être déclenchées par ordre provenant de l'équipement embarqué sur le mobile. Les systèmes à balises actives sont le moyen de navigation le plus communément utilisé pour les bateaux et les avions.

Au sein de notre laboratoire, un système de localisation absolue basé sur le repérage de balises actives infrarouges par un système de vision monoculaire a été développé et implanté [BRA95]. Les balises artificielles sont constituées par trois diodes électroluminescentes émettant des impulsions infrarouges codées (Figure 1-11). La méthode de localisation consiste à extraire à partir de l'image d'une balise les distances des trois balises par rapport au robot et pour ensuite effectuer une triangulation.

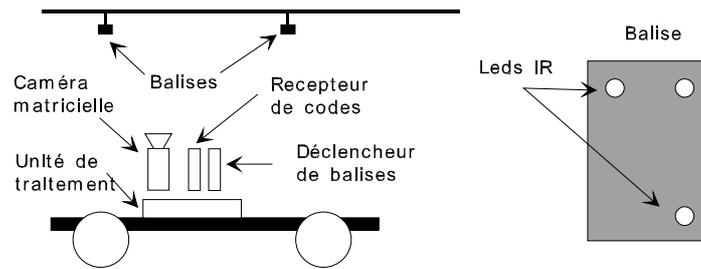


Figure 1-11 : système de localisation par repérage de balises actives.

Le système GPS est finalement un système de localisation en environnement extérieur basé sur le repérage de balises actives que sont les satellites. Ce système, développé par le Département de la Défense des Etats-Unis, est composé de vingt-quatre satellites. En utilisant des méthodes avancées de trilatération, le récepteur terrestre peut calculer une position en mesurant le temps de vol des ondes renvoyées par les satellites. En théorie, trois satellites peuvent permettre de calculer la latitude, la longitude et l'altitude du récepteur.

Pour empêcher l'utilisation du GPS par des nations ennemies, le gouvernement américain a appliqué délibérément une dégradation sur les signaux satellites, de sorte que la précision n'est que de cent mètres. Toutefois, ces effets pouvaient être éliminés en utilisant le DGPS (Differential Global Positioning System). Ce système utilise un deuxième récepteur terrestre fixe situé à proximité du premier (moins de dix kilomètres). La position de ce deuxième récepteur calculée à partir du GPS est comparée à sa position réelle sur le terrain pour générer un vecteur d'erreur permettant la correction de la position du premier récepteur. On obtenait ainsi une précision de quatre à six mètres [BOR97]. Depuis mai 2000, le gouvernement américain a supprimé ce système de cryptage.

L'inconvénient du GPS se situe au niveau des signaux satellites qui peuvent ne pas parvenir au récepteur à cause du relief du terrain et de l'environnement (hauts bâtiments et ponts en villes, relief montagneux, etc.). En solution à cette contrainte, on peut citer les travaux de Cui pour localiser un véhicule dans un environnement urbain [CUI01]. De nombreux systèmes de navigation basés sur l'utilisation du GPS seul ou en couplage avec d'autres capteurs ont été développés [KOT98][SUK98].

Balises passives

On appelle balises passives des repères artificiels n'échangeant pas de signaux avec le système de perception. Comme précédemment ce type de méthode est basé sur le repérage de balises artificielles placées à des positions connues dans le milieu d'évolution du robot. La différence se situe au niveau de l'identification de la balise qui doit être faite par le système de perception. Ainsi le problème de la mise en correspondance est généralement facilité par un codage au niveau du marquage de la balise. Ces systèmes ont l'avantage d'être moins lourds à installer que les systèmes précédents.

Les véhicules mobiles industriels AGV (Autonomous Guided Vehicle), développés par la société GEC-Caterpillar utilisent un système de navigation goniométrique constitué par un laser tournant et des balises passives rétro réfléchissantes [HOL91]. La position du chariot est ensuite déterminée par triangulation. La précision obtenue sur l'estimation de position est excellente puisqu'avoisinant les 2 cm.

Durrant-Whyte décrit dans [DUR95] l'architecture d'un système de navigation pour un véhicule mobile AGV (Autonomous Guided Vehicle) devant transporter automatiquement

des containers sur un quai d'embarquement. Le système de navigation embarqué est basé sur l'utilisation d'un radar à ondes millimétriques permettant de mesurer, par rapport au mobile, la distance et l'orientation des balises. Ces balises sont des trièdres et fixées à des positions connues. L'algorithme de navigation est un filtre de Kalman étendu (EKF). La précision de ce système de navigation, par rapport à la taille du véhicule, est excellente puisqu'avoisinant les 6 cm.

Armingol [ARM98] utilise comme balise artificielle des ronds noirs collés sur les murs d'un environnement intérieur. Ces ronds sont détectés à l'aide d'une caméra CCD orientable. La mise en correspondance avec une image modèle est réalisée grâce à l'étude de la corrélation des niveaux de gris. La position est déterminée à l'aide d'un filtre de Kalman étendu.

Kortenkamp [KOR94a] utilise des cylindres avec différents motifs disposés dans l'environnement pour localiser le robot Carmel (Figure 1-12).

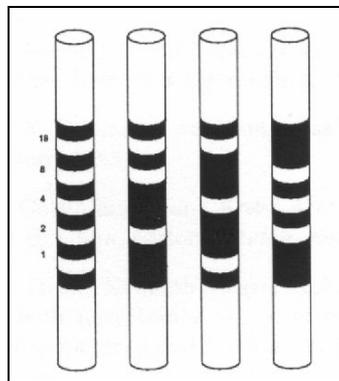


Figure 1-12 : Balises passives utilisées par Kortenkamp [KOR94a].

Il est à noter que le point commun de toutes ces méthodes est le calcul de la configuration du robot basée systématiquement sur une des deux méthodes :

- ✓ la trilatération (triangulation par télémétrie)
- ✓ la triangulation (triangulation par goniométrie)

Enfin, la précision obtenue sur la localisation et l'adéquation avec la contrainte temps réel font que cette technique est souvent intégrée pour des systèmes de convoyage industriels.

1.3.2.2. Repères naturels

Cette technique consiste à utiliser les éléments caractéristiques de l'environnement pour estimer la position du robot. L'intérêt de ces méthodes est donc sa souplesse d'utilisation puisqu'elles ne nécessitent pas d'aménager le milieu d'évolution du robot. Pour la problématique localisation stricto sensu, une connaissance de l'environnement sera nécessaire. Il s'agira d'une représentation cartographique qui intégrera la position des amers qui serviront à localiser le robot. Suivant le niveau sémantique adopté pour décrire l'environnement, plusieurs types de représentations cartographiques pourront être gérés. Ces types seront détaillés dans le paragraphe suivant qui concerne la génération incrémentale de carte. La fonction localisation peut se décliner en deux étapes :

- ✓ la construction du modèle sensoriel : elle consiste à conditionner les mesures capteur.

- ✓ la mise en correspondance de ce modèle avec la carte de l'environnement : elle consiste à apparier les observations avec des primitives cartographique. Il s'agit là de l'étape prépondérante permettant d'estimer la configuration du robot.

La mise en correspondance

La robustesse de la phase de localisation dépend de celle de mise en correspondance. Elle consiste à apparier les observations issues du ou des capteurs avec les éléments de la carte de l'environnement. En raison de son aspect combinatoire, c'est un problème complexe à résoudre et généralement coûteux en temps de calcul. On cherche donc à mettre en correspondance des primitives singulières ou bien des groupements de primitives (modèles) de l'environnement avec une image de celui-ci. Dans ce cadre, le terme "image" est à considérer au sens large ; c'est une représentation sensorielle de l'environnement issue d'un capteur.

□ ***Les méthodes de prédiction et vérification***

Ces méthodes consistent dans un premier temps à générer des hypothèses d'appariement, ce par rapport à un historique sur les observations ou par rapport à des contraintes relatives à l'environnement. Dans un deuxième temps, ces hypothèses sont infirmées ou confirmées par une vérification des hypothèses par rapport aux observations. Ceci constitue le fondement des systèmes de localisation basés sur le Filtrage de Kalman Etendu qui sont largement utilisés en robotique mobile. Dans ce cas l'odométrie permet de fournir une prédiction à partir de laquelle il est possible de générer un vecteur prédictif qui sera ensuite vérifié à l'aide des observations [LEO92][DUR95][DEL98][VIC00].

□ ***Les méthodes basée sur des arbres de recherche***

Le principe de la recherche dans un arbre d'interprétation nécessite dans un premier temps la construction d'un arbre représentant l'ensemble des appariements possibles entre les primitives du modèle et celles détectées dans l'image. La seconde étape consiste alors à trouver le chemin optimal en profondeur dans cet arbre en déterminant à chaque niveau un appariement entre primitives en utilisant des contraintes ou des fonctions de coût. L'utilisation de ces facteurs discriminants est justifiée afin de limiter le caractère combinatoire de cette technique.

Un des précurseurs ayant utilisé cette approche pour la localisation est Drumheller [DRU87]. Des tests, prenant en compte les contraintes physiques des capteurs ultrasoniques ("The sonar barrier test"), permettent de générer un arbre de décision.

Simsarian utilise un arbre de recherche pour effectuer la mise en correspondance [SIM96]. Il réduit la combinatoire de la méthode en prenant avantage des *view-invariant regions*. Durant une phase de pré-traitement, la carte est décomposée en région ayant la propriété suivante : tous les points d'une même région voient le même ensemble d'amers.

□ ***Les méthodes utilisant les invariants géométriques***

Ces méthodes caractérisent certaines configurations géométriques de primitives par une quantité qui reste inchangée aux points de vue et aux projections. Les valeurs de ces quantités, les invariants, sont directement utilisées pour effectuer la mise en

correspondance. Des travaux ont été réalisés dans notre laboratoire par rapport à cet aspect de la mise en correspondance [MAR99]. Ce type de méthode de mise en correspondance est utilisé dans de nombreux travaux sur la localisation, comme par exemple dans [DEA00] [LOW01].

□ ***Propagation d'une mise en correspondance initiale : le suivi multi-pistes***

Le suivi multi-pistes consiste à effectuer une mise en correspondance temporelle entre deux observations consécutives. Elle peut alors être vue comme la possibilité de propager au cours de acquisitions une mise en correspondance initiale. La variation de configuration entre deux acquisitions consécutives étant généralement faible, l'appariement réalisé entre deux acquisitions consécutives permet d'obtenir une robustesse importante sur la mise en correspondance observations - primitives cartographiques. Le suivi multi-pistes est fortement utilisé dans le domaine militaire par exemple pour le suivi de missiles.

Un algorithme de suivi est généralement constitué de deux modules :

- ✓ Le premier est chargé de détecter et d'initialiser de nouvelles pistes. Il s'agit de reconnaître des objets faisant réellement partie de l'environnement et qui peuvent donner naissance à une nouvelle piste.
- ✓ Le deuxième est chargé de suivre la trajectoire des objets observés en estimant leur état. Il doit être également capable de supprimer des pistes lorsque l'objet les concernant disparaît du champ de perception. Ce module peut-être décomposé en deux sous-modules :
 - Le premier assure la fonction de filtre de poursuite. Dans ce cadre le filtre le plus utilisé est le filtre de Kalman [JOU99].
 - Le deuxième est utilisé pour l'association cible-piste. Lorsque plusieurs pistes sont gérées en même temps, il est généralement nécessaire de déterminer les données capteurs attachées à chaque filtre de poursuite : il s'agit de l'étape d'association.

Les méthodes d'association, qui sont le fondement de la propagation d'une mise en correspondance initiale, sont en grande majorité probabilistes. Les deux algorithmes les plus répandus sont le « *Multiple Hypothesis Tracker* » (MHT) imaginé par D.B. REID en 1979 [COX96] et le « *Joint Probabilistic Data Association Filter* » (JPDAF) de Bar-Shalom (1988) [COX95].

La première de ces méthodes est issue d'une méthode plus ancienne également développée par Bar-Shalom : la méthode PDAF (Probabilistic Data Association Filter, 1973). Il s'agit d'une méthode bayésienne qui consiste à construire un estimateur prenant en compte toutes les observations contenues dans la fenêtre. Le JPDAF ajoute la possibilité de gérer l'information multi-cibles. Elle est généralement vue comme une extension du filtrage de Kalman. L'inconvénient du JPDAF réside dans son incapacité à initialiser de nouvelles pistes.

La deuxième méthode d'association, contrairement à la méthode JPDAF, autorise l'initialisation de nouvelles pistes. Il s'agit ici d'une méthode orientée mesure ce qui signifie qu'au lieu de considérer les pistes et d'associer à celles-ci les différentes mesures, on considère les mesures pour lesquelles on génère trois hypothèses de provenance :

- ✓ la mesure provient d'une cible déjà existante
- ✓ la mesure provient d'une nouvelle cible
- ✓ la mesure est une fausse alarme

Le principe de la méthode est de calculer la probabilité a posteriori de chaque hypothèse en tenant compte des détections manquantes, des fausses alarmes et de l'apparition possible de nouvelles cibles. L'inconvénient de cette méthode est son explosion combinatoire. En effet, elle construit un arbre dont chaque branche est une hypothèse et à chaque itération, chaque hypothèse donnera naissance à un arbre constitué des mêmes hypothèses.

Les deux méthodes présentées possèdent des inconvénients. La méthode JPDAF est incapable de générer de nouvelles pistes tandis la méthode MHT est coûteuse en temps de calcul. En outre, elles nécessitent toutes deux la connaissance du taux de fausses alarmes. Il existe d'autres méthodes d'association moins utilisées telles que les méthodes hybrides (associant MHT pour l'initialisation des pistes et JPDAF pour l'entretien des pistes) et neuronales. Mais aucune d'elles ne gèrent l'imprécision et l'incertitude sur les données. C'est pourquoi Arnaud Clémentin [CLE01a] [CLE01b] a proposé une méthode originale basée sur une méthode d'association qui utilise la théorie des croyances [SHA76]. Cette méthode permet de propager une mise en correspondance avec un coefficient d'incertitude associé. L'estimation de configuration du robot est d'une part robuste et d'autre part précise.

Exemples de méthodes de localisation utilisant des amers naturels

Nous pouvons regrouper les systèmes de navigation utilisant des amers naturels par rapport aux capteurs extéroceptifs utilisés. Les méthodes de localisation basées l'utilisation de télémètre consiste généralement, après segmentation, à effectuer une mise en correspondance par rapport aux parois de l'environnement. Les précurseurs par rapport à ce type d'approche sont Drumheller [DRU87] et Crowley [CRO89] qui utilisent des capteurs à ultrason. De nombreuses extensions ont été donné à ces travaux, comme par exemple [SAB94] où une caractérisation des paramètres géométriques de balises naturelles de type parois ou arêtes permet de localiser de robot. Citons enfin les travaux de Kieffer [KIE99] sur lesquels nous nous sommes appuyés dans ce mémoire, qui propose finalement une méthode de localisation à erreur bornée qui utilise des données ultrasonores. La mise en correspondance est ici faite au sens de l'inversion ensembliste.

La télémétrie laser, plus précise que celle à ultrasons, est très souvent utilisé et permet de rendre plus robuste la phase de mise en correspondance. Ainsi l'appariement de « segments laser » avec ceux répertoriés au sein d'une carte de l'environnement est géré dans [WEI94] [PER99][RUP00] pour la localisation absolue d'un robot dans son environnement.

Enfin les systèmes de vision, qu'ils soient monoculaire, omnidirectionnels ou stéréoscopiques, par la densité et la pertinence des informations sensorielles qu'ils fournissent, sont eux aussi fréquemment utilisés pour la problématique qu'est la localisation absolue. Le principe général sera alors de mettre en correspondance des points d'intérêt extraits de l'image tels que des droites radiales [ATI93] [YAG94] [MAR99], des droites horizontales [SHA97], des coins [LIT98], des balises naturelles de type SIFT (Scale Invariant Feature Transform) [LOW01]. Une mise en correspondance directe de zones d'intérêt de l'image peut être aussi gérée, comme dans [MAL01] par une minimisation d'une distance entre une image 3D locale et le modèle.

1.3.3 Méthodes hybrides

Les chercheurs se sont rendu compte de l'intérêt à utiliser conjointement la localisation à l'estime et la localisation absolue. On parle alors de recalage dynamique. Mais les méthodes hybrides intègrent aussi celles qui s'appuient sur la coopération de plusieurs capteurs proprioceptifs ou extéroceptifs. Dans cette optique, nous proposons une synthèse non exhaustive des méthodes basées sur le recalage et celles basées sur l'association de plusieurs capteurs. C'est dans ce cadre général que des problématiques de fusion de données sont à traiter.

1.3.3.1. Le recalage dynamique

Les méthodes de localisation à l'estime ont pour gros inconvénient de générer une erreur cumulative avec la distance parcourue. Pour pallier ce problème de dérive, on utilise un système de localisation absolue qui sera chargé de corriger régulièrement l'estimation relative de position. Cette méthode de correction dynamique est désignée sous le terme de recalage dynamique. Pour une optimisation en temps de l'algorithme de navigation, le recalage de la localisation relative par une estimation de position absolue ne s'effectuera que lorsque l'imprécision sur la configuration du véhicule devient supérieure à un seuil prédéfini.

Parmi les méthodes de fusion existantes, la plus communément utilisée est celle basée sur le Filtrage de Kalman Etendu (EKF). Cette méthode permet de fournir une estimation de la configuration du robot ainsi que de son imprécision associée au sens de la minimisation du critère quadratique.

Leonard et Durrant-Whyte dans [LEO92] ont été, après Crowley [CRO89], dans les premiers à utiliser le Filtrage de Kalman Etendu (EKF) pour localiser un robot dans son environnement par recalage dynamique. Une prédiction de la configuration du robot fournie par l'odométrie est utilisée d'une part pour simplifier la mise en correspondance avec les observations fournies par des capteurs à ultrasons et d'autre part pour linéariser l'équation de mesure. Les écarts entre les mesures et les prédictions sont utilisés pour estimer la position finale. Cette approche générique sera étendue par Durrant-Whyte à d'autres systèmes de perception extéroceptifs telles qu'un radar à ondes millimétriques dans [DUR95] ou à un télémètre laser embarqué sur un véhicule routier [GUI00a] [GUI00b]. Cette dernière étude portera sur une extension concernant une démarche de localisation et de modélisation simultanée.

Bonnifait et Garcia, de l'IRCyN, ont développé, sur le véhicule Melody, un système de localisation pour environnement extérieur utilisant odométrie et balises artificielles [BON98]. Ces balises sont des sources lumineuses et sont détectées avec une caméra CCD linéaire tournant à 1 rad/s. Un Filtrage de Kalman Etendu réalise la localisation.

Un Filtrage de Kalman Etendu est également utilisé par Arsénio et Ribeiro pour fusionner des données odométriques et un relevé télémétrique laser [ARS98]. Tout comme [SIM96], une localisation dite « active » est proposée : elle consiste à orienter le télémètre laser de manière à ne percevoir que les amers intéressants, à savoir ceux qui sont visibles à partir de la région où se situe le robot.

Dans [VIC00], un algorithme basé sur la fusion d'observations télémétriques issues d'un télémètre laser avec des données odométriques est proposé. Une représentation sous forme de segments auxquels est associée une incertitude est obtenue à partir du relevé laser. Le processus de fusion se fait grâce à un Filtre de Kalman Étendu.

Virgil, le robot destiné à faire une visite guidée de l'université de Rice aux Etats-Unis, fusionne les données issues d'un récepteur GPS avec les informations odométriques à l'aide d'un filtrage de Kalman étendu [THR01]. Le problème de l'obstruction des signaux GPS est notamment traité. La précision de la localisation atteint 40 cm.

Chung utilise un filtrage de Kalman pour fusionner les informations issues d'un gyroscope à fibre optique et d'un odomètre [CHU01]. Une calibration soignée du gyroscope a été réalisée, notamment au niveau de l'erreur due à la dérive en température du capteur. Au final, il s'avère que la précision de localisation, qui est très bonne, est due d'une part à la calibration méticuleuse du gyroscope et d'autre part à son couplage avec un odomètre.

Le filtrage de Kalman Étendu est l'outil le plus couramment utilisé dans les processus de recalage dynamique pour sa relative simplicité d'utilisation et pour son efficacité. En revanche son gros inconvénient se situe au niveau de l'étape de linéarisation, qui, si l'estimation odométrique est éloignée de la position réelle du robot, fera diverger le filtre.

1.3.3.2. La coopération multi-capteur

Le recalage dynamique n'est qu'un cas particulier d'association possible de capteurs. En effet, d'autres combinaisons, telles qu'un ensemble de capteurs extéroceptifs fournissant des données complémentaires, sont intéressantes à utiliser pour la perception en robotique mobile. En effet, un capteur utilisé seul ne peut relever complètement toutes les caractéristiques de l'environnement. Il est généralement nécessaire de combiner des données provenant de plusieurs capteurs extéroceptifs.

Deux types de combinaison peuvent être réalisés dans les problèmes de fusion de données :

- ✓ une combinaison d'informations *redondantes*. L'intégration et la fusion de l'information redondante aide alors à accroître la précision et le degré de confiance d'une estimation obtenue avec plusieurs observations distinctes.
- ✓ une combinaison d'informations *complémentaires*. L'intégration et la fusion de données complémentaires permettent généralement de fournir une information ne pouvant être obtenue séparément avec chacun des capteurs.

De nombreux travaux utilisent la coopération de capteurs pour contribuer aux problématiques que sont la localisation et la modélisation de l'environnement.

Dans [NEI99], Neira ajoute à l'information odométrique classique, un système associant une image de profondeur et une image d'intensité, toutes deux issues d'un capteur laser. Cependant, il fait une utilisation limitée des deux images puisqu'il ne se sert de l'image d'intensité que pour affiner la position des coins détectés avec l'image de profondeur. Un filtrage de Kalman achève ensuite la localisation du robot. La méthode s'avère précise puisque, à la suite de résultats expérimentaux, Neira aboutit à une erreur d'environ 10 cm en position et 0.15 degrés en orientation.

Yagi utilise conjointement un capteur de vision omnidirectionnel et un capteur ultrasonique [YAG95]. Le système de vision omnidirectionnel donne l'angle d'azimut des amers verticaux de l'environnement. Le capteur ultrasonore fournit une confirmation de l'espace libre entre deux amers verticaux.

Arras et Tomatis localisent le robot Pygmalion par coopération entre un télémètre laser et un système de vision monoculaire [ARR99]. Une segmentation du relevé laser lui permet d'obtenir un ensemble de primitives de type segment. L'application de la transformée de Hough sur l'image CDD fournit les angles des amers verticaux devant le robot. La carte théorique de l'environnement est composée d'une liste de segments et d'une liste d'amers verticaux. L'appariement des deux types d'objets est réalisé grâce à un flitrage de Kalman étendu.

Lallement, du CRAN, utilise conjointement un télémètre laser et une caméra monoculaire CCD orientable. La construction de la carte sensorielle de l'environnement est effectuée en deux temps [LAL98] :

- ✓ A partir du télémètre laser, il obtient un relevé 2D de l'environnement qu'il segmente avec un algorithme type Duda-Hart. Deux segments consécutifs lui permettent d'obtenir une primitive qu'il appelle "coin".
- ✓ Ensuite, avec le système de vision monoculaire, il inspecte ces primitives "coins" pour détecter des amers verticaux contenus dans les deux plans formant ce "coin" grâce à un filtre de Deriche. C'est donc une approche de type vision active.

Enfin, il réalise un appariement de la carte théorique de l'environnement avec le modèle sensoriel ainsi obtenu en utilisant la notion d'invariant [LAL99].

1.4. Localisation et modélisation simultanée de l'environnement

Lorsque la carte de l'environnement du robot n'est pas connue a priori, un module de génération de carte incrémentale doit obligatoirement être intégré au système de navigation. Il s'agit là d'une étape supplémentaire qui consiste à mettre à jour au fur et à mesure des acquisitions, une représentation du milieu d'évolution du robot. Ce processus se décline comme une fusion successive des différents modèles sensoriels générés lors du déplacement du robot. Il paraît clair à ce niveau que la fusion incrémentale de modèles sensorielles nécessite obligatoirement leurs recalages systématiques avec les primitives déjà insérées dans la carte. Cette étape de recalage est ni plus ni moins une localisation du robot par rapport à une connaissance acquise au cours du déplacement. L'étape de localisation devient alors indissociable de celle de modélisation et on parle alors d'un système de « localisation et modélisation simultanée ».

Les méthodes de modélisation incrémentale sont classées en deux grandes familles qui sont fonction du type de représentation choisi :

- ✓ Les méthodes de modélisation métriques, qui décrivent explicitement la position « géométrique » des éléments de l'environnement [NEI99]

- ✓ Les méthodes de modélisation topologiques basées sur des graphes représentant des informations de plus haut niveau comme certaines places caractéristiques de l'environnement (coins, croisement de deux couloirs, jonctions en T, etc.).

La première catégorie peut être subdivisée en deux sous familles :

- ✓ Les méthodes de modélisation purement géométrique qui gèrent explicitement les positions « cartésiennes » des primitives cartographiques.
- ✓ Les méthodes probabilistes, appelées encore méthodes de modélisation par grille d'occupation, qui décrivent les propriétés métriques par discrétisation de l'environnement en y ajoutant des informations d'incertitude.

Nous arrivons donc finalement aux trois principaux types distingués et décrits par Dudek dans [DUD97].

1.4.1 *Les approches métriques*

Les méthodes de modélisation métriques intègre la notion de distance au sein de la carte. Comme nous l'avons mentionné précédemment deux grands types de méthodes métriques se distinguent : les méthodes géométriques et les méthodes dites probabilistes.

1.4.1.1. Les méthodes de modélisation géométriques

Ces méthodes de modélisation décrivent explicitement les coordonnées x , y voire z des amers caractéristiques de l'environnement. Il s'agit là d'une représentation cartésienne de l'environnement. L'outil le plus couramment utilisé pour gérer la mise à jour de la carte est ici le Filtrage de Kalman Étendu. En effet l'estimation optimale ou pseudo-optimale au sens de la minimisation du critère quadratique de la configuration du robot peut être étendue à l'estimation des coordonnées (x,y) des primitives cartographiques. En outre cet outils, comme nous l'avons vu pour la localisation, permet de gérer efficacement la notion d'imprécision. Les précurseurs de ce type d'approche sont là encore Crowley et Durrant-Whyte. Crowley dans [CRO89], propose un algorithme de localisation et de modélisation de l'environnement basé sur le filtrage de Kalman Étendu. Les mesures issues de capteurs ultrasonores sont fusionnées avec des données odométriques.

Leonard et Durrant-Whyte dans [LEO92] propose un paradigme de localisation et de mise à jour incrémentale de carte basé sur le filtrage de Kalman Étendu. Là aussi des capteurs ultrasonores et des odomètres sont utilisés. Cependant l'étude porte sur une mise à jour incrémentale d'un niveau sémantique plus élevé que précédemment puisque les primitives gérées sont de type RCD (Region of Constant Depth). Le formalisme adopté permet de quantifier les erreurs sous forme d'ellipse d'imprécision.

Plus récemment, Durrant-Whyte a rendu pérenne son étude en transposant la démarche précédente à un système de localisation et de modélisation simultanée embarqué sur un véhicule routier (système SLAM pour Simultaneous Localization and Map Building) [GUI00a] [GUI00b] [DIS00].

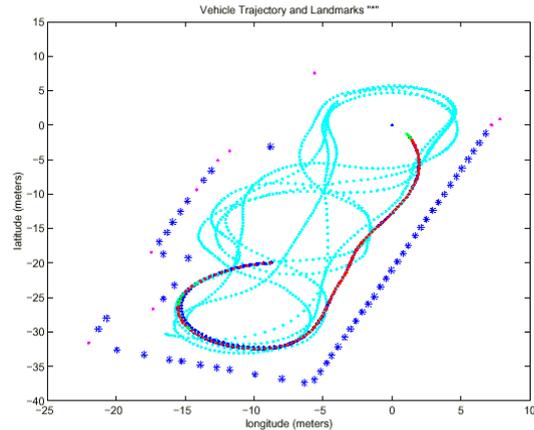


Figure 1-13 : Système SLAM développé dans [GUI00a].

Dans ces articles les auteurs propose un système de navigation en milieu extérieur (SLAM) basé sur un système perceptif composé d'un télémètre laser et d'odomètres. De nombreux essais ont pu validé l'efficacité de ce système puisque des erreurs en position de l'ordre du cm ont été constaté par rapport à une vérité terrain fournie par un GPS. Précisons que la robustesse de l'algorithme est notamment due au retrait de certaines primitives cartographiques caractérisées par une incertitude importante.

Sur le plan national, des approches similaires aux travaux précédents ont été réalisés. Nous pouvons citer les travaux [PAR92][MOU91][DEL98][RIV99], proposant des algorithmes de localisation et de reconstruction simultanée basés sur l'utilisation du filtrage de Kalman Etendu.

Le filtrage de Kalman Etendu est aussi utilisé dans [SHA97] avec un système de vision omnidirectionnel stéréoscopique pour modéliser des points d'intérêt de l'environnement fournis par des horizontales. Ces points d'intérêt intégrés incrémentalement dans une carte servent à localiser le robot.

Le Filtrage de Kalman Etendu pour les approches géométriques n'est pas obligatoirement utilisé. Nous pouvons dans ce cadre citer les travaux de Yagi [YAG95] qui à l'aide du capteur de vision omnidirectionnel COPIS modélise l'environnement sous forme de polygone. Les points correspondent aux amers verticaux et les liens entre les points aux secteurs de niveaux de gris appariés. Dans la même optique les extrémités des segments sont gérés au sein de la carte dans [GON94].

Tout comme pour la localisation le filtrage de Kalman Etendu est un formalisme efficace permettant de fournir une estimation précise, mais il possède l'inconvénient de diverger en cas d'erreur sur l'estimation odométrique.

1.4.1.2. Les méthodes de modélisation basées sur des grilles d'occupation

Ces méthodes sont basées sur une discrétisation de l'environnement sous forme de grille. Chaque élément de la grille est une cellule qui fournit bien sûr la position d'un amer mais aussi une information sur la probabilité d'existence de l'amer. Ce type de méthode est robuste, mais possède l'inconvénient d'être relativement lourde d'un point de vue mémoire.

La précision est évidemment limitée à la résolution de la grille. Précisons que cette résolution conditionne aussi le coût en temps de calcul : plus cette résolution sera forte, plus les temps de calcul seront importants.

Les premiers travaux à avoir intégré ce type de représentation sont ceux de Moravec et Elfes [MOR88] [ELF87]. Dans ces travaux des capteurs à ultrasons sont utilisés et permettent de déterminer l'état d'une cellule (occupé, indéterminé, libre) en fonction d'un modèle de l'erreur de mesure. Ces travaux ont été repris et étendus dans de nombreuses autres études telles que [BOR91] [SCH94].

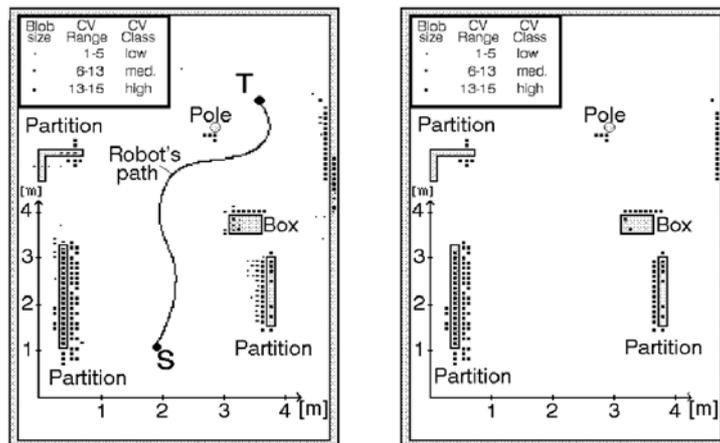


Figure 1-14 : Résultats expérimentaux obtenus par Borenstein [BOR91].

Burgard dans [BUR96] utilise une grille de probabilité pour représenter l'environnement mais il l'utilise aussi pour se localiser en tirant partie d'informations odométriques et ultrasonores.

Plus récemment, Pagac et Durrant-Whyte dans [PAG98] à affiner la modélisation sous forme de grille de probabilité en utilisant un modèle décisionnel de fusion et d'intégration de données basé sur la théorie de Dempster-Shafer.

Enfin, des travaux récents et porteurs ont été proposés par Fox et Thrun [FOX99][THR99] [THR00] qui se proposent de gérer le paradigme de localisation et modélisation sous forme de grilles probabilistes avec des processus de Markov. Les capteurs utilisés pour reconstruire l'environnement sont des capteurs à ultrasons. Une quantification robuste et fine du modèle d'erreur de mesure est obtenu avec des réseaux de neurones. Le système est globalement vu comme étant entropique et des algorithmes d'optimisation sont utilisés pour augmenter la robustesse du module de génération de carte. La méthodologie développée ici a permis d'obtenir une grande précision de la représentation cartographique sur des environnements de grandes tailles.

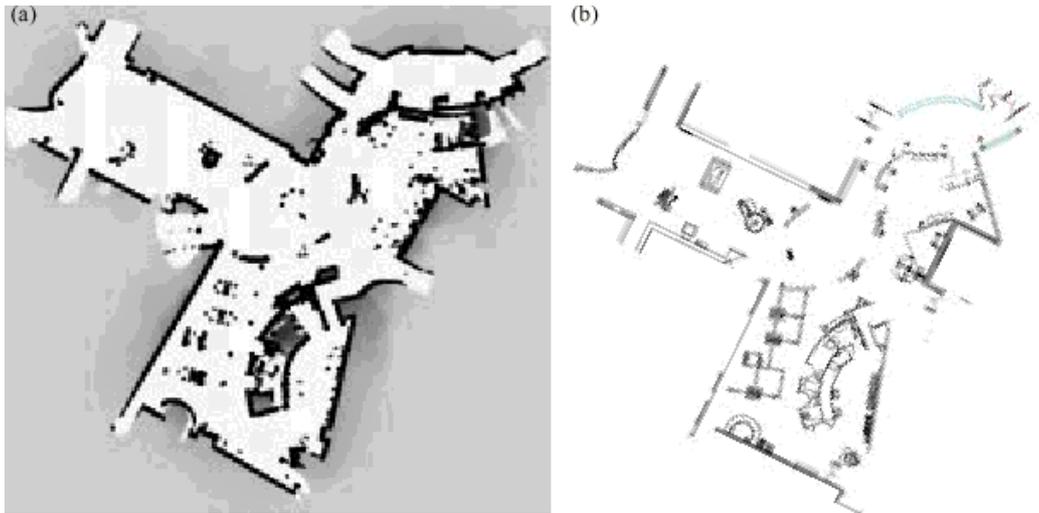


Figure 1-15 : (a) grille obtenue sur un environnement de grande dimension par la méthode de Fox et Thrun (b) carte réelle de l'environnement

Globalement nous avons pu constater que les approches permettent de gérer une représentation de l'environnement avec plus ou moins de précision. Ces cartes intègrent un nombre limitée de caractéristiques, comme par exemple une quantification de l'imprécision pour les méthodes géométrique, ou la notion d'état d'une cellule pour les grilles. Pour pouvoir gérer des caractéristiques de plus haut niveau, telles que la notion de coins, de jonctions, il est nécessaire d'utiliser un type de représentation plus descriptif : l'approche topologique.

1.4.2 Les approches topologiques

Les méthodes de modélisation topologiques permettent de monter en niveau sémantique par rapport aux représentations précédentes. Le principe est ici de décrire l'environnement sous forme de graphes qui représentent la connectivité entre des lieux caractéristiques et prépondérants de l'environnement au sens de la navigation. Ces lieux doivent correspondre à des places fortes permettant de localiser le robot de façon sûre, robuste et précise. Elles sont généralement significatives d'une densité forte d'amers naturels observables. Ces configurations privilégiées de l'environnement pourront par exemple décrire des coins, des croisements de couloirs, des jonctions en T.

Tout comme pour le filtrage de Kalman dans les représentations géométriques, le graphe de Voronoï est un outils fortement utilisé pour les représentations topologiques [CHO96] [NAG98] [MAH98] [BLA00]. Un diagramme de Voronoï est défini de la manière suivante : Soit $S = \{p_1, p_2, \dots, p_n\}$ un ensemble de points du plan. La cellule de Voronoï $Cell_S(p_i)$ est la région du plan contenant les points r qui sont plus proches de p_i que de n'importe quel autre point $p_j \in S - \{p_i\}$. Le diagramme de Voronoï $VOR(S)$ est la réunion des $Cell_S(p)$, pour tous $p_i \in S$.

Les espaces libres sont utilisés comme lieu distinctif ("distinctive places") dans [KUI91]. Les auteurs décrivent l'espace de manière hiérarchique en prenant des niveaux topologiques qui représentent la connectivité entre lieux distinctifs ("distinctive places"). Les arcs du graphe correspondent à la stratégie utilisée par le robot pour aller du nœud de départ au nœud d'arrivée de l'arc.

Kortenkamp [KOR94] (Figure 1-16), de l'Université du Michigan, représente l'environnement d'évolution du robot par un arbre : chacun des nœuds représente une région distincte de l'environnement et les arcs représentent les chemins entre ces emplacements caractéristiques.

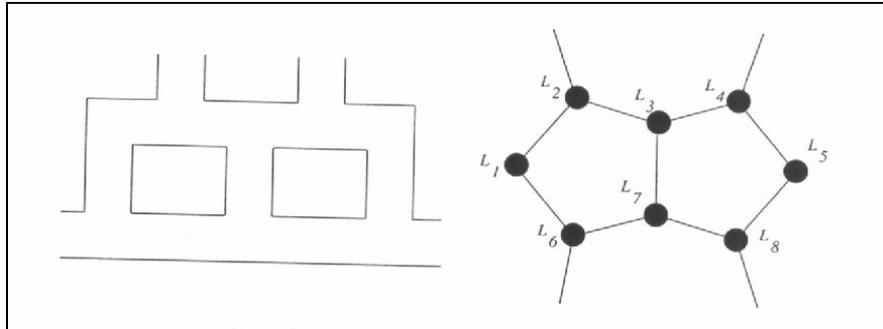


Figure 1-16 : Un exemple d'environnement et graphe correspondant [KOR94]

Thrun dans [THR98] propose une modélisation hybride métrique-topologique. La carte topologique permet de corriger les désalignements obtenus sur la carte métrique après des déplacements cyclique du robot. Les désalignements sont ici significatifs des erreurs cumulatives sur un trajet important (Figure 17). La carte topologique sert à répertorier les places significatives de l'environnement telles que des coins par exemple. L'auteur précise ici qu'il n'y a pas intégration d'un niveau sémantique comme dans [KUI91]. La mise en correspondance de ses places significatives « non étiquetées » permet de corriger les désalignements dans la carte métrique.

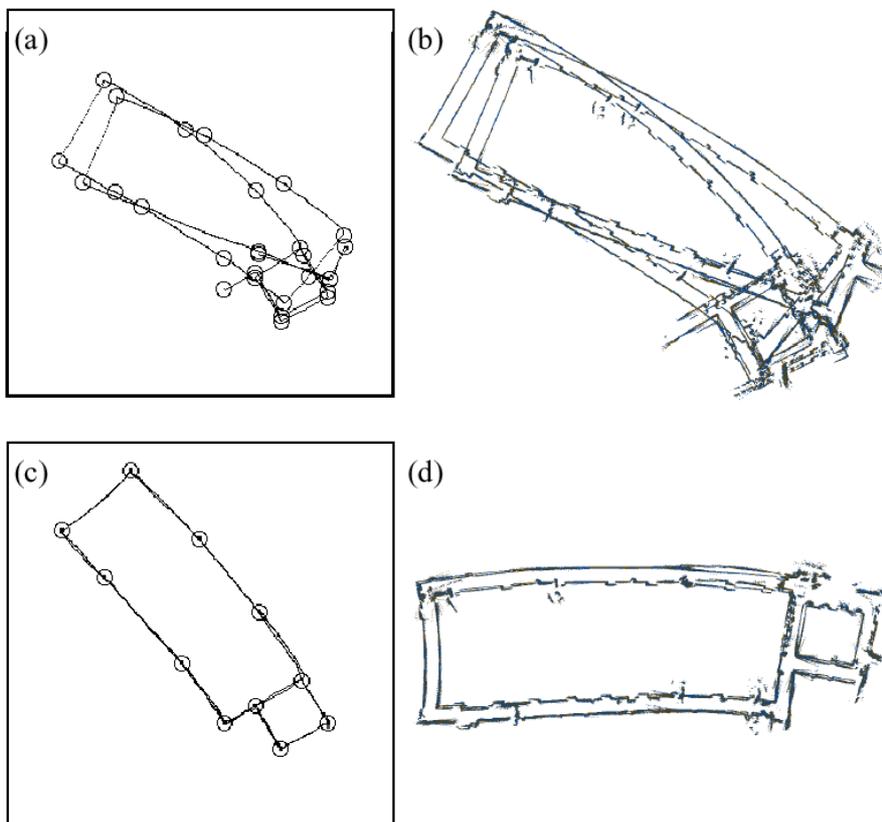


Figure 17 : Carte topologique et métrique obtenue par Thrun dans [THR98]

Les systèmes de vision sont des systèmes privilégiés pour construire des graphes de zones significatives au sens de la navigation. Des méthodes propres au traitement d'image permettant d'établir des corrélations fortes entre des images et une base de données pourront dans ce cadre être utilisées pour déterminer les places significative. Ainsi l'analyse en composante principale est utilisée par Dudek dans [DUD00] pour sélectionner des points d'intérêt dans des séquence d'images. La vision omnidirectionnelle, particulièrement bien adaptée dans ce cas de figure, est aussi utilisée dans [ULR00] où un système de localisation reposant sur une représentation topologique est proposé. Là encore, la méthode repose sur la comparaison de l'image perçue par une caméra panoramique avec des images appartenant à une base de données représentatives des différents lieux de l'environnement.

1.4.3 *Conclusion sur les méthodes de modélisation*

Les deux types de méthodes modélisation que nous avons présentées ont des avantages et des inconvénients. Les approches métriques dites probabilistes, sont robustes et relativement efficaces. En outre elles permettent de répondre au problème de divergence lié à l'utilisation du filtrage de Kalman Etendu dans des approches dites géométriques. En revanche elles possèdent le gros désavantage d'être lourdes en temps de calcul et limitées en précision. L'autre problème de ces méthodes est leurs niveaux sémantiques faibles qui globalement se limitent à une certitude sur l'existence d'une cellule. La deuxième approche quant à elle permet justement d'intégrer une description de plus haut niveau que dans les approches métriques, ceci en intégrant la notion de places significatives au sens de la navigation. Ces lieux caractéristiques, ainsi que les liens entre ces lieux sont généralement gérés dans un graphe, d'où une combinatoire moindre et une optimisation du paramètre « mémoire » pour le stockage des données. En outre la quantification précise des contraintes géométriques entre ces places significatives, par exemple pour des petits déplacement du robot, s'avère être un problème.

Nous pouvons finalement constater que ces deux approches sont complémentaires. Il est donc intéressant de pouvoir associer ces deux types de représentation dans un module de génération de carte. Cette approche complémentaire, que l'on peut qualifier d'approche hybride, est de plus en plus intégrée dans des travaux récents [THR98][GAS99][VAN01]. Les résultats obtenus montrent clairement l'efficacité d'une telle approche.

1.5. **Conclusion**

Le paradigme de localisation et modélisation incrémentale de l'environnement nécessite de s'intéresser obligatoirement aux éléments de la chaîne de perception que sont les capteurs, les méthodes de localisation et celles de modélisation. C'est ce que nous avons fait de façon la plus synthétique possible dans cet état de l'art. Les constats pouvant être dégagés quant à ces trois modules sont multiples.

Par rapport aux systèmes de perception utilisés en robotique mobile, deux familles de capteurs peuvent être utilisées. Nous avons pu constater que celles ci sont plus complémentaires que concurrentes. C'est pour cette raison qu'un système de navigation sera généralement basé sur l'exploitation des informations sensorielles émanant de ces deux

catégories de capteurs. Nous pouvons étendre cette remarque à la problématique spécifique qu'est la perception du milieu d'évolution du robot : l'emploi d'un unique capteur extéroceptif pour cette mission sera généralement insuffisant. Ainsi, l'association de plusieurs capteurs extéroceptifs, qu'ils soient complémentaires ou redondants, permettra d'obtenir un modèle sensorielle robuste et hautement descriptif. Enfin parmi les capteurs extéroceptifs à privilégier, les systèmes de vision omnidirectionnelle semblent être particulièrement adaptés pour le problème de la navigation en robotique mobile. Un système de perception embarqué efficace et robuste sera généralement fondé sur la coopération de plusieurs capteurs.

Des remarques similaires peuvent être amenées sur les méthodes de localisation. Les deux grandes familles de méthodes, relative et absolue, possèdent chacune des avantages et des inconvénients. C'est pour cette raison qu'une grande partie des systèmes de navigation intègre les deux solutions pour donner lieu à des algorithmes de recalage de l'estimation odométrique par l'estimation absolue. Là encore, parmi toutes les méthodes décrites, un système de localisation se démarque par rapport aux contraintes de robustesse, de précision et de gestion temps réel imposées par la navigation des engins mobiles : il s'agit de celles basées sur le repérage de balises artificielles. Néanmoins, le manque de flexibilité de ces méthodes est un inconvénient relativement lourd. Une solution pouvant être dégagée consiste à développer des méthodes de localisation qui se basent sur des amers naturels "les plus observables possibles". Ainsi, la notion de "balises naturelles verticales" est à même de répondre à ce problème "d'observabilité" au sens de la perception, et est donc par la même en mesure d'augmenter de manière significative la robustesse de l'étape de localisation. La coopération de capteurs adaptés à la détection de ce type de balises naturelles, par exemple les systèmes de vision omnidirectionnelles, sera de nature à renforcer cette robustesse.

La localisation sans connaissance a priori nécessite d'intégrer un module de construction incrémentale de carte. Nous avons pu remarquer que quelque soit la catégorie de méthode de modélisation employée, l'étape de localisation est indissociable de celle de mise à jour. En d'autres termes, toute erreur sur l'estimation de la configuration du robot aura une influence sur celle commise lors de la mise à jour. Cette forte interaction est d'ailleurs à la base du constat, souvent fait dans les travaux portant sur cette problématique, qui souligne les phénomènes d'erreurs cumulatives sur des longs trajets. Une robustesse importante du paradigme de localisation et de modélisation passera donc par une gestion la plus rigoureuse possible de ces imprécisions. C'est notamment ce que nous prendrons en compte dans ces travaux avec l'utilisation du formalisme de l'inversion ensembliste.

Chapitre 2

Coopération entre deux systèmes de vision omnidirectionnelle pour la localisation

Sommaire partiel

2.1.	INTRODUCTION	38
2.2.	LE SYSTEME DE PERCEPTION OMNIDIRECTIONNEL STEREOSCOPIQUE	38
2.3.	LA SEGMENTATION RADIALE	41
2.4.	APPARIEMENT DES SECTEURS	44
2.4.1	<i>La théorie des croyances de Dempster-Shafer</i>	45
2.4.2	<i>Calcul des critères d'appariement</i>	48
2.4.3	<i>Fusion des critères d'appariement</i>	50
2.4.4	<i>Algorithme global d'appariement des secteurs</i>	51
2.5.	CALCUL DES POINTS DE L'ENVIRONNEMENT DANS LE MODELE SENSORIEL	53
2.6.	ALGORITHMES DE LOCALISATION	53
2.6.1	<i>Les primitives considérées</i>	53
2.6.2	<i>Algorithmes de mise en correspondance</i>	55
2.7.	RESULTATS EXPERIMENTAUX	60
2.8.	CONCLUSION	66

2.1. Introduction

Nous avons pu constater, par rapport à l'état de l'art que nous avons précédemment établi, que la robustesse d'une méthode de localisation dépend essentiellement de celle de la phase de mise en correspondance. Or, plus le modèle sensoriel sera précis et riche en information, plus la phase de mise en correspondance sera robuste. Dans ce chapitre nous présentons un système de perception original faisant coopérer deux capteurs de vision omnidirectionnelle et qui génère un nombre important d'informations sensorielles. La précision et la densité de ces données seront exploitées pour construire un modèle sensoriel robuste de l'environnement. En mettant en correspondance ce modèle avec des données cartographiques nous localiserons le robot de façon absolue.

Dans la première partie, nous présentons le principe du système de perception omnidirectionnel stéréoscopique. Dans une seconde partie, nous proposons une méthode de construction du modèle sensoriel utilisant la fusion de plusieurs critères, fusion réalisée avec les règles de Dempster-Shafer. La méthode de localisation absolue utilisée sera présentée dans la dernière partie. Les expérimentations seront réalisées dans des environnements intérieurs de type polyédrique.

2.2. Le système de perception omnidirectionnel stéréoscopique

Le système perception que nous avons développé et utilisé pour la localisation et la modélisation de l'environnement est basé sur la coopération de deux capteurs de vision omnidirectionnelle SYCLOP (Figure 2-1). Rappelons brièvement que le système SYCLOP est similaire au système COPIS utilisé par Yagi [YAG90]. Le prototype SYCLOP réalisé pour les besoins de l'expérience mesure environ 60 cm de hauteur et se compose d'un miroir conique et d'une caméra CCD (Figure 2-1). Les caractéristiques du miroir conique sont un angle au sommet de 100 degrés et une base b de 19 cm. Ce système de vision, utilisé depuis de nombreuses années dans notre laboratoire, permet notamment de détecter tous les objets verticaux de l'environnement par une projection 2D sur le plan image de la caméra [PEG96] [DEL97a] [DEL97b]. Une image prise par ce capteur est donnée en Figure 2-1.

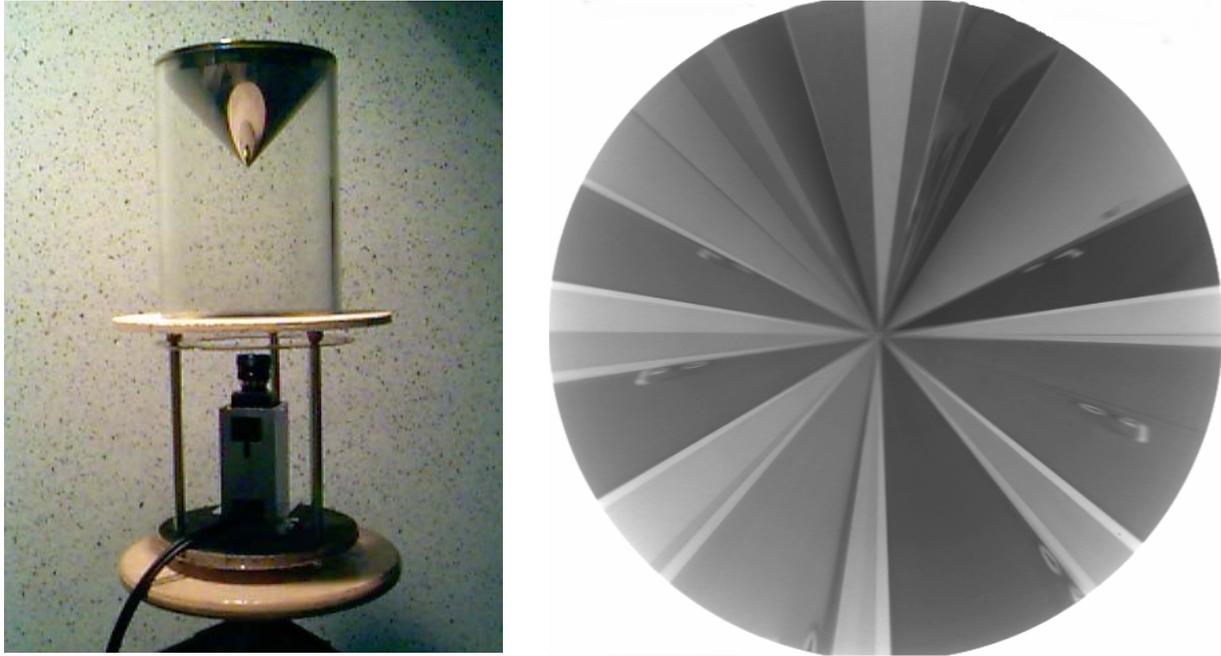


Figure 2-1 : Le capteur SYCLOP et une de ses acquisitions.

La coopération des deux capteurs SYCLOP consiste à prendre deux acquisitions par rapport à deux positions distinctes éloignées d'une distance d , le mouvement rigide étant une translation.

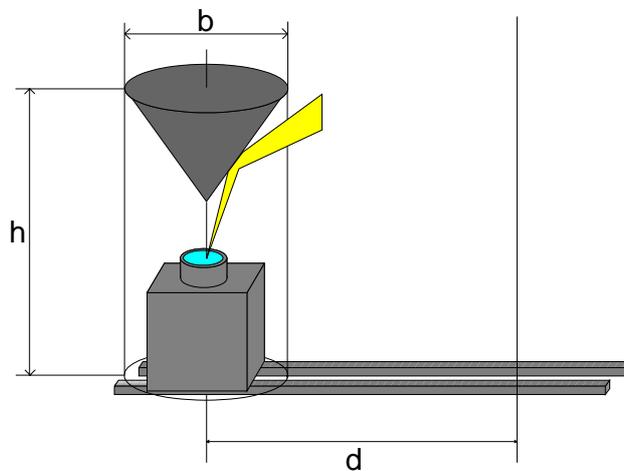


Figure 2-2 : Principe du capteur omnidirectionnel stéréoscopique.

La translation rigide est réalisée à l'aide de deux rails horizontaux qui permettent un déplacement rectiligne précis dans le plan horizontal du capteur SYCLOP (Figure 2-2). Ce système assure l'acquisition de deux images omnidirectionnelles de l'environnement distantes de 40 cm. Cette distance entre les deux prises a été déterminée de façon expérimentale en effectuant un compromis entre encombrement et précision dans la détermination des primitives sensorielles. Le capteur est déplacé sur ses rails à l'aide d'un moteur pas à pas et d'une crémaillère, le robot se trouvant à l'arrêt. Une solution plus intégrée aurait été d'utiliser deux miroirs coniques comme dans l'étude menée à l'université de Besançon avec le projet STAN [BID95], qui propose une méthode de localisation basée sur les réseaux de neurones.

Svoboda [SVO98a] a également développé un système basé sur l'utilisation de deux capteurs omnidirectionnels utilisant la géométrie épipolaire. Cette approche permet de mettre en correspondance un point de l'espace sur les deux images. Toutefois, ce système n'est pas adapté à notre problème pour plusieurs raisons :

- Une calibration complète du système est obligatoire,
- Ce sont les points qui sont mis en correspondance, alors que nous cherchons uniquement les droites radiales représentant les amers verticaux de l'environnement,
- Le capteur utilisé doit avoir un point de vue unique, comme l'hyperbole, ce qui n'est pas le cas du miroir conique utilisé.

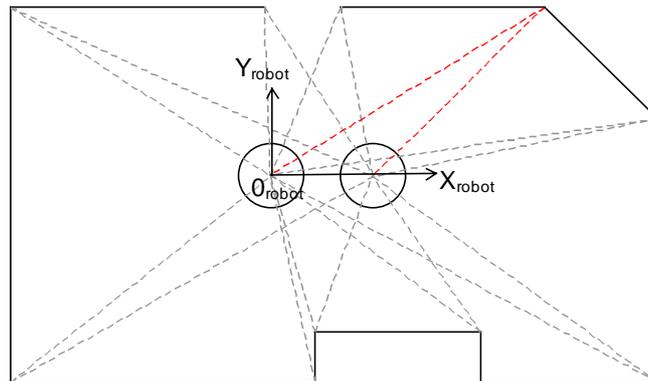


Figure 2-3 : Détermination des primitives sensorielles.

Pour une acquisition, un amer vertical est caractérisé par une droite radiale de fort contraste. En connaissant pour chacun des points, les deux droites radiales générées sur chaque cône, on peut calculer leurs positions dans le plan, ce qui nous donne la position des amers verticaux (bords, coins, portes...) dans le repère du robot (Figure 2-3).

Trois traitements seront nécessaires à l'obtention des points:

- ✓ La segmentation radiale des deux images,
- ✓ La mise en correspondance des secteurs trouvés,
- ✓ Le calcul des coordonnées des points obtenus.

Avant de pouvoir effectuer des traitements sur les images obtenues il faut calibrer le système de vision. Toutefois, C. Cauchois a montré que pour l'extraction unique des droites radiales avec un miroir conique, seul un alignement précis du centre optique de la caméra avec la pointe du cône était nécessaire. Par contre pour pouvoir utiliser d'autres primitives comme des points particuliers, ou des segments horizontaux, une étape complète de calibration relativement complexe est nécessaire. Pour plus d'information sur la calibration complète de capteur SYCLOP, on se référera à [CAU99a] [CAU99b] et [CAU01]. En nous basant sur ces travaux et en notant que nous ne traitons spécifiquement que des primitives radiales, nous pouvons affirmer que l'amélioration (par rapport à la précision et à la robustesse) alourdit considérablement les temps de calcul.

Pour retrouver le centre du cône sur une image, nous utilisons une mire constituée de bandes noires et blanches, qui est placée autour du système. La segmentation de l'image ainsi obtenue, et le calcul de l'intersection des droites en résultant, nous donne le centre du miroir conique dans l'image (Figure 2-4).

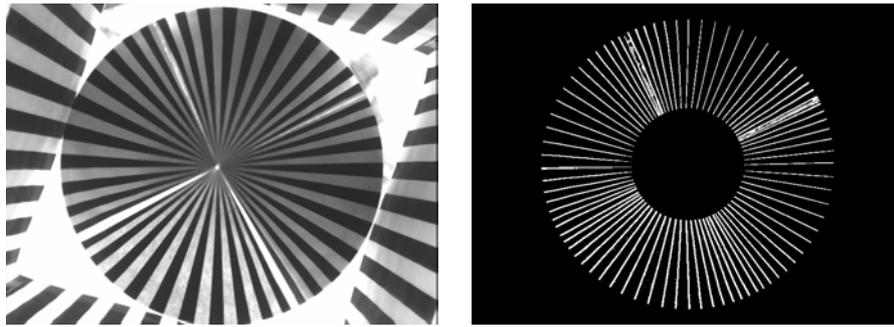


Figure 2-4 : Image de la mire et segmentation en résultant.

Ensuite, un banc micrométrique nous permet de centrer l'axe optique avec l'axe du miroir conique.

2.3. La segmentation radiale

On cherche à traiter dans les images les secteurs de niveaux de gris homogènes qui sont délimités par des droites radiales caractérisant des amers verticaux. Les droites étant radiales et distribuées sur 360 degrés à partir du centre de l'image, il est possible de travailler sur un signal mono-dimensionnel (Figure 2-5). On considère donc sur l'image un cercle concentrique de niveaux de gris (par rapport au centre optique de l'image). Afin de gérer une densité maximum d'information, ce cercle est pris sur la périphérie. On obtient alors une courbe de niveaux de gris représentative pour chaque image (Figure 2-5).

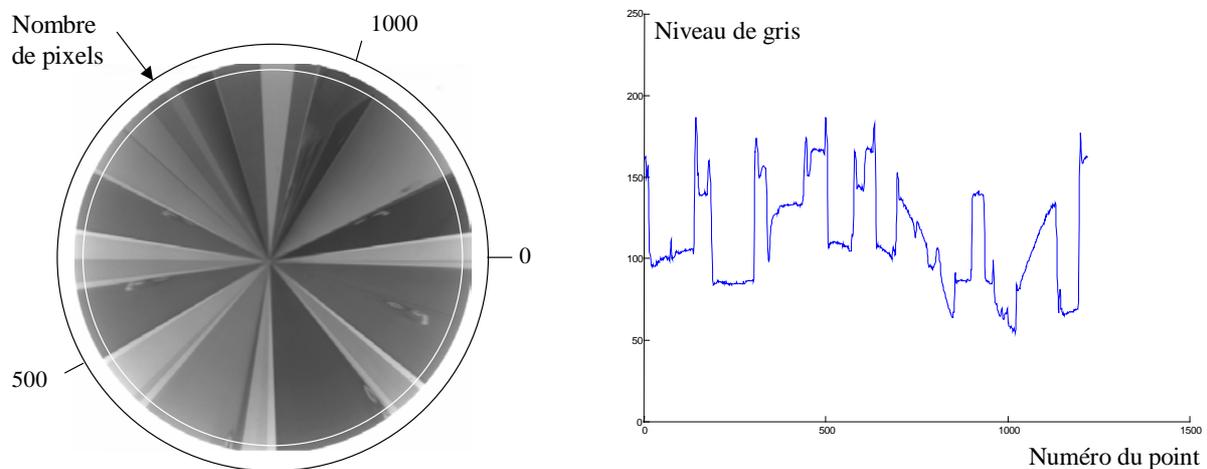


Figure 2-5 : Courbe des niveaux de gris pour une image panoramique prise dans l'un des environnements expérimentaux.

Il faut alors traiter ce signal monodimensionnel de niveaux de gris par rapport aux variations de contraste qui délimitent chaque secteur. Ce traitement devra fournir des résultats les plus proches possibles sur les deux images. Nous avons pour cela utilisé comme critère discriminant et robuste le gradient.

La segmentation que nous avons appliquée est basée sur le calcul de la dérivée discrète de la courbe des niveaux de gris. Afin de limiter l'importance des bruits, nous avons décidé d'utiliser un vecteur de dimension sept : $[-2 \ -2 \ -1 \ 0 \ 1 \ 2 \ 2]$. Il suffit ensuite de déterminer un

seuil permettant de séparer l'information essentielle du reste du signal. Le résultat obtenu par seuillage de cette courbe est donné par la Figure 2-6.

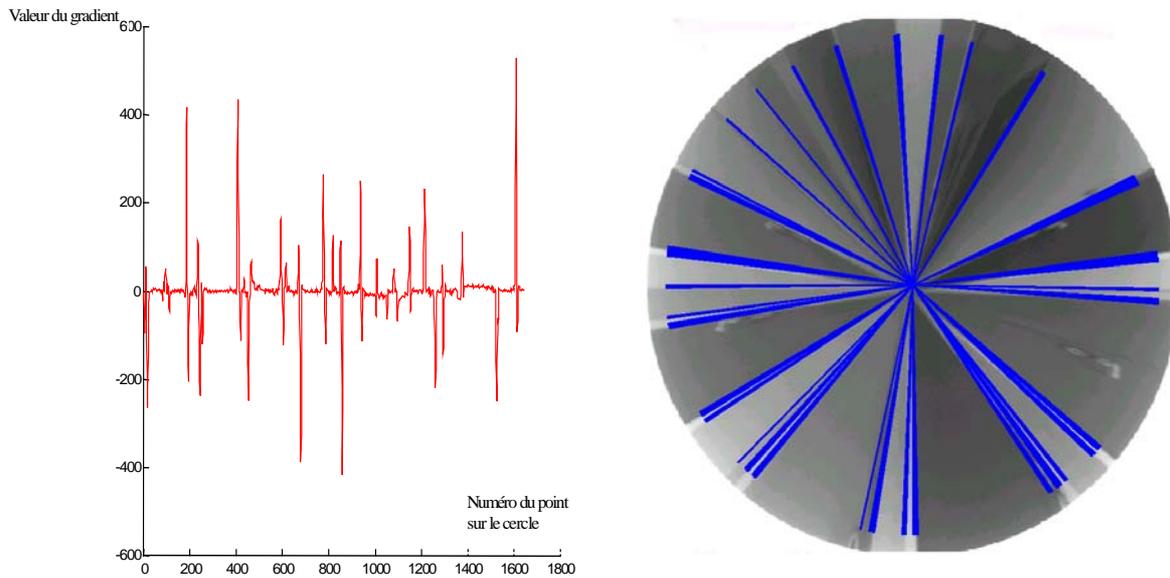


Figure 2-6 : Courbe de gradient et segmentation en résultant.

Le problème de cette approche très classique est qu'à chaque changement de secteur, on obtient un grand nombre de droites radiales. Nous avons donc intégré un post-traitement qui filtre la courbe de gradient. Ce traitement consiste à lisser la courbe en cherchant les maximums et les minimums locaux pour ensuite passer les autres points à 0.

Un schéma explicatif de ce post-traitement est donné en Figure 2-7, la courbe avant traitement comporte une multitude de valeurs au dessus du seuil fixé. Après l'application du filtre, nous constatons que seules les valeurs caractéristiques de la courbe ont été conservées.

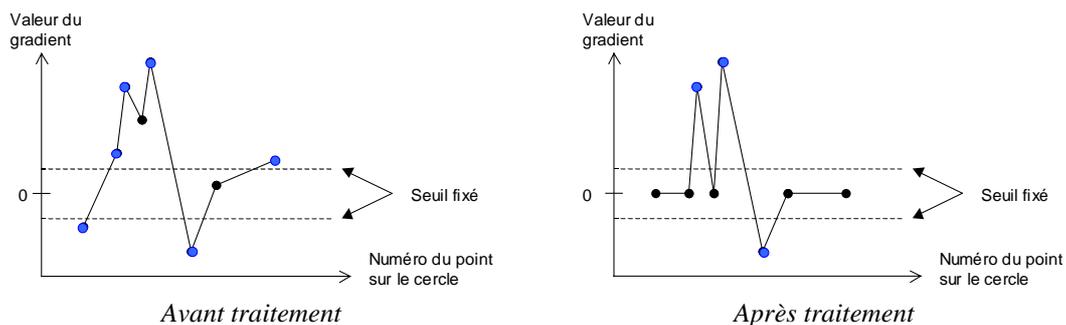


Figure 2-7 : Post-traitement appliqué à la courbe de gradient.

Malgré la bonne qualité de la segmentation obtenue, certaines droites radiales parasites peuvent apparaître, dues par exemple à la détection d'une poignée de porte (Figure 2-8). L'étape de segmentation étant primordiale pour la suite du traitement, il nous a semblé nécessaire d'envisager un traitement complémentaire. Pour cela, nous utilisons un deuxième cercle concentrique, d'un rayon plus petit que le premier, sur lequel nous effectuons le même traitement. Ce deuxième signal est utilisé en complément du premier de la manière suivante : lorsque sur le premier signal aucune droite radiale n'est détectée, il n'y a alors aucune ambiguïté. Par contre, si le seuillage détecte un amer vertical, la deuxième courbe est

consultée, pour s'assurer de la validité de l'information. Si l'information existe bien dans les deux courbes, la création de la droite est validée ; dans le cas contraire, il s'agit d'une droite parasite et l'information n'est pas utilisée.

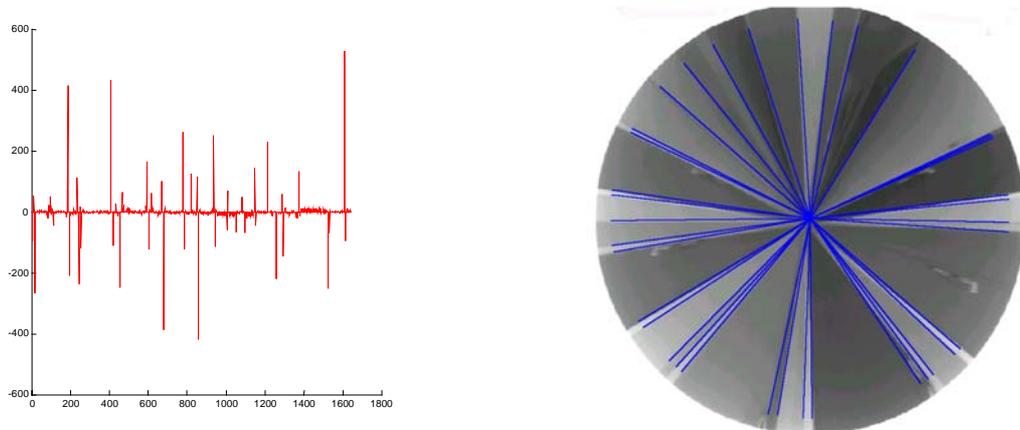


Figure 2-8 : Exemple de courbe de gradient et de segmentation résultante.

La dernière information à fixer précisément est la valeur du seuil utilisée. Cette valeur influence également sur le nombre de primitives détectées. Nous avons donc décidé de ne pas utiliser un seuil fixe mais de le calculer à partir de l'information de variation du niveau de gris. Les deux valeurs que nous utilisons pour fixer ce seuil sont :

- La moyenne prise sur la valeur absolue de la courbe de gradient, qui donne l'information de variation moyenne des niveaux de gris dans l'image. Une valeur en dessous de cette moyenne signifie que l'information n'est pas prépondérante.
- L'écart type, qui représente les fluctuations autour de cette moyenne, et reflète donc le contraste de l'image. En effet, une image composée de niveaux de gris relativement uniforme donnera un écart type faible.

Afin d'avoir un seuil prenant en compte la particularité de chaque image (contraste), nous fixons la valeur de ce seuil à :

$$\text{Seuil} = \text{Moyenne} + \text{Ecart type} \quad [2-1]$$

Cet ensemble de post-traitements permet d'obtenir une segmentation robuste et cohérente des secteurs de niveaux de gris. Les résultats de la Figure 2-9 montrent que nous avons une segmentation tolérante aux bruits et aux différences de luminosité, nous permettant d'extraire uniquement les informations essentielles. Cette robustesse permettra bien sûr d'obtenir un appariement cohérent des secteurs de niveaux de gris entre les deux images.

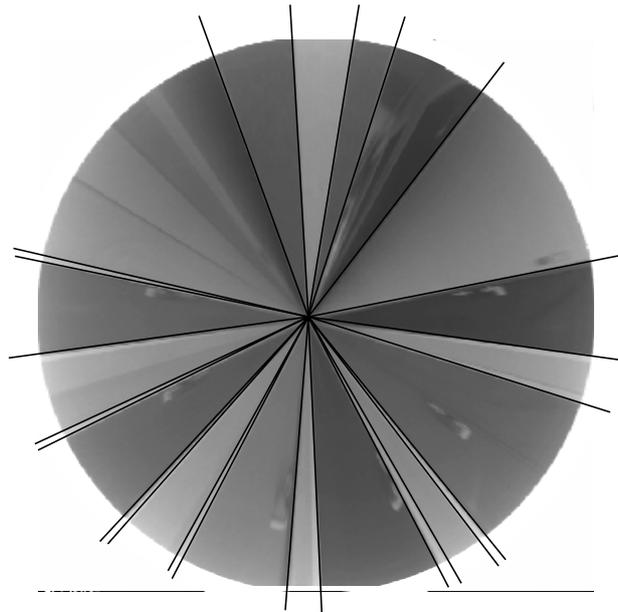


Figure 2-9 : Exemple de segmentation résultant de la méthode utilisée.

La phase de segmentation génère ainsi dans les deux images un nombre de secteurs à peu près identique, ce qui peut favoriser la mise en correspondance. Ces tests montrent d'une part la robustesse de cette phase de bas niveau dont les autres vont dépendre, et d'autre part la précision importante avec laquelle sont obtenues les primitives sensorielles considérées : les droites radiales délimitant un secteur de niveaux de gris. Cette précision sera des plus prépondérante dans l'étape de calcul de position des amers qui sera basée sur la triangulation.

2.4. Appariement des secteurs

La phase de mise en correspondance consiste dans notre problème à appairer avec un maximum de robustesse les différents secteurs de niveaux de gris détectés dans les deux images. Nous utiliserons pour cela différents critères complémentaires que nous fusionnerons en utilisant les règles de combinaisons de Dempster-Shafer (Figure 2-10).

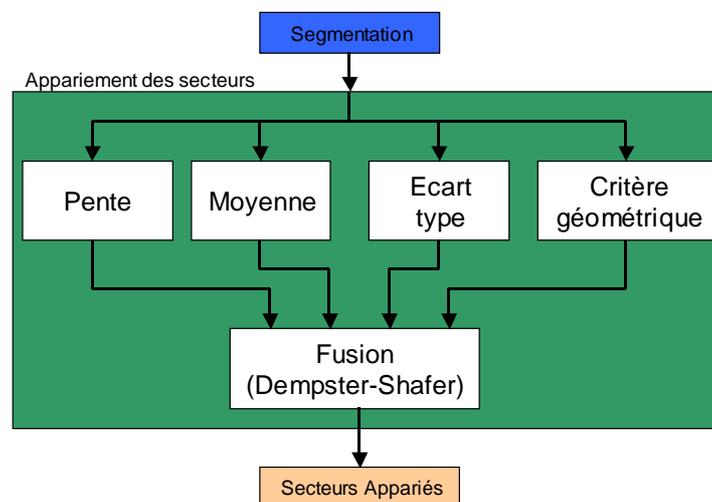


Figure 2-10 : Algorithme global d'appariement des secteurs.

Les deux cônes du système étant décalés l'un par rapport à l'autre, les amers ne peuvent apparaître sur les deux images de façon identique. En effet, certains secteurs peuvent être observés sur une image mais pas sur la deuxième, à cause de la configuration de la pièce (cf. Figure 2-11, la succession des secteurs sur le premier cône sera 1-2-3-5 alors que sur le deuxième ce sera 1-3-4-5, ceci à cause d'un problème d'occultation) ou simplement parce qu'ils n'ont pas été détectés par l'une des deux segmentations. Ceci implique que les secteurs ne se succéderont pas systématiquement dans le même ordre sur les deux images : il nous faut donc trouver un ensemble de critères les plus discriminant possible et autres que ceux liés à l'ordre d'apparition.

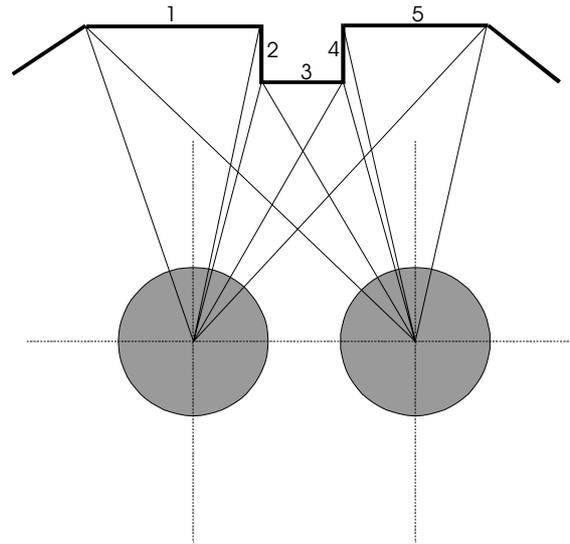


Figure 2-11 : Différence de projection d'un amer suivant l'environnement.

Pour résoudre le problème d'appariement, nous avons listé des critères de comparaison pour chacune des deux images. Nous en avons retenu quatre qui semblent robustes et intuitifs :

- ✓ La pente de la droite qui approxime au mieux l'ensemble des points du secteur sur la courbe des niveaux de gris (repère : numéro du point/niveau de gris).
- ✓ La moyenne du niveau de gris du secteur.
- ✓ L'écart type du niveau de gris du secteur.
- ✓ Les contraintes géométriques du secteur, qui peuvent être assimilées à « géométrie épipolaire une dimension simplifiée »

2.4.1 La théorie des croyances de Dempster-Shafer

Cette théorie est non seulement une théorie de l'évidence car elle utilise des poids d'évidence et des degrés numériques de support basés sur l'évidence, mais également une théorie du raisonnement probable car elle donne une combinaison d'évidence [DEM67] [SHA76]. Cette théorie est dite subjective car elle repose sur le jugement d'un individu et de plus elle traite de crédibilité et de plausibilité plutôt que de probabilité. La méthode de Dempster-Shafer permet également de représenter l'ignorance d'une connaissance et ainsi de fonctionner avec des connaissances partielles.

La théorie de Dempster-Shafer débute en supposant un cadre de discernement Θ qui est un ensemble d'alternatives mutuellement exclusives. Ainsi, ce cadre peut être constitué des valeurs possibles pour un attribut quelconque. Par exemple, si on désire diagnostiquer une maladie chez un patient, le cadre de discernement Θ sera l'ensemble des maladies possibles. On peut alors définir la notion de fonction de masse m qui représente la mesure de croyance attribuée à un sous-ensemble A de Θ . Cette fonction est également appelée l'assignement de probabilité basique. Ainsi si Θ est un cadre de discernement alors une fonction $m : 2^\Theta \rightarrow [0,1]$ est appelée assignement de probabilité basique ou fonction de masse si :

$$\begin{aligned} m(\emptyset) &= 0 \\ \text{et } \sum_{A \subset \Theta} m(A) &= 1 \end{aligned} \quad [2-2]$$

Ainsi aucune croyance ne doit être accordée à l'ensemble vide et une croyance totale possède comme valeur, 1. Cette quantité mesure la croyance que l'on assigne exactement à A , et non pas la croyance totale que l'on assigne à A . Afin d'obtenir cette mesure de croyance totale assignée à A , il est nécessaire d'ajouter à $m(A)$ les quantités $m(B)$ de tous les sous-ensembles propres B de A :

$$\text{Bel}(A) = \sum_{B \subset A} m(B) \quad [2-3]$$

Si Θ est un cadre de discernement, alors une fonction $\text{Bel} : 2^\Theta \rightarrow [0,1]$ (également noté Cr) est une fonction de croyance si et seulement si elle satisfait les conditions suivantes :

- $\text{Bel}(\emptyset) = 0$
- $\text{Bel}(\Theta) = 1$
- \forall entier $n > 0$ et toute collection A_1, A_2, \dots, A_n de sous-ensembles de Θ alors :

$$\text{Bel}(A_1 \cup \dots \cup A_n) \geq \sum_{\substack{I \subset \{1, \dots, n\} \\ I \neq \emptyset}} (-1)^{|I|+1} \text{Bel}(\bigcap_{i \in I} A_i) \quad [2-4]$$

De la même façon, il est possible de définir une fonction de plausibilité définie comme :

$$\text{Pl}(A) = \sum_{B \cap A = \emptyset} m(B) = 1 - \text{Bel}(\bar{A}) \quad [2-5]$$

Ainsi, $1 - \text{Pl}(A)$ correspond à la mesure du doute en A , notée $\text{Dou}(A) = \text{Bel}(\bar{A})$. Un sous-ensemble A d'un cadre de discernement Θ est appelé un élément focal d'une fonction de croyance Bel sur Θ si $m(A) > 0$. L'union de tous les éléments focaux d'une fonction de croyance est appelée son noyau N . De cette façon :

$$N = \bigcup_{X/m(X) > 0} X \quad [2-6]$$

Supposons que Θ soit constitué de $\{H, C, P\}$ et que

$$\begin{aligned} m(\{H\}) &= 0.3 \\ m(\{H, C\}) &= 0.2 \\ m(\{H, C, P\}) &= 0.5 \end{aligned}$$

alors

$$\begin{array}{ll} \text{Bel}(\{H\})=0.3 & \text{Pl}(\{H\})=1.0 \\ \text{Bel}(\{H, C\})=0.5 & \text{Pl}(\{H, C\})=1.0 \\ \text{Bel}(\{H, P\})=0.3 & \text{Pl}(\{H, P\})=1.0 \\ \text{Bel}(\{H, C, P\})=1.0 & \text{Pl}(\{H, C, P\})=1.0 \end{array}$$

La quantité $Q(A) = \sum_{\substack{B \subset \Theta \\ A \subset B}} m(B)$ mesure la masse de probabilité totale qui peut se déplacer librement à chaque point de A (Notons que $Q(\emptyset) = 1$). $Q(A)$ est appelé le nombre de communalité pour A, et la fonction $Q : 2^\Theta \rightarrow [0,1]$ correspond à la fonction de communalité pour Bel.

Les fonctions de croyance sont adaptées à la représentation de l'évidence car elles admettent une véritable règle de combinaison. Etant donné plusieurs fonctions de croyance sur le même cadre de discernement mais basées sur des corps d'évidence distincts, la règle de Dempster permet de calculer leur somme orthogonale, qui résulte en une nouvelle fonction de croyance basée sur l'évidence combinée.

Si $k = \sum_{\substack{B, C \\ B \cap C = \emptyset}} m_1(B)m_2(C) < 1$ on obtient alors la règle suivante :

$$(m_1 \oplus m_2)(A) = \frac{\sum_{\substack{B, C \\ A = B \cap C}} m_1(B) m_2(C)}{1 - \sum_{\substack{B, C \\ B \cap C = \emptyset}} m_1(B) m_2(C)} \quad [2-7]$$

k représente la mesure du conflit, on a en particulier les valeurs suivantes :

- ✓ $k=1$ signifie que les deux sources d'informations sont totalement en conflit. Dans ce cas, la combinaison est impossible. Il est alors nécessaire de revoir le problème.
- ✓ $k=0$ signifie que les sources sont totalement concordantes.

Par exemple, si $\Theta = \{D, D'\}$, et

$$\begin{array}{lll} m_1(\{D\})=0.8 & m_1(\{D'\})=0 & m_1(\{D, D'\})=0.2 \\ m_2(\{D\})=0.9 & m_2(\{D'\})=0 & m_2(\{D, D'\})=0.1 \end{array}$$

il est possible de créer la table suivante :

			m_2		
			0.9	0	0.1
			{D}	{D'}	{D, D'}
m_1	0.8	{D}	0.72	0	0.08
	0	{D'}	0	0	0
	0.2	{D, D'}	0.18	0	0.02

ainsi,

$$\begin{aligned}
 m_1+m_2(\{D\}) &= 0.72+0.08+0.18=0.98 \\
 m_1+m_2(\{D'\}) &= 0 \\
 m_1+m_2(\{D, D'\}) &= 0.02
 \end{aligned}$$

Pour la combinaison de n sources différentes, on désire alors calculer la relation suivante :

$$\left(\left(\dots (m_1 \oplus m_2) \dots \right) \oplus m_{n-1} \right) \oplus m_n$$

Ainsi,

$$k_n = \sum_{X_1 \cap X_2 \cap \dots \cap X_n = \emptyset} m_1(X_1)m_2(X_2) \dots m_n(X_n) \tag{2-8}$$

$$\bigoplus_{i=1}^n m_i(A) = \frac{1}{1 - k_n} \sum_{X_1 \cap X_2 \cap \dots \cap X_n = A} m_1(X_1)m_2(X_2) \dots m_n(X_n) \tag{2-9}$$

Soit

$$\bigoplus_{i=1}^n Q_i(A) = \frac{1}{1 - k_n} Q_1(X_1)Q_2(X_2) \dots Q_n(X_n) \tag{2-10}$$

Finalement, plusieurs critères permettent de prendre une décision finale :

- ✓ Croyance maximale : La crédibilité constitue le degré de croyance minimal que l'on attribue à une hypothèse. Cette loi permet de déterminer la meilleure hypothèse en donnant le minimum de chances à chacune d'elles. C'est donc un critère de décision "pessimiste".
- ✓ Plausibilité maximale : La plausibilité constitue le degré de confiance maximal que l'on attribue à une hypothèse. Ce critère permet donc de déterminer la meilleure hypothèse en donnant le maximum de chances à chacune d'elles. C'est donc un critère de décision "optimiste".
- ✓ Autre : Il existe de multitudes règles de décisions que nous ne détaillerons pas ici, et qui peuvent être choisies en fonction du problème à résoudre.

2.4.2 Calcul des critères d'appariement

Critère 1 : paramètres de la droite d'approximation

Le premier critère d'appariement est calculé à partir de la courbe des niveaux de gris, et plus particulièrement sur la zone correspondant au secteur fourni par l'étape de segmentation.

On approxime par une droite, avec la méthode des moindres carrés, la portion de courbe correspondant à chaque secteur sur la courbe des niveaux de gris. En effet, la variation des niveaux de gris est linéaire pour chaque secteur. On peut constater expérimentalement que la pente de cette droite varie peu entre les deux images, ceci pour un secteur donné. On peut donc affirmer que, qualitativement, cette modélisation est un critère d'appariement robuste (Figure 2-12).

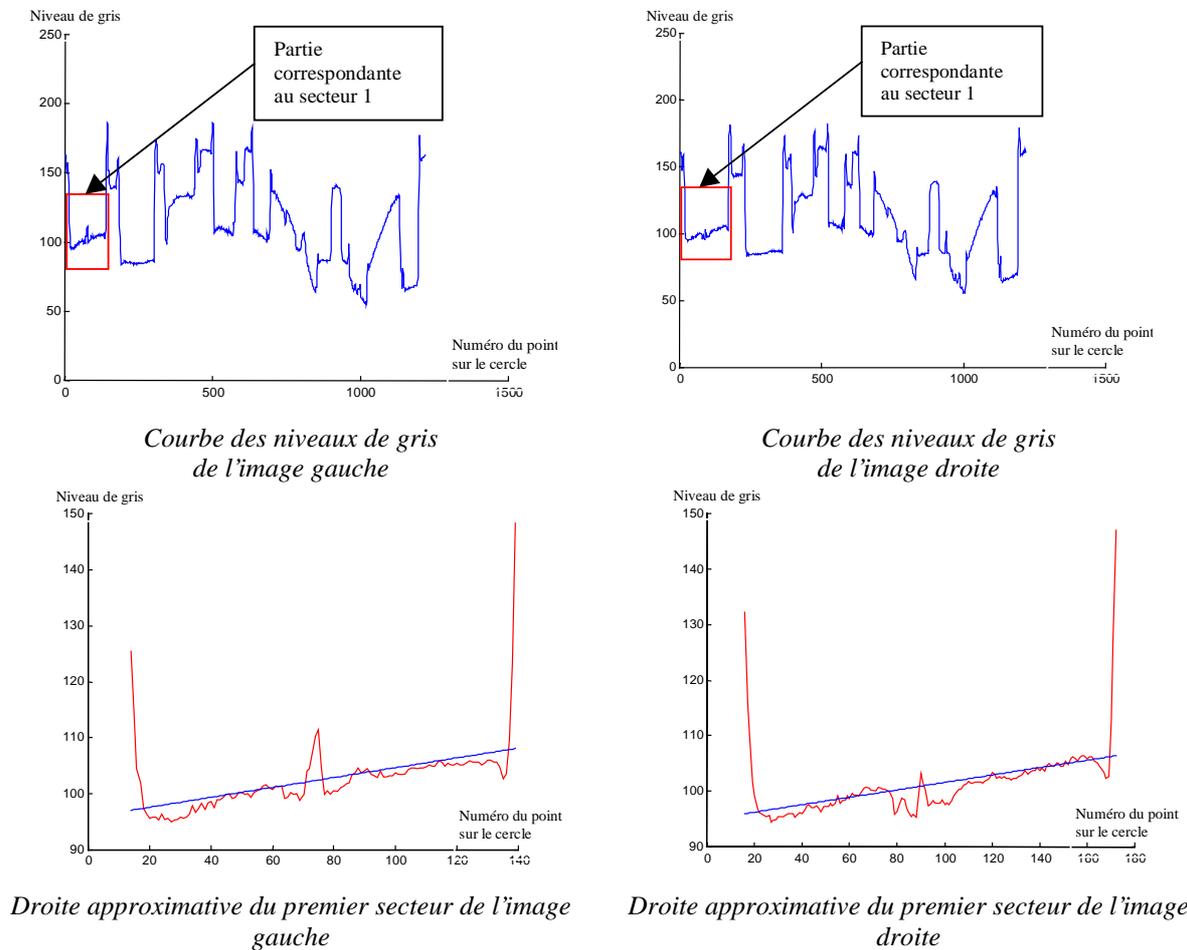


Figure 2-12 : Exemple d'invariance du premier critère d'appariement.

Critère 2 : moyenne des niveaux de gris

Le deuxième critère choisi est la moyenne des niveaux de gris de l'ensemble des points qui forment un secteur sur une image. Là encore on constate expérimentalement une similitude importante de cette grandeur sur deux secteurs associés, et ce même lorsque l'angle d'ouverture des secteurs diffère de façon importante.

Critère 3 : écart type sur les niveaux de gris

Le troisième critère est l'écart type des points de la courbe des niveaux de gris pour chaque secteur. Il est un complément intuitif et logique du deuxième puisque représentant la dispersion autour de la grandeur précédemment calculée.

Critère 4 : contraintes géométriques

Le dernier critère retenu est lié aux contraintes géométriques "une dimension" existant entre les secteurs des deux images. Ces contraintes géométriques peuvent être vues comme une simplification de la géométrie épipolaire au cas "une dimension". On considère le plan contenant la base des deux cônes A et B de rayon r , éloignés d'une distance d . Ce plan est supposé parallèle au plan sur lequel le robot se déplace. Nous considérons que l'origine du repère du robot se situe sur le centre du cône A et que les deux cônes sont alignés sur l'axe Ox (Figure 2-13).

Deux cas de figures peuvent être envisagés :

- ✓ L'amer considéré est détecté dans le même quart de cercle sur les deux images, mais avec deux angles différents,
- ✓ L'amer considéré est détecté dans des quarts de cercle dont les zones sont adjacentes (Figure 2-13).

Un amer vu à droite par le capteur droit est forcément vu à droite par le capteur gauche. Dans cette optique on remarquera que systématiquement l'angle formé par chaque côté du secteur par rapport à l'axe Ox du cône droit, devra être supérieur à l'angle de chaque côté correspondant à ce secteur sur le cône gauche (Figure 2-13).

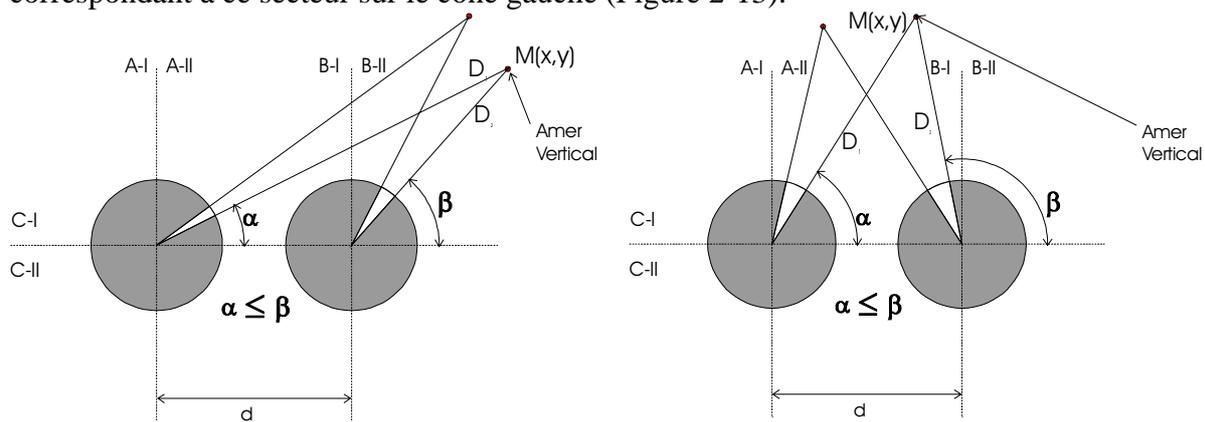


Figure 2-13 : Contraintes géométriques pour l'appariement des secteurs.

2.4.3 Fusion des critères d'appariement

L'étape suivante consiste donc à fusionner ces quatre critères. L'hétérogénéité de ceux-ci justifie ici l'utilisation de la théorie de Dempster Shafer pour réaliser cette fusion.

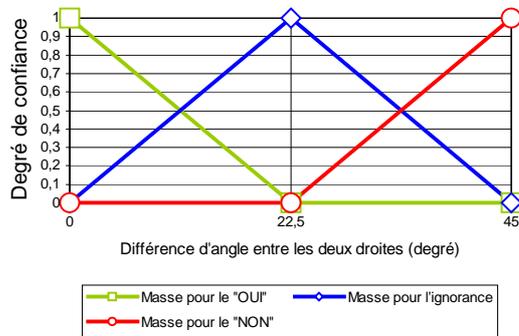
Lorsque nous cherchons à appairer un secteur d'une image à un secteur de l'autre image, nous souhaitons avoir une réponse positive ou négative. C'est pour cette raison que notre cadre de discernement est constitué de deux éléments :

- ✓ "OUI" pour appairer ces deux secteurs,
- ✓ "NON" pour ne pas appairer ces deux secteurs.

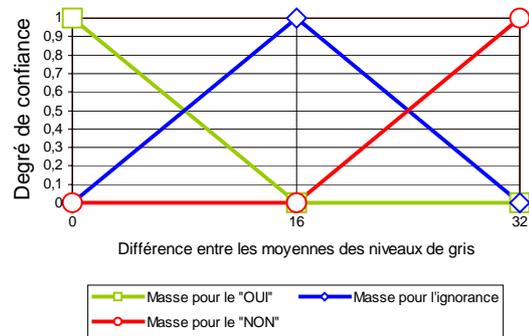
En considérant les valeurs des quatre critères d'appariement, nous pouvons établir les fonctions de masse de chacun de ces capteurs pour chaque élément du cadre de discernement. Les trois premières fonctions correspondent à la différence des trois premiers critères entre le secteur γ de l'image gauche et le secteur λ de l'image droite.

La dernière fonction est donnée par le critère géométrique suivant :

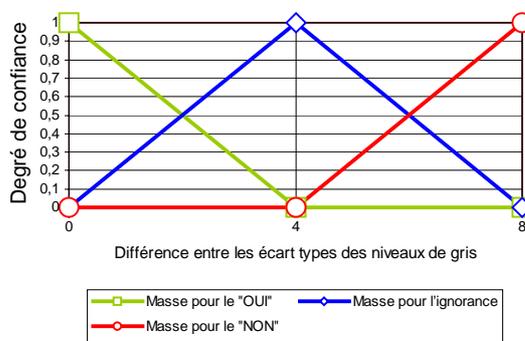
- ✓ Si le secteur γ peut géométriquement être apparié au secteur λ , nous accordons alors un degré de confiance pour le "OUI" égal à 0.5,
- ✓ sinon nous lui accordons un degré de confiance égal à 0.



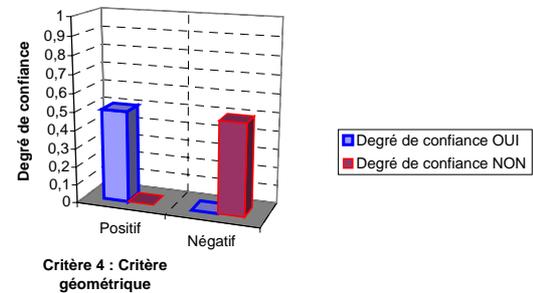
Critère 1



Critère 2



Critère 3



Critère 4

Figure 2-14 : Fonctions de croyance pour l'étape d'appariement des secteurs.

La détermination des fonctions de croyance nous permet de calculer les fonctions de masse et la mesure de conflit k entre nos deux éléments du cadre de discernement et l'ignorance avec les règles de combinaisons de Dempster-Shafer.

2.4.4 Algorithme global d'appariement des secteurs

Nous savons que le critère géométrique réduit la recherche des secteurs pouvant être appariés, mais aussi qu'un secteur ne peut être apparié qu'une seule fois. Dans cette optique il nous faut éliminer des associations au fur et à mesure de la scrutation des secteurs dans une image. L'algorithme général d'appariement se compose de trois modules (Figure 2-15) :

- ✓ On calcule d'abord la mise en correspondance des secteurs qui ont le critère géométrique le plus fort, c'est-à-dire les secteurs de la zone A-I. Ils doivent alors être appariés avec des secteurs de la zone B-I.
- ✓ Ensuite, ce sont ceux de la zone B-II que nous cherchons à appairer avec des secteurs de la zone A-II.
- ✓ Et enfin, nous terminons par la zone A-II dont les secteurs doivent être appariés avec ceux restants de la zone B-I.

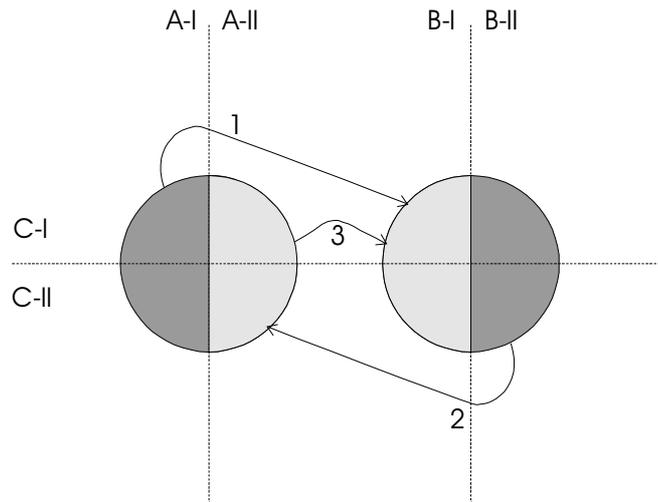


Figure 2-15 : Ordre de scrutation pour l'appariement des secteurs.

Pour commettre le minimum d'erreurs d'appariement, une solution est retenue seulement si elle est unique, sinon aucune décision n'est prise et l'algorithme continue avec les autres secteurs. On appelle ici solution unique, la solution maximale dont le degré de confiance pour le OUI est supérieur à celui pour le NON et l'ignorance, mais aussi éloignée des autres solutions possibles (d'au moins 10%). A chaque appariement, la phase en cours est réitérée. En effet, un appariement peut permettre de lever des ambiguïtés antérieures d'appariement. Dans le cas où l'ignorance est maximum, nous préférons ne pas porter de choix sur la fusion. En effet, nous considérons qu'il est préférable d'avoir moins de primitives sensorielles mais de meilleure qualité.

Un exemple final d'appariement est présenté sur la Figure 2-16, où nous constatons qu'un grand nombre de secteurs sont correctement appariés. Toutefois, nous voyons apparaître un problème que devra prendre en compte la phase de localisation suivante. En effet, les deux secteurs 13 (qui sont correctement appariés) ne représentent pas tout à fait la même information. Ceci est une conséquence de la phase de segmentation qui a détecté dans l'image droite une droite radiale supplémentaire que sur l'image gauche.

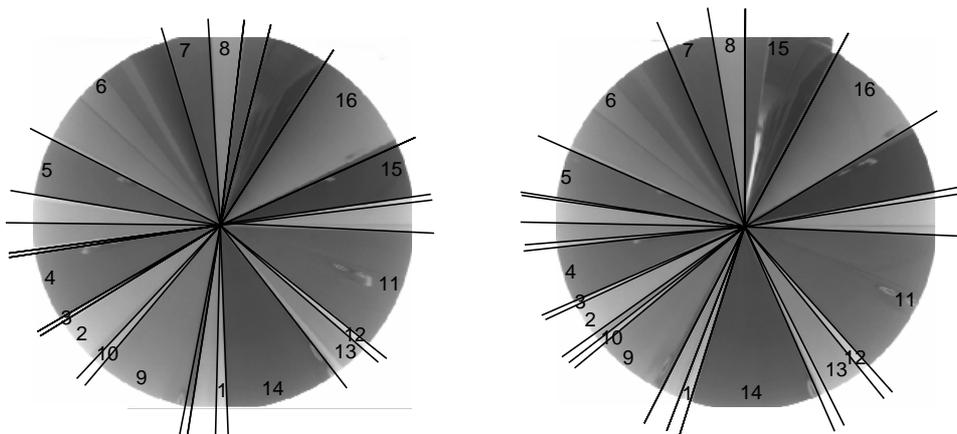


Figure 2-16 : Segmentation et appariement final des secteurs pour les deux images correspondant à une acquisition stéréoscopique.

2.5. Calcul des points de l'environnement dans le modèle sensoriel

L'appariement des secteurs entre eux réalisé, il nous reste à calculer les coordonnées des points du segment qu'ils représentent. Pour cela, nous connaissons l'angle d'orientation des droites support des cotés des secteurs et la distance qui sépare les deux cônes.

Les coordonnées de tous les points dans le repère du capteur (situé sur le centre du cône A) sont obtenues par triangulation avec les formules suivantes :

$$x = \frac{d \times \tan(\beta)}{\tan(\beta) - \tan(\alpha)} \quad y = \frac{d \times \tan(\beta) \times \tan(\alpha)}{\tan(\beta) - \tan(\alpha)} \quad [2-11]$$

Comme nous l'avons déjà expliqué, cette solution « élémentaire » ne tient pas compte des paramètres de la caméra. Rappelons à ce niveau que les travaux sur la calibration du système de vision SYCLOP ont été réalisés dans [CAU01].

2.6. Algorithmes de localisation

2.6.1 Les primitives considérées

Nous avons pu constater que nous obtenons un modèle sensoriel où deux types de primitives sensorielles peuvent être considérés :

- ✓ Les points, caractérisant chaque amer vertical,
- ✓ les segments dont les extrémités représentent des amers verticaux.

Pour localiser le robot, nous avons donc deux choix possibles vis-à-vis de l'algorithme de mise en correspondance :

- ✓ Mettre en correspondance les points du modèle sensoriel avec les points répertoriés dans la carte théorique de l'environnement.
- ✓ Mettre en correspondance les segments du modèle sensoriel avec les segments répertoriés dans la carte théorique de l'environnement.

En théorie, chaque couple de secteurs appariés doit représenter un amer horizontal avec les deux extrémités trouvées. En fait, on peut constater qu'il existe de nombreux cas où ceci n'est pas vérifié. Par exemple, l'absence de paroi entre deux amers verticaux (ouverture de porte, couloir, ...) est caractérisée par un secteur et donc un segment. Dans ce cas particulier, en considérant des segments comme primitives sensorielles, il sera évidemment impossible de réaliser un appariement avec une primitive de la carte théorique de l'environnement (Figure 2-17).

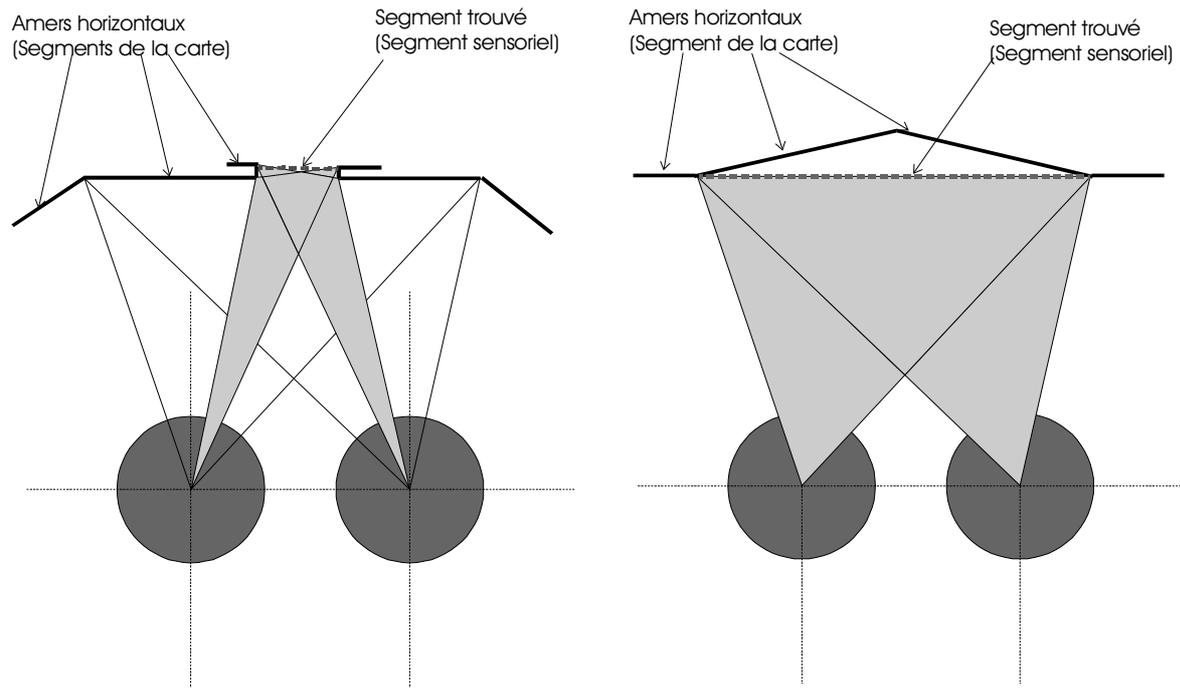


Figure 2-17 : Problèmes liés à la prise en compte de segments.

De plus, un couple de secteurs peut représenter un ensemble d'amers horizontaux : si le contraste entre plusieurs secteurs n'est pas suffisant, la phase de segmentation n'aura pas pu détecter la totalité des droites radiales (Figure 2-17). Enfin, une segmentation légèrement erronée sur une droite radiale caractérisant un secteur accroît de façon importante l'erreur commise sur les coordonnées du point considéré. On rappellera à ce niveau que cette erreur est limitée par la distance maximale au delà de laquelle on ne peut plus détecter de balises naturelles. D'une manière générale, on peut affirmer qu'une segmentation différente sur les deux images ou un mauvais appariement, aura plus d'influence sur l'entité segment que sur l'entité point. Toutes ces erreurs cumulées donnent dans certains cas un modèle sensoriel peu exploitable d'un point de vue segment, ce qui complique la mise en correspondance avec le modèle théorique. On pourra noter que ces erreurs sont imputables à la chaîne segmentation/appariement et non à un problème de calibration.

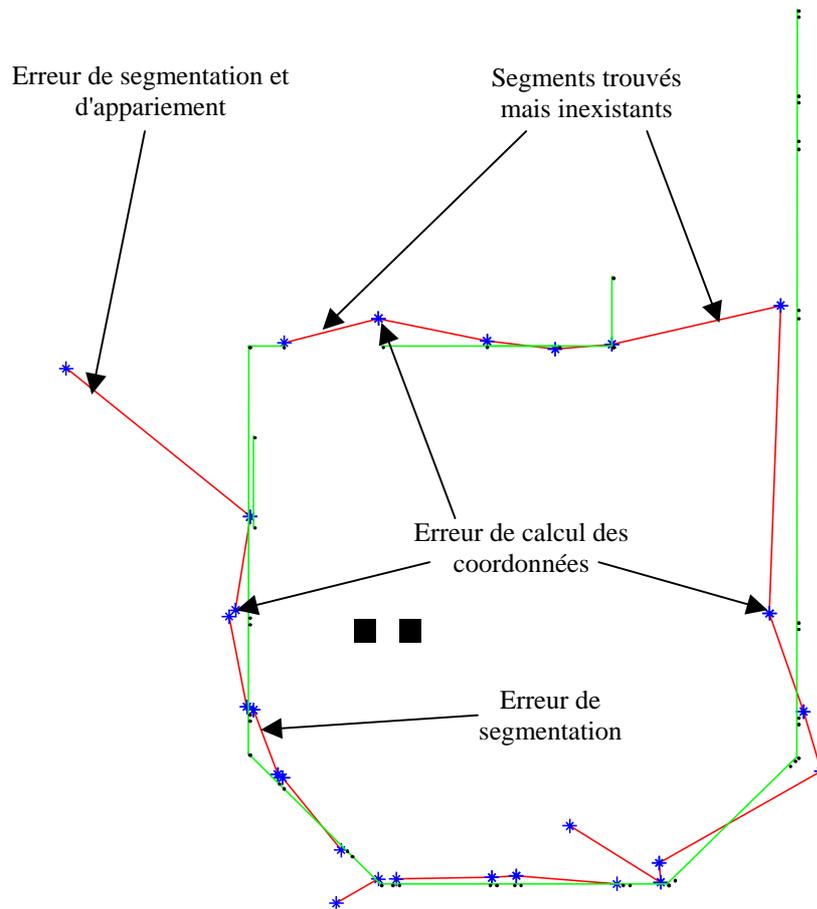


Figure 2-18 : Comparaison du modèle sensoriel en fonction des primitives considérées.

Les considérations précédentes nous ont amené à la conclusion qu'il fallait considérer des primitives géométriques plus simples que les segments, à savoir les points. Même si un segment est fortement entaché d'erreurs, au moins une de ses deux extrémités représente un point qui existe. Nous avons donc développé un algorithme de mise en correspondance du modèle sensoriel avec le modèle théorique de l'environnement qui sera composé de points.

2.6.2 Algorithmes de mise en correspondance

A ce niveau de l'étude la problématique qui se pose est finalement l'appariement d'un nuage de points avec un autre. Le premier est constitué par les observations et le second est quant à lui composé par l'intégralité des points de la carte de l'environnement. Cette problématique est connue dans la littérature et plusieurs méthodes peuvent ici être mises en jeu. Elle est notamment traitée dans les problèmes de mise en correspondance de deux images stéréoscopiques [HUT93] et consiste à appairer des points d'intérêt extraits de chacune d'elles. Un critère de correspondance souvent utilisé est la distance de Hausdorff. Pour estimer la position du robot dans son environnement nous avons développé et testé deux stratégies : une basée sur la distance de Hausdorff et l'autre basée sur la distance cartésienne entre deux points.

Appariement par rapport à la distance de Hausdorff

Nous cherchons ici la meilleure correspondance en utilisant la distance de Hausdorff comme critère de minimisation. Cette distance mesure l'ampleur entre tous les points d'un ensemble du "modèle" qui sont proches de quelques points d'un ensemble de "l'image" et vice versa. Donc, cette distance peut être utilisée pour déterminer le degré de ressemblance entre deux objets qui sont superposés l'un sur l'autre.

Soient deux ensembles finis $A=\{a_1, \dots, a_p\}$ et $B=\{b_1, \dots, b_q\}$, la distance de Hausdorff est définie comme

$$H(A, B) = \max(h(A, B), h(B, A)) \quad [2-12]$$

où

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad [2-13]$$

et $\|\cdot\|$ est une norme sous-jacente des ensembles de points A et B (par exemple, L_2 ou bien la norme Euclidienne).

La fonction $h(A, B)$ est appelée la distance de Hausdorff *orientée* de A vers B . Elle identifie le point $a \in A$ qui est le plus éloigné de tout point de B , et mesure la distance de a à son voisin dans B (en utilisant la norme donnée). En fait, elle a pour effet d'effectuer un classement de chaque point de a basé sur sa distance au point le plus proche de B . Ensuite, elle sélectionne le point de plus haut rang (le point le plus mal assorti de A). Intuitivement, si $h(A, B) = d$, alors chaque point de A doit être à une distance maximum de d de n'importe quel point de B , et certains points de A se trouvent exactement à une distance d du point le plus proche de B (le point le plus mal assorti).

La distance de Hausdorff, $H(A, B)$, est le maximum entre $h(A, B)$ et $h(B, A)$. Donc, elle mesure le degré de disparité entre deux ensembles, en mesurant la distance du point de A qui est le plus éloigné de tout point de B et vice versa. Intuitivement, si la distance de Hausdorff est d , alors chaque point de A doit être à une distance maximum de d de n'importe quel point de B et vice versa. Donc, la notion de ressemblance codée par cette distance est que chaque membre de A doit être proche de quelques membres de B et vice versa.

Pour estimer la mise en correspondance qui minimise la distance de Hausdorff, il est nécessaire de calculer cette distance pour toutes les positions possibles du robot sur une zone de solutions admissibles (Figure 2-19). Cette stratégie étant fortement combinatoire il est nécessaire de réduire au maximum l'ensemble des positions admissibles. Pour cela nous utilisons un fenêtrage fourni par le domaine d'erreur lié à l'estimation odométrique. Nous nous servons à ce niveau de la modélisation classique de l'imprécision odométrique gérée dans [WAN90] [DEL97]. Ainsi nous déterminons un domaine de configurations absolues possibles qui est obtenu par une majoration de l'ellipse d'erreur par un cercle. Le rayon de ce cercle est calculé en prenant la longueur du grand axe de l'ellipse d'erreur.

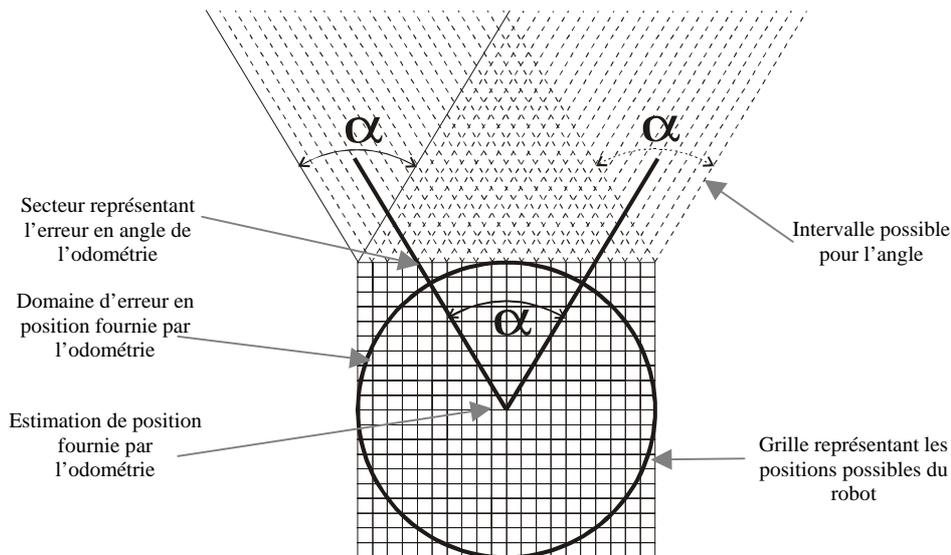


Figure 2-19 : Ensemble des positions admissibles par rapport à l'estimation odométrique.

Pour calculer la distance de Hausdorff nous gérons donc une grille discrétisée de points correspondant aux positions possibles du robot. Cette grille inclut le cercle précédent. Pour tout point (x,y) et pour un ensemble d'orientations possibles (là aussi discrétisées et bornées par l'erreur odométrique), nous calculons la distance de Hausdorff entre les points du modèle sensoriel et les points de la carte. Finalement, la distance minimale de Hausdorff fournit le meilleur appariement de points et donc la configuration optimale du robot dans son environnement.

Nous avons testé cette méthode de localisation sur plusieurs séries d'acquisitions. Nous en avons déduit que cette approche au sens de la distance de Hausdorff n'était pas adaptée à notre problème et ce pour trois raisons principales :

- ✓ Il y a généralement trop peu de points dans le modèle sensoriel pour garantir une grande robustesse,
- ✓ La distance de Hausdorff est très sensible au bruit et nos points sont souvent entachés d'une erreur.
- ✓ Le temps de calcul, même avec un fenêtrage issu de l'odométrie, reste très long et n'est donc pas adapté à un traitement en ligne.

Nous nous sommes tout naturellement tournés vers une autre stratégie de mise en correspondance basée sur un critère robuste : la proximité des points au sens de la distance cartésienne.

Appariement par rapport à la distance cartésienne

Le critère de convergence de l'algorithme de mise en correspondance le plus robuste que nous avons retenu est basé sur la distance entre chaque point observé et le point le plus proche de la carte de l'environnement.

La superposition du modèle sensoriel avec la carte de l'environnement nous a permis de constater qu'une majorité de points avaient leurs coordonnées déterminées de façon très précise (Figure 2-18). C'est ce qui justifie le choix intuitif du critère de convergence qu'est la distance cartésienne. Le principe de l'algorithme est le suivant :

- ✓ On considère deux points du modèle sensoriel.
- ✓ On cherche deux points du modèle théorique correspondants aux deux points choisis.
- ✓ Enfin, on calcule pour l'ensemble des points restants du modèle sensoriel, la distance qui les sépare des points les plus proches dans le modèle théorique.

Le problème de cet algorithme est lié à son aspect fortement combinatoire. Afin de réduire cette complexité et donc le temps de calcul, nous avons décidé de ne pas considérer tous les couples, mais uniquement les deux points issus de l'appariement d'un même secteur. Ceci permet de réduire le nombre de couples possibles.

Pour trouver le couple de points dans le modèle théorique correspondant le mieux à celui du modèle sensoriel, nous calculons la distance qui sépare ces points (Figure 2-20). Puis nous comparons les deux distances trouvées (sensorielle et théorique), ceci pour tous les couples de points du modèle théorique. Une fois qu'une solution a été trouvée, nous calculons l'angle de rotation qui est à appliquer à l'ensemble des points restants.

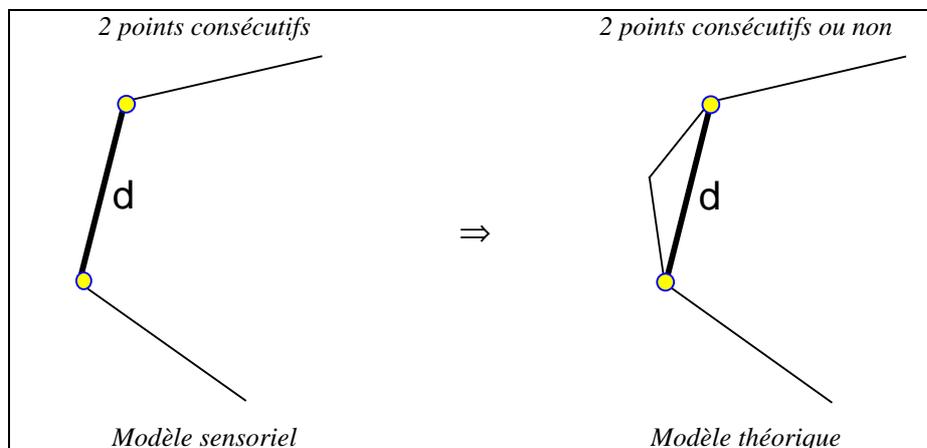


Figure 2-20 : Appariement de deux points du modèle théorique avec deux points du modèle sensoriel par rapport au critère de distance.

Il reste ensuite à calculer, pour chaque point restant du modèle sensoriel, la distance minimum qui les séparent d'un point du modèle théorique. C'est la distance cumulée qui nous permet ensuite de faire un choix sur le meilleur appariement modèle sensoriel-modèle théorique (Figure 2-21). Notre critère de sélection de la solution finale est une erreur en distance cumulée minimum.

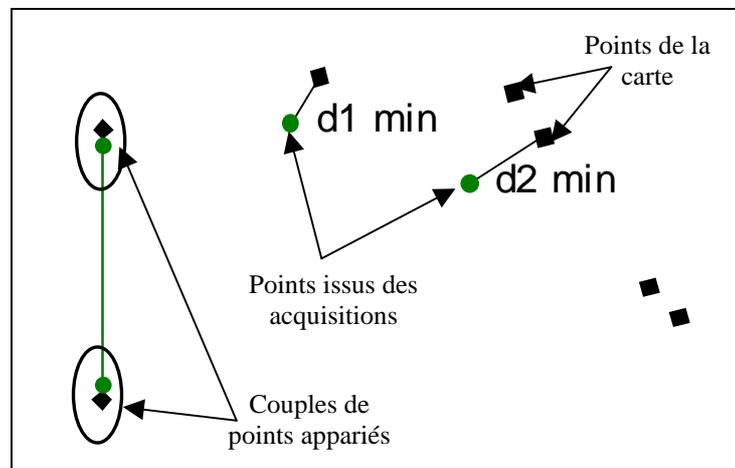


Figure 2-21 : Calcul du critère de convergence cumulée.

Malgré le bon fonctionnement de cet algorithme, le temps de traitement pour obtenir une position du robot reste encore élevé (plusieurs minutes). Pour cette raison, mais aussi pour limiter la possibilité d'obtenir une position de robot incompatible avec la position précédente, nous avons décidé d'utiliser l'information odométrique. Comme avec la méthode utilisant la distance d'Hausdorff, nous utilisons cette information uniquement comme initialisation de l'espace de recherche de notre algorithme. Il faut préciser ici que même si nous utilisons un modèle classique de modélisation odométrique, le modèle d'erreur est par contre maximisé pour ne pas influencer de manière négative l'algorithme.

Voici finalement l'algorithme global de localisation utilisé :

```

Distance_Cumulée_Retenue = Infini;
Pour tout secteur apparié s faire,
    Distancel = distance entre les deux points trouvés du secteur Psens1 et Psens2;
    Pour tout point Pthéo1 du modèle théorique faire,
        Pour tout point Pthéo2 du modèle théorique faire,
            Distance2 = distance entre les points Pthéo1 et Pthéo2;
            Distance_Cumulée_En_Cours = 0;
            Si Distancel = Distance2 ( $\pm$  DistanceMini) faire,
                Calculer AngleRotation = Angle de rotation entre le segment [Psens1;Psens2] et le
                    segment [Pthéo1;Pthéo2];
                Calculer TranslationPoint = Translation à effectuer pour placer Pthéo1 sur Psens1;
                Si AngleRotation et TranslationPoint compatible avec l'odométrie
                    Pour tout point Psens3 du modèle sensoriel faire,
                        Distance3 = infini;
                        Pour tout point Pthéo3 du modèle théorique faire,
                            Calculer la rotation de Pthéo3 de centre Pthéo1;
                            Calculer la translation de Pthéo3 avec TranslationPoint;
                            Calculer Distance4 = distance entre Pthéo3 et Psens3;
                            Si Distance4 < Distance3 faire,
                                Distance3 = Distance4;
                            Fin si
                        Fin pour
                    Distance_Cumulée_En_Cours = Distance_Cumulée_En_Cours+Distance3
                Fin pour
            Si Distance_Cumulée_En_Cours < Distance_Cumulée_Retenue faire,
                Distance_Cumulée_Retenue = Distance_Cumulée_En_Cours;
                SolutionAngle = AngleRotation;
                SolutionTranslation = TranslationPoint;
            Fin si
        Fin si
    Fin si
Fin pour
Calculer Position_capteur = la position Xc, Yc et l'orientation  $\theta_c$  du capteur dans le repère
                            du modèle théorique avec SolutionAngle et
                            SolutionTranslation;

```

2.7. Résultats expérimentaux

Nous avons testé la robustesse de notre algorithme de localisation sur plusieurs acquisitions bisensorielles, réalisées sur des déplacements élémentaires du robot, dans une pièce polyédrique de type hall d'environ 5 mètres de coté (Figure 2-22). Les acquisitions sont réalisées lorsque le robot est à l'arrêt.

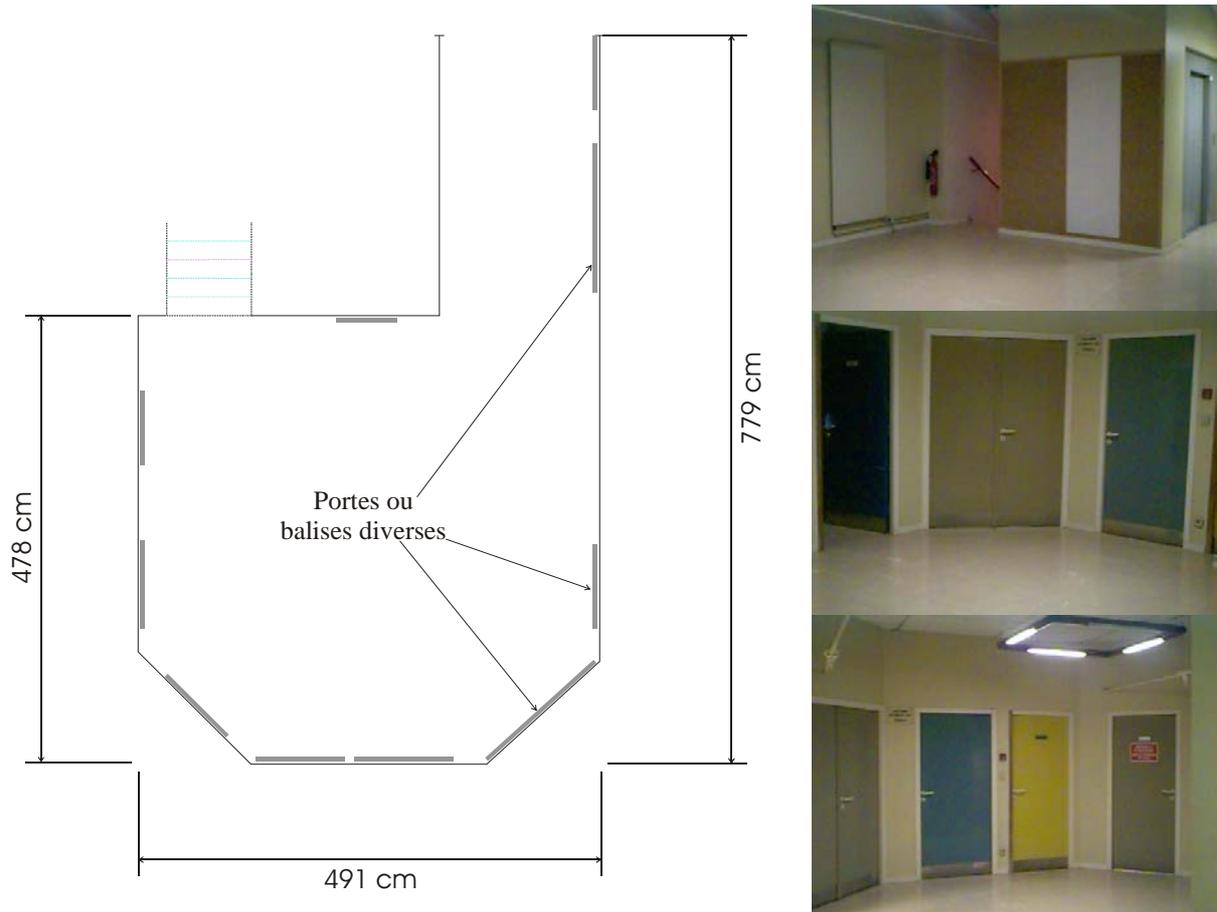


Figure 2-22 : Environnement d'expérimentation.

Nous présentons sur la Figure 2-23 un récapitulatif de la trajectoire effectuée comportant 7 acquisitions, les estimations de positions du robot ainsi que les erreurs odométriques associées. Ensuite nous présentons chaque acquisition une par une, avec :

- La segmentation des deux images,
- Le modèle sensoriel correspondant superposé à l'environnement réel,
- La localisation estimée par notre algorithme.

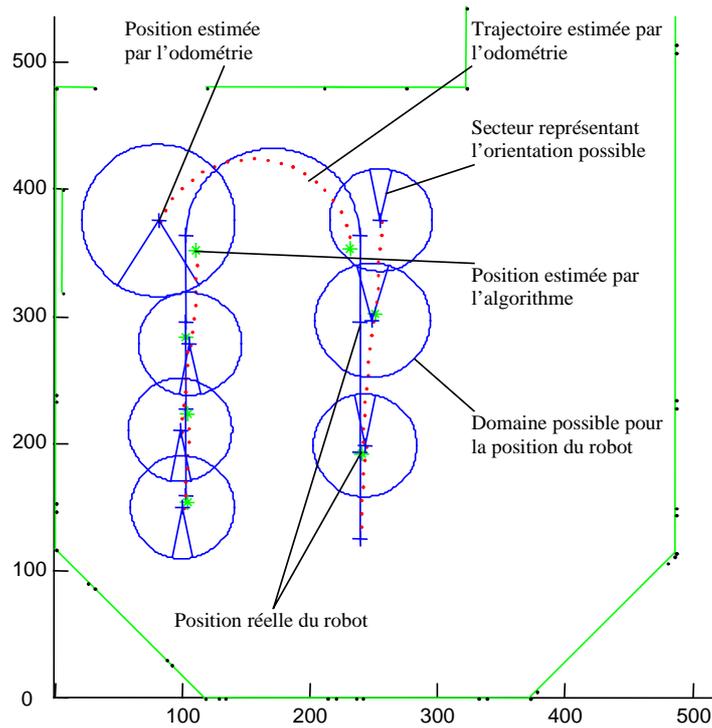
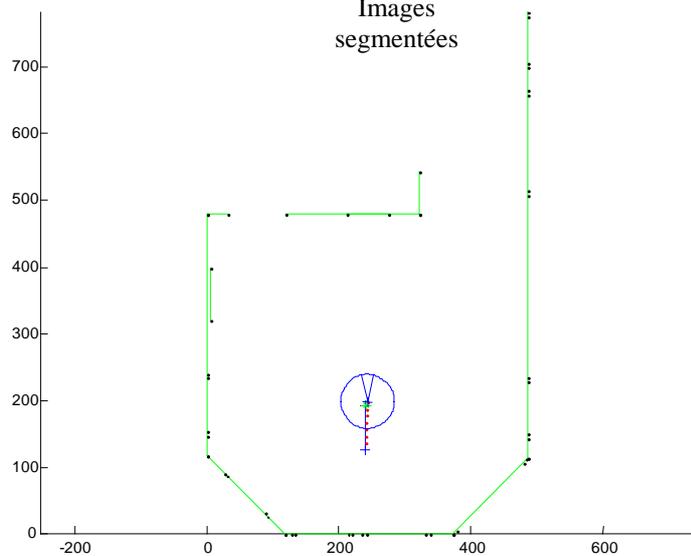
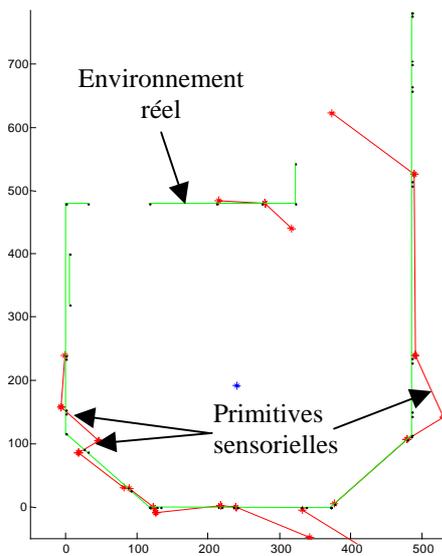
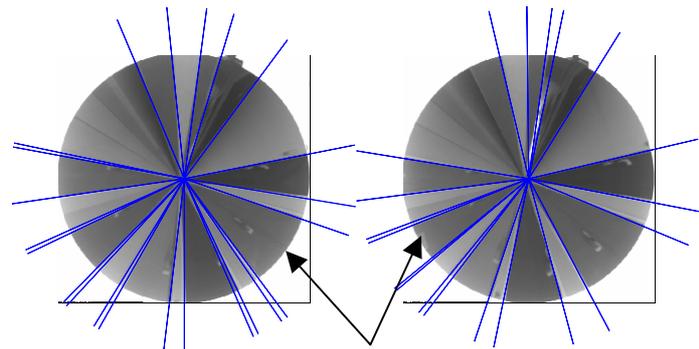


Figure 2-23 : Résultats de l'algorithme de localisation sur une trajectoire de 7 acquisitions.

1^{ère} Acquisition

Erreur en distance : 2,51 cm

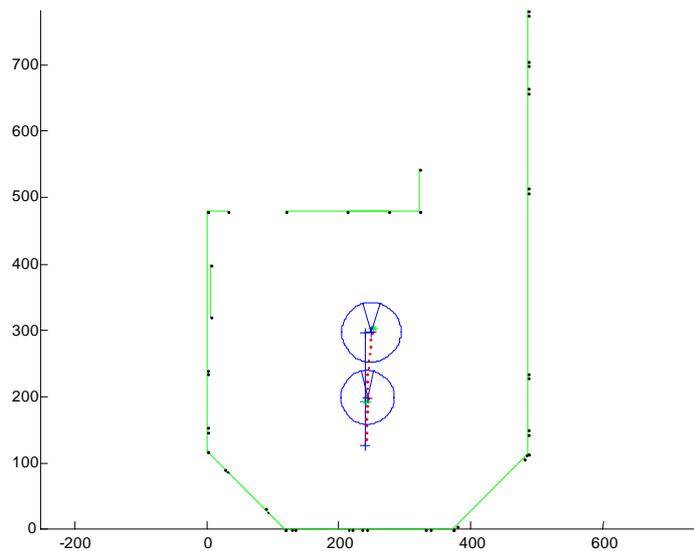
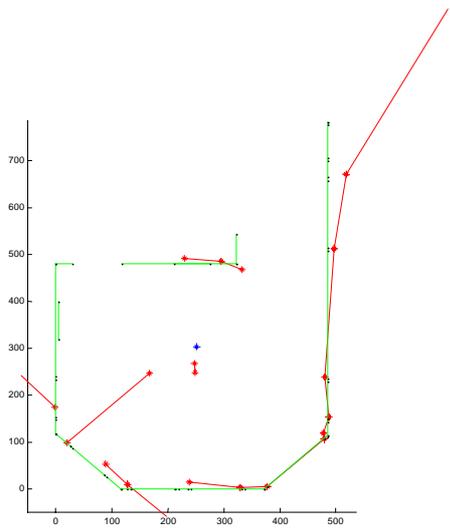
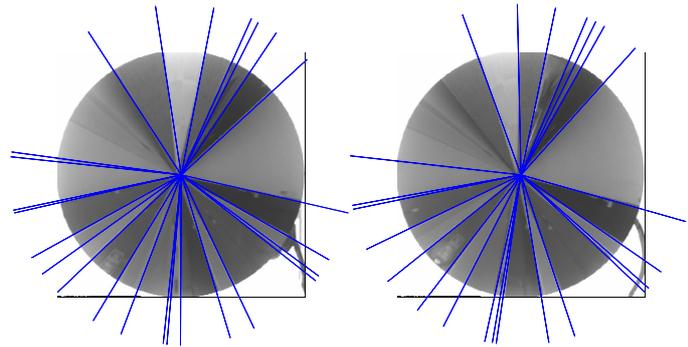
Erreur en angle : 0,38 °



2ème Acquisition

Erreur en distance : 16,81 cm

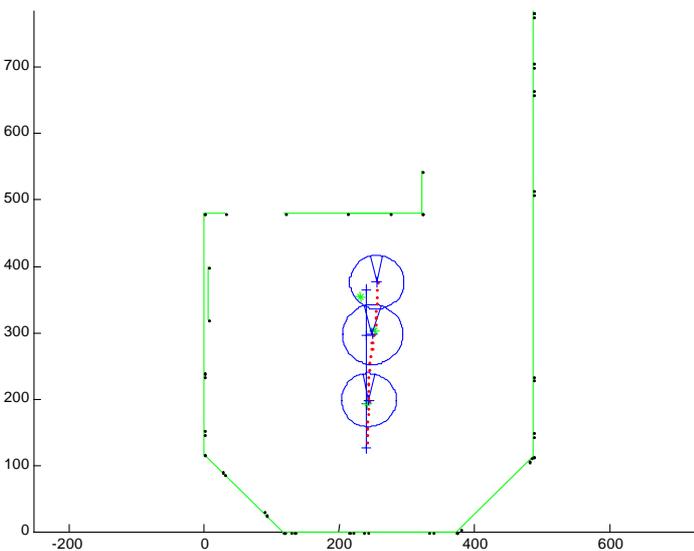
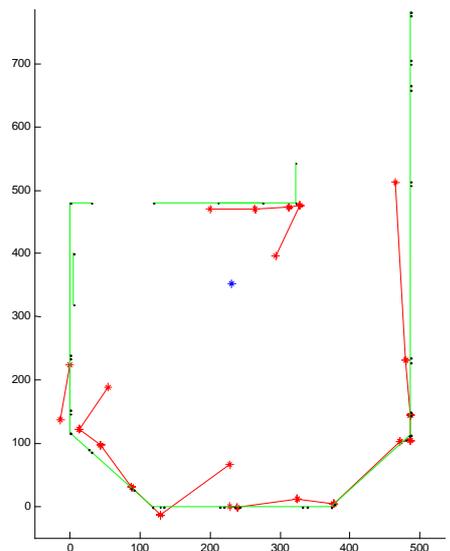
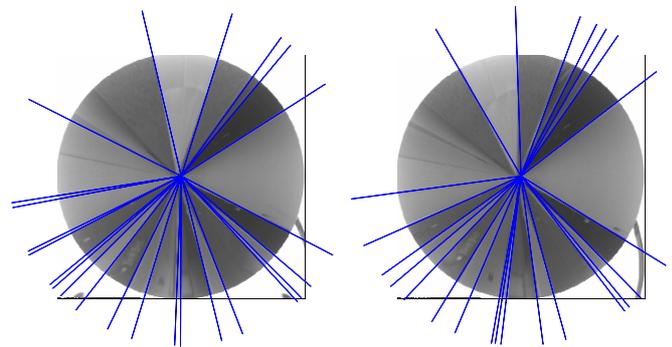
Erreur en angle : 1,39 °



3ème Acquisition

Erreur en distance : 10,82 cm

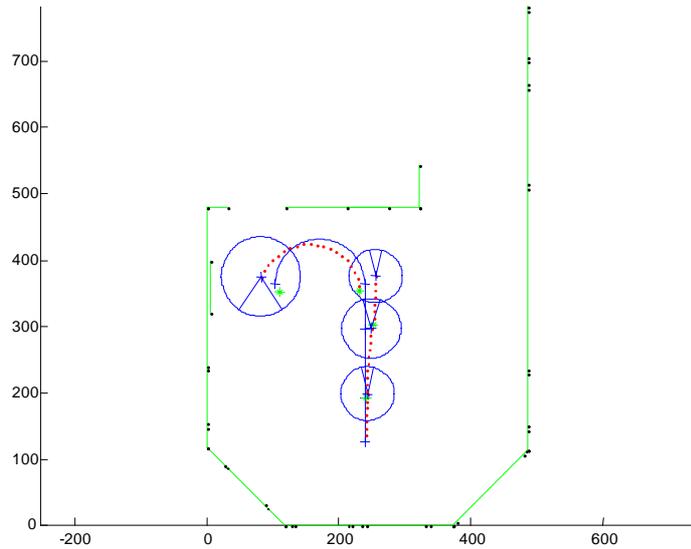
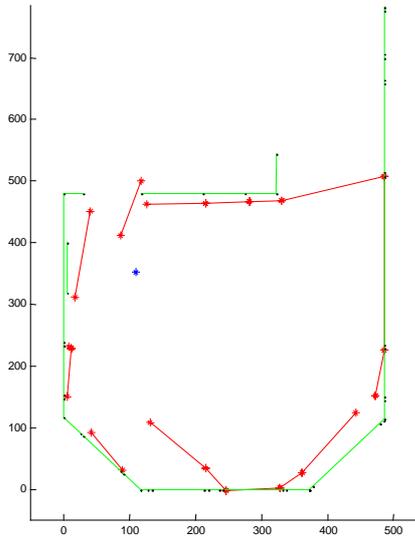
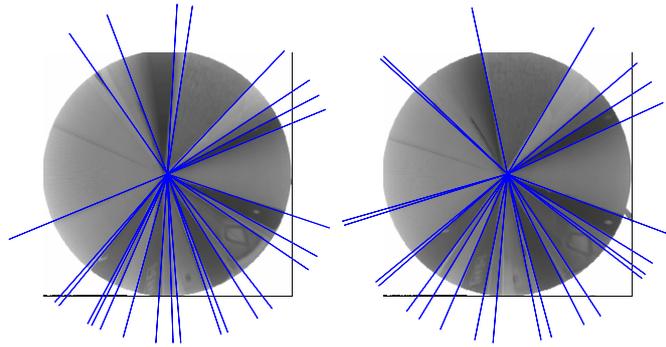
Erreur en angle : 0,44 °



4ème Acquisition

Erreur en distance : 14,27 cm

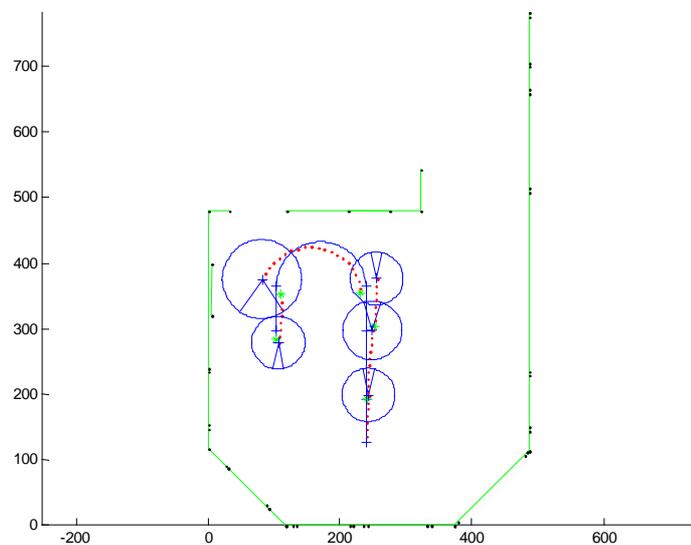
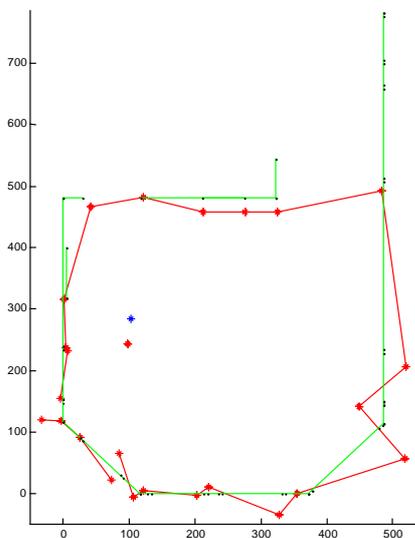
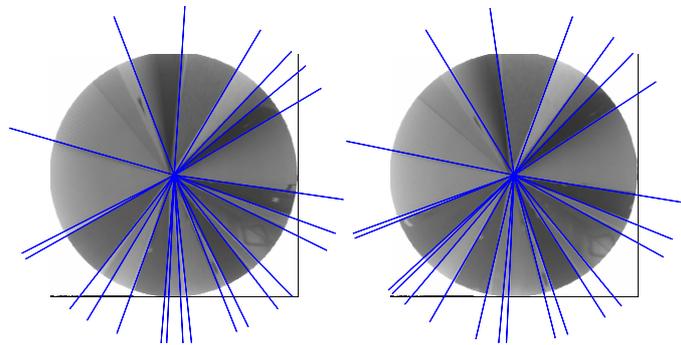
Erreur en angle : 0,21 °



5ème Acquisition

Erreur en distance : 11,59 cm

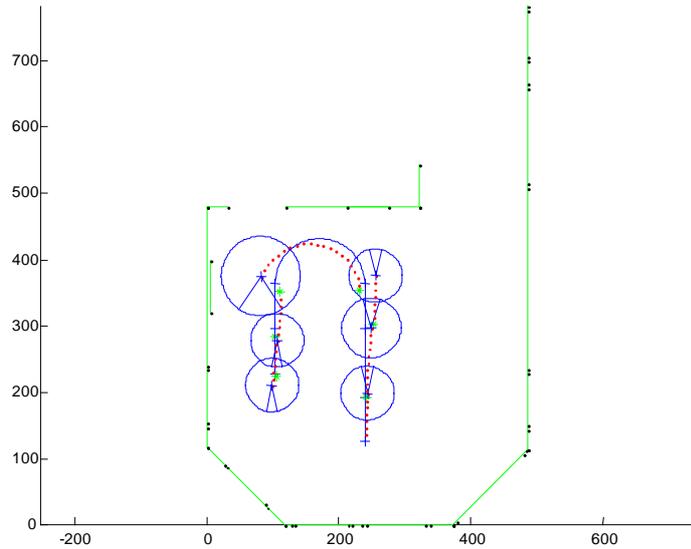
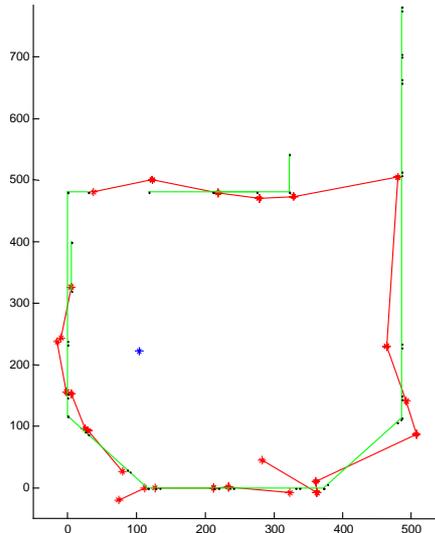
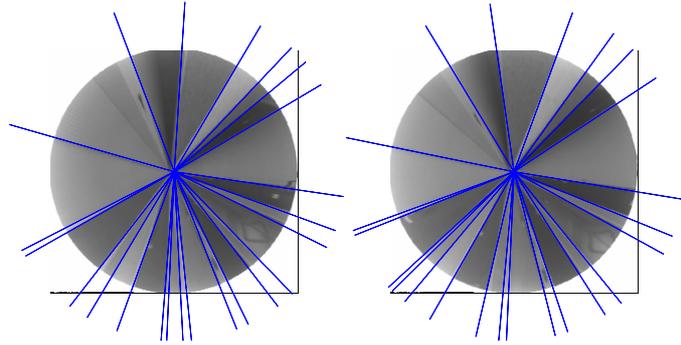
Erreur en angle : 2,03 °



6ème Acquisition

Erreur en distance : 4,16 cm

Erreur en angle : 1,25 °



7ème Acquisition

Erreur en distance : 5,53 cm

Erreur en angle : 1,42 °

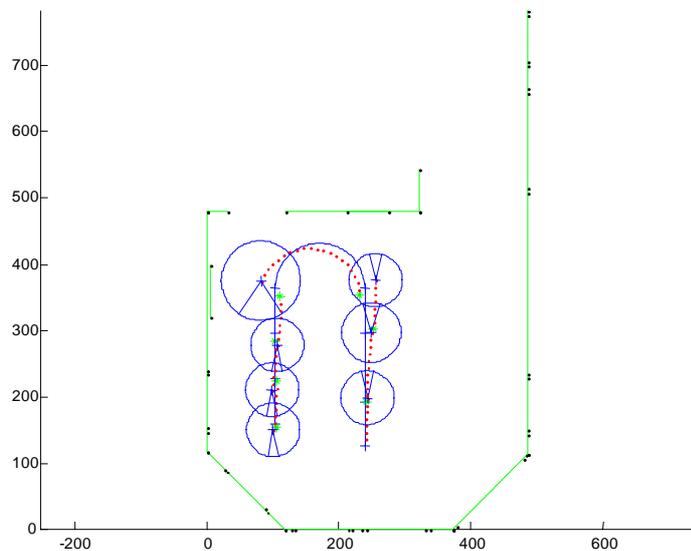
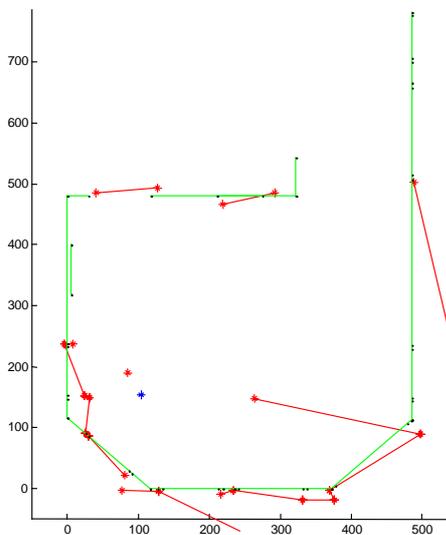
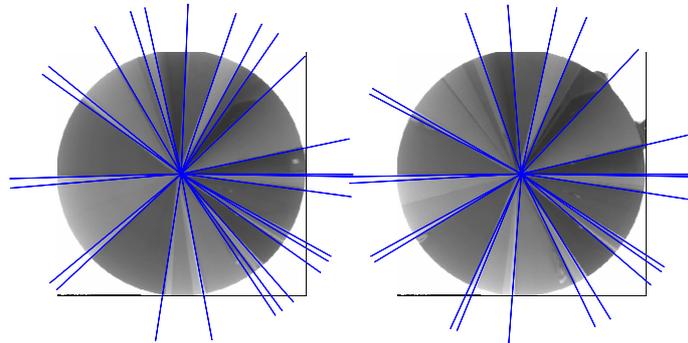


Figure 2-24 : Résultats obtenus avec notre méthode de localisation sur chaque acquisition.

Les Figure 2-23 et Figure 2-24 présentent les résultats de notre méthode de localisation montrant la mise en correspondance optimale du modèle bisensoriel avec la carte de l'environnement. Nous pouvons tout d'abord noter que la précision de notre méthode est relativement bonne. La robustesse de la méthode est ici mise en évidence par un taux d'échec quasi nul de l'étape de mise en correspondance. L'aspect fortement discriminant de l'erreur cartésienne cumulée peut être souligné.

En outre, on peut aussi noter que les variations de l'erreur montrées sur la Figure 2-24 sont fortement corrélées avec les erreurs introduites par la chaîne segmentation/appariement. Rappelons qu'une étape de calibration n'aurait pas d'influence prépondérante sur la diminution de l'erreur, ceci remis dans le contexte particulier d'utilisation des primitives radiales.

On pourra enfin souligner que la combinatoire de la phase de mise en correspondance n'a pas été diminuée au détriment de la robustesse. Il est important de replacer notre stratégie « stéréoscopique » qui nous permet de gérer des primitives de type point par rapport à une stratégie mettant directement en jeu des droites radiales [DEL97] [MAR99]. La mise en correspondance de droites radiales avec des points de l'environnement souffre essentiellement d'une grande complexité combinatoire, d'une tolérance très faible aux observations ne correspondant à aucun élément de la carte et enfin d'une convergence non garantie pour certaines configurations angulaires. Le fait de monter en niveau descriptif, c'est-à-dire de gérer des points et non plus des droites, permet de pallier aux problèmes précédents.

D'un point de vue système de perception, nous pouvons remarquer que sur toutes les acquisitions bisensorielles réalisées, un nombre important de primitives sensorielles se révèle exploitable dans la phase de mise en correspondance. Ce volume important de données sensorielles confère à notre système de localisation une grande robustesse quelque soit le type du milieu d'évolution.

2.8. Conclusion

Nous avons développé dans ce chapitre une méthode de localisation absolue basée sur la coopération de deux systèmes de perception omnidirectionnel stéréoscopique. Nous avons été amenés à résoudre le problème crucial de la construction d'un modèle sensoriel robuste à partir du système de vision conique. La robustesse est obtenue grâce à la fusion d'informations issues de traitements élémentaires complémentaires. La méthode de fusion multi-critères que nous avons choisi d'utiliser est la théorie des croyances de Dempster-Shafer. En premier lieu, elle nous a permis de prendre en compte la notion d'hétérogénéité de chaque traitement. D'autre part, associée à des critères pertinents, cette méthode a aussi permis de répondre efficacement au problème d'association de primitives dans nos deux images stéréoscopiques.

L'étude qualitative faite sur plusieurs acquisitions en milieu structuré (environnement intérieur) a permis de mettre en évidence le bien-fondé de notre approche puisque quelque soit le type d'environnement, nous obtenons une précision importante sur les primitives sensorielles considérées. Cette description sensorielle robuste nous a tout naturellement amenés à développer une méthode de localisation basée sur la mise en correspondance des points observés avec ceux répertoriés dans une carte de l'environnement. Deux stratégies ont alors été envisagées, la première basée sur un critère d'association qui est la distance de Hausdorff et la seconde basée sur le critère qu'est la distance cartésienne cumulée. De nombreux essais expérimentaux ont montré l'intérêt de la deuxième approche. La méthode de localisation ainsi développée est basée sur la prise en compte d'un critère de convergence simple et robuste puisqu'il s'agit de la distance qui sépare deux balises naturelles issues d'un appariement. Cette méthode de mise en correspondance permet de déterminer la configuration du robot avec une précision importante (10 cm en moyenne) et surtout avec un taux d'échec faible.

Enfin, la richesse du modèle sensoriel, obtenue en une acquisition, permet d'envisager l'extension de cette étude au problème de modélisation incrémentale de l'environnement du robot et ce dans le cas d'une absence totale de connaissance a priori. Cette approche consistera à placer le robot dans un environnement inconnu mais toujours polyédrique, et d'obtenir une carte de l'environnement mise à jour à chaque déplacement élémentaire. Il s'agit là de l'étude proposée dans le chapitre suivant. Le problème de la génération de carte incrémentale de l'environnement étant indissociable de celui de la localisation, l'approche proposée dans ce chapitre sera réutilisée dans le suivant.

Chapitre 3

Mise à jour de la carte de l'environnement au sens des moindres carrés

Sommaire partiel

3.1.	INTRODUCTION	68
3.2.	METHODE DE DECISION PAR FUSION D'INFORMATIONS	69
3.2.1	<i>Détermination de la croyance accordée à chaque association.....</i>	69
3.2.2	<i>Généralisation de l'opérateur de combinaison de Dempster-Shafer.....</i>	73
3.2.3	<i>Algorithme de décision.....</i>	74
3.3.	MISE A JOUR INCREMENTALE DE LA CARTE DE L'ENVIRONNEMENT	76
3.3.1	<i>Rappel sur les M.C.R.....</i>	77
3.3.2	<i>Mise à jour des coordonnées d'un amer</i>	78
3.3.3	<i>Phase d'initialisation de l'estimation de position d'une balise</i>	80
3.3.4	<i>Traitement des liens entre amers verticaux.....</i>	81
3.4.	TRAITEMENTS POST-MODELISATION	84
3.4.1	<i>Maintien de la cohérence par filtrage</i>	84
3.4.2	<i>Gestion des liens dans le cas d'une suppression d'un point de la carte</i>	85
3.5.	RESULTATS EXPERIMENTAUX	86
3.5.1	<i>Tests en environnement fermé et structuré sur de petits trajets</i>	86
3.5.2	<i>Résultats en environnement de type couloir sur des trajets importants.....</i>	89
3.6.	CONCLUSION	94

3.1. Introduction

Dans le chapitre précédent, nous avons obtenu des données sensorielles issues de deux images omnidirectionnelles. Ces primitives sont d'un type particulier puisqu'elles représentent uniquement les amers verticaux de l'environnement. C'est d'ailleurs cette particularité qui nous a guidé vers le choix d'algorithmes de mise en correspondance basés sur des points pour la localisation, et non sur des primitives plus élaborées tel que les segments.

La robustesse des résultats obtenus lors de la phase de localisation nous permettent maintenant d'aborder l'étape suivante qui consiste à construire une carte de l'environnement au fur et à mesure des acquisitions, et cela sans connaissance *a priori*. Pour réaliser cette étape nous devons résoudre deux problèmes principaux :

- Décider quelles sont les informations sensorielles à fusionner avec les éléments de la carte.
- Mettre à jour la carte de manière incrémentale, en fusionnant certaines données et en initialisant celles correspondant à de nouveaux amers.

Ce chapitre va donc s'articuler autour de trois parties. La première sera consacrée à l'aspect décision, c'est à dire comment connaître les informations sensorielles qui font déjà partie de la carte et celles que nous devons ajouter. Dans un deuxième temps, nous présenterons la méthode que nous avons utilisée pour mettre à jour les informations de la carte et qui est basée sur les moindres carrés. Enfin, nous exposerons différents résultats expérimentaux de construction incrémentale d'une carte dans deux environnements de types différents.

3.2. Méthode de décision par fusion d'informations

La construction de la carte de l'environnement est basée sur un processus incrémental d'acquisitions et de mises à jour. Toutefois, pour pouvoir utiliser ces données, il faut tout d'abord savoir quelles sont les informations à fusionner et les nouvelles qu'il faut ajouter à la carte.

Nous exposerons la méthode de décision retenue, consistant à calculer pour chaque primitive sensorielle de type amer, la croyance accordée à son appariement avec un point de la carte. Précisons à ce niveau que nous appelons carte la modélisation de l'environnement obtenue à l'aide des $n-1$ acquisitions précédentes à celle considérée. Nous utiliserons pour ceci les règles de combinaison de Dempster-Shafer [SHA76] et la formulation introduite et développée par Gruyer [GRU99a]. Nous adopterons dans ce chapitre les notations suivantes :

- S : Ensemble des données sensorielles,
- S_i : Primitive sensorielle i ,
- s : Nombre de primitives sensorielles,
- P : Ensemble des points de la carte,
- P_j : Point j de la carte,
- p : nombre de points dans la carte.

3.2.1 Détermination de la croyance accordée à chaque association

Avant de décider quels sont les points à fusionner, nous devons trouver des critères discriminants permettant de calculer une croyance correspondant à l'affirmation "Le point observé S_i doit être fusionné avec le point P_j de la carte".

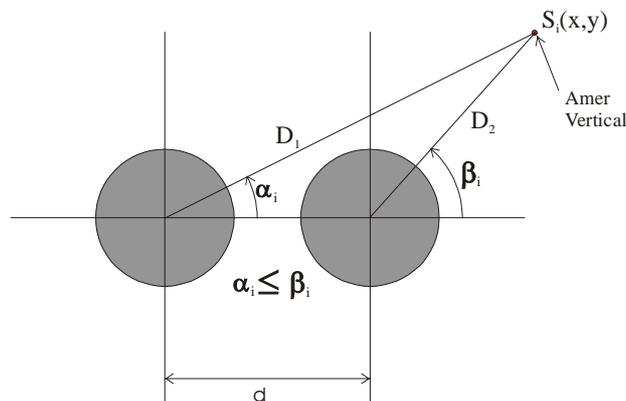


Figure 3-1 : Calcul des coordonnées d'un point.

Le calcul des coordonnées du point observé S_i est issu des deux angles d'observation α_i et β_i donnés par le capteur (Figure 3-1).

Nous avons constaté qu'il était plus intéressant de conserver ces deux informations (à savoir les deux angles α_i et β_i) plutôt que de gérer directement les coordonnées du point. Ceci est justifié par le fait qu'une erreur sur un des deux angles nous donne obligatoirement des coordonnées fausses, mais n'influe en rien sur le deuxième angle (Figure 3-2). Ce deuxième angle est donc dans ce cas une information exploitable au sens du critère d'appariement. En

revanche, un point calculé à partir d'un angle faux ne pourra pas être exploitable au sens de la mise en correspondance.

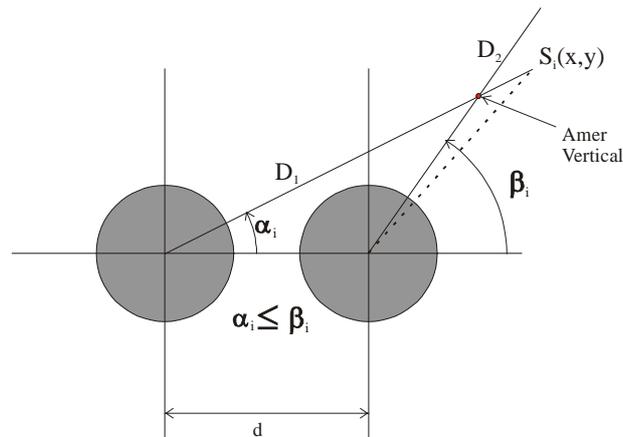


Figure 3-2 : Erreur pouvant entacher un angle.

Pour les raisons précédemment exposées, nous avons décidé d'utiliser l'information d'angle pour chacun des deux capteurs omnidirectionnels comme critère unique et donc prépondérant de mise en correspondance d'une primitive sensorielle avec un élément de la carte de l'environnement.

Comme nous le montre la Figure 3-3, nous comparons deux à deux la différence d'angle entre un point de la carte et un point observé, ceci pour les deux images stéréoscopiques. Nous obtenons donc deux grandeurs, $\gamma = \text{abs}(\alpha_i - \varphi_j)$ et $\mu = \text{abs}(\beta_j - \sigma_j)$ qui caractérisent la similitude entre l'observation et le point de la carte.

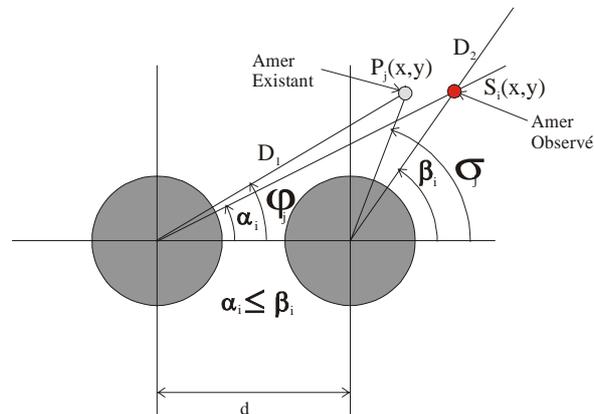


Figure 3-3 : Critère d'appariement : différence d'angle entre l'amer observé et l'amer existant.

La fusion de ces deux informations est réalisée avec les règles de combinaison de Dempster-Shafer, dont les fondements ont été détaillés dans le précédent chapitre [SHA76] [DEM67].

Notre cadre de discernement est constitué de deux éléments :

$$\Theta = \{\text{OUI}, \text{NON}\}$$

Qui correspondent aux assertions :

- "OUI, il faut fusionner le point observé i avec le point j de la carte",
- "NON, il ne faut pas fusionner le point observé i avec le point j de la carte".

Nous pouvons calculer à partir de ces valeurs nos fonctions de masses données par la Figure 3-4. Tout comme dans le chapitre précédent, les fonctions de masse utilisées à ce niveau intègrent l'ignorance.

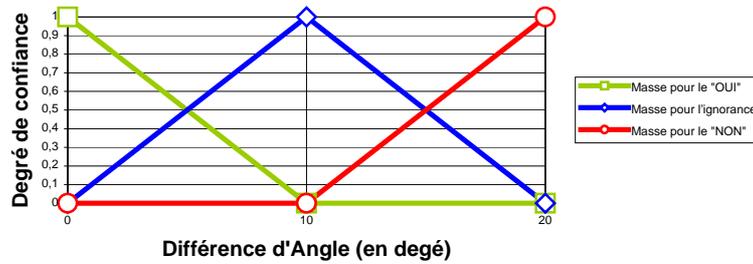


Figure 3-4 : Fonction de masse pour les différences d'angles.

Les seuils liés aux angles qui apparaissent au niveau de la fonction de masse ont été déterminés expérimentalement.

Nous obtenons :

$$m^c(OUI) = \begin{cases} 1 - \frac{x}{10} & \text{Si } x < 10 \\ 0 & \text{Sinon} \end{cases} \quad [3-1]$$

$$m^c(NON) = \begin{cases} \frac{x}{10} - 1 & \text{Si } x > 10 \\ 0 & \text{Sinon} \end{cases} \quad [3-2]$$

$$m^c(\Theta) = \begin{cases} 1 - m^c(OUI) & \text{Si } x < 10 \\ 1 - m^c(NON) & \text{Sinon} \end{cases} \quad [3-3]$$

Avec "x" représentant la différence d'angle et "c" représentant le capteur 1 ou 2.

La fusion de ces deux fonctions de masse est donnée par les équations suivantes :

$$K = m^1(OUI) \times m^2(NON) + m^1(NON) \times m^2(OUI) \quad [3-4]$$

$$m(OUI) = \frac{m^1(OUI) \times m^2(OUI) + m^1(OUI) \times m^2(\Theta) + m^1(\Theta) \times m^2(OUI)}{1 - K} \quad [3-5]$$

$$m(NON) = \frac{m^1(NON) \times m^2(NON) + m^1(NON) \times m^2(\Theta) + m^1(\Theta) \times m^2(NON)}{1 - K} \quad [3-6]$$

$$m(\Theta) = \frac{m^1(\Theta) \times m^2(\Theta)}{1 - K} \quad [3-7]$$

Dans le but de mieux comprendre cette fusion, il nous a semblé intéressant de représenter graphiquement les zones correspondant à ces trois jeux de masses (Figure 3-5). Pour ce faire, nous avons calculé chacun de ces trois jeux de masses pour toutes les valeurs d'angles pouvant être prises, par rapport à une variation autour du point observé. Cette variation est appliquée sur les deux angles.

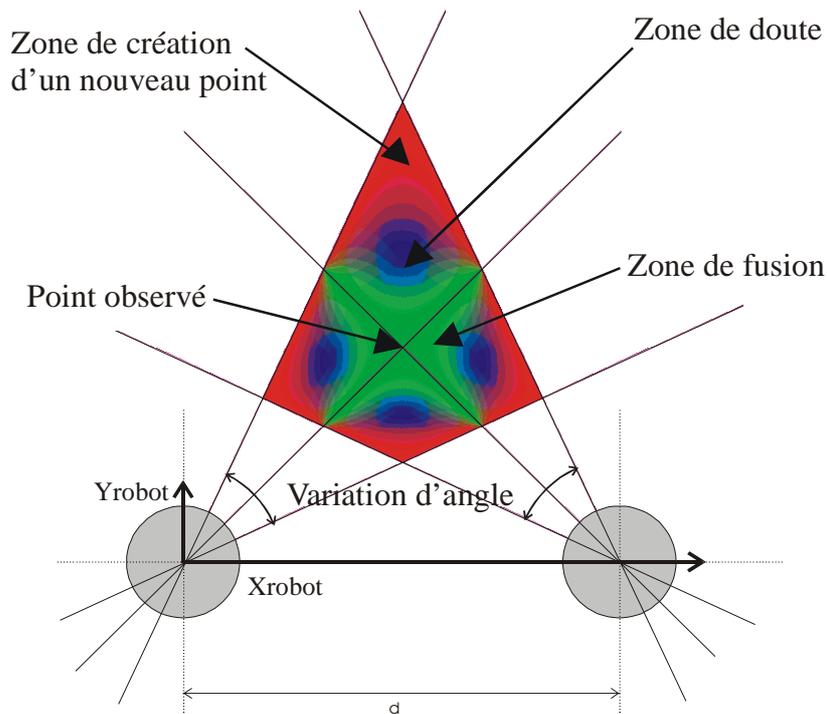


Figure 3-5 : Représentation des différentes zones pour la fusion des points.

Nous pouvons constater graphiquement la pertinence du critère choisi puisqu'il apparaît bien que le degré de confiance maximal est obtenu le long des droites matérialisant le point observé. La zone de croyance maximum intègre donc bien la possibilité d'avoir un angle observé "bon" et l'autre erroné. Ce cas de figure n'est pas marginal puisqu'il est souvent observé expérimentalement.

Finalement, l'appariement avec un point de la carte passe par les phases suivantes : Pour un point i observé parmi s , il faut calculer ces trois valeurs pour les p points de la carte. Nous utiliserons dans la suite de ce chapitre les notations suivantes :

- $m_{i,j} (P_j)$: masse associée à la proposition "Le point observé S_i doit être fusionné avec le point P_j de la carte",
- $m_{i,j} (\overline{P_j})$: masse associée à la proposition "Le point observé S_i ne doit pas être fusionné avec le point P_j de la carte",
- $m_{i,j} (\Theta)$: masse représentant l'ignorance,
- $m_{i..} (*)$: masse représentant le rejet, "Le point observé S_i ne doit pas être fusionné avec un des points P_j de la carte",

Remarque :

Le premier indice représente le point observé alors que le deuxième représente le point de la carte. Le point noté "." à la place de l'indice représente l'ensemble des points observés ou existants.

Nous avons alors, pour un point observé S_i , les p triplets :

$$\begin{array}{lll} m_{i,1}(P_1) & m_{i,1}(\overline{P_1}) & m_{i,1}(\Theta_1) \\ m_{i,2}(P_2) & m_{i,2}(\overline{P_2}) & m_{i,2}(\Theta_2) \\ \dots & \dots & \dots \\ m_{i,p}(P_p) & m_{i,p}(\overline{P_p}) & m_{i,p}(\Theta_p) \end{array}$$

Nous pouvons alors calculer ces p triplets pour les s points observés, ce qui nous donne $s \times p$ triplets. La problématique introduite à ce niveau réside dans la fusion de l'ensemble de ces informations, afin de pouvoir effectuer des choix. Nous avons résolu ce problème en utilisant la généralisation de l'opérateur de combinaison de Dempster-Shafer.

3.2.2 Généralisation de l'opérateur de combinaison de Dempster-Shafer.

Dans le chapitre précédent, nous avons présenté les règles de combinaison de Dempster-Shafer (appariement des secteurs de niveau de gris). Nous allons ici détailler l'utilisation de l'opérateur généralisé introduit par D. Gruyer et V. Cherfaoui [GRU99a] [GRU99b] dont le premier formalisme a été donné par M. Rombaut [ROM97] [ROM98]. Cette généralisation permet notamment d'éviter l'explosion combinatoire due à la combinaison d'un grand nombre de sources.

Notre problème est le suivant : un point S_i observé lors d'une acquisition est soit un point P_j existant dans la carte, soit un nouveau point. Par conséquent, notre cadre de discernement est :

$$\Theta = \{ P_1, P_2, \dots, P_s, * \}$$

où P_j caractérise l'assertion "Le point observé S_i est le point P_j de la carte"

Afin d'avoir un cadre de discernement exhaustif, nous ajoutons l'hypothèse "*" qui signifie que le point S_i ne correspond à aucun point de la carte. Nous travaillons donc en monde ouvert étendu [ROY01]. Par contre, nous pouvons limiter le référentiel de définition en posant comme contrainte qu'un point observé ne peut être mis en relation qu'avec un et un seul point de la carte.

Dans ces conditions, D. Gruyer [GRU99] a démontré par récurrence que l'on obtient le jeu d'équations suivant :

$$m_{i,\cdot}(P_j) = K_{i,\cdot} \times m_{i,j}(P_j) \times \prod_{\substack{k=1,p \\ k \neq j}} (1 - m_k(P_k)) \quad [3-8]$$

$$m_{i..} (*) = K_{i..} \times \prod_{j=1,p} m_{i,j}(\overline{P_j}) \quad [3-9]$$

$$m_{i..} (\Theta) = K_{i..} \times \left(\prod_{j=1,p} (m_{i,j}(\Theta) + m_{i,j}(\overline{P_j})) - \prod_{j=1,p} m_{i,j}(\overline{P_j}) \right) \quad [3-10]$$

$$\text{avec } K = \frac{1}{\prod_{j=1,p} (1 - m_{i,j}(P_j)) \times \left(1 + \sum_{j=1}^p \frac{m_{i,j}(P_j)}{1 - m_{i,j}(P_j)} \right)} \quad [3-11]$$

Finalement, nous pouvons calculer ces $p+2$ valeurs pour les n points observés. Nous obtenons la matrice de croyance suivante :

	P_1	P_2	...	P_j	...	P_p	*	Θ
S_1	$m_{1..}(P_1)$	$m_{1..}(P_2)$...	$m_{1..}(P_j)$...	$m_{1..}(P_p)$	$m_{1..} (*)$	$m_{1..}(\Theta)$
S_2	$m_{2..}(P_1)$	$m_{2..}(P_2)$...	$m_{2..}(P_j)$...	$m_{2..}(P_p)$	$m_{2..} (*)$	$m_{2..}(\Theta)$
...
S_i	$m_{i..}(P_1)$	$m_{i..}(P_2)$...	$m_{i..}(P_j)$...	$m_{i..}(P_p)$	$m_{i..} (*)$	$m_{i..}(\Theta)$
...
S_s	$m_{n..}(P_1)$	$m_{n..}(P_2)$...	$m_{n..}(P_j)$...	$m_{n..}(P_p)$	$m_{n..} (*)$	$m_{n..}(\Theta)$

Par rapport à cette matrice, il nous reste maintenant à choisir l'ensemble des associations entre les points observés et les points de la carte. Nous discuterons d'un algorithme de décision dans la partie suivante.

3.2.3 Algorithme de décision

Avec la théorie de l'évidence, il existe plusieurs lois de décisions dont les plus connues sont :

- **Le maximum de crédibilité** : Elle consiste à garder la solution la plus crédible, la crédibilité représentant le degré de croyance minimal. Ce critère de décision est dit *pessimiste*.
- **Le maximum de plausibilité** : Elle consiste à garder la solution la plus plausible, la plausibilité représentant le degré de croyance maximal. Ce critère de décision est dit *optimiste*.
- **Le maximum de probabilité pignistique** [SME94] : Elle consiste à équi-répartir la masse placée sur des propositions, sur les disjonctions d'objets qui la compose. Ce critère de décision est dit *prudent*.

Le résultat de notre matrice de croyance fournit une distribution de masse sur les hypothèses singletons, donc une loi de décision basée sur le maximum de probabilité pignistique n'apporterait rien ici. De plus, les valeurs de cette matrice sont directement des mesures crédibilistes. C'est pourquoi nous avons basé notre critère de décision sur le maximum de crédibilité de cette matrice.

Voici notre algorithme :

```

Tant que le maximum de la matrice > 0
  mi,j = maximum de la matrice.
  Si j ≤ (nombre de points de la carte) alors
    Le point observé i doit être fusionné avec le point j de la carte
    Passer tous les éléments de la colonne j à 0
  sinon
    Le point observé i est un nouveau point
  fin si
  Passer tous les éléments de la ligne i à 0
fin tant que
    
```

Cet algorithme est basé sur la recherche de la valeur maximum dans la matrice précédemment construite. La valeur ainsi trouvée permet de déterminer si le point observé est en relation avec un point existant ou si un nouveau point est créé. En cas de doute (crédibilité maximale sur " Θ "), nous avons choisi de créer un nouveau point. Ce choix est lié à un post-traitement que nous avons intégré dans le cadre du maintien de la cohérence de la carte : nous verrons dans l'étape suivante qu'à la fin de la phase de mise à jour, les points qui sont proches (au niveau de leur distance cartésienne) sont automatiquement fusionnés.

Une fois cette association effectuée, tous les éléments de la ligne contenant la valeur maximale sont mis à 0, ainsi que ceux de la colonne mais uniquement si cette dernière est différente de "*" et de " Θ ". En effet, l'initialisation de l'ensemble des éléments de la ligne à 0 signifie qu'un élément observé ne peut être en relation qu'avec un seul élément de la carte, de même pour la colonne qui correspond au fait que plusieurs observations ne peuvent être associées au même point de la carte. Par contre, plusieurs observations peuvent être de nouveaux points ("*"), de même que l'ignorance peut être maximum sur plusieurs observations (" Θ "). L'algorithme est réitéré tant qu'il reste des valeurs positives.

Nous présentons un exemple de fonctionnement de cet algorithme sur la Figure 3-6 où les variables S représentent les observations et les variables P sont les éléments de la carte :

- Etape 1 : Matrice de départ,
- Etape 2 : La recherche du maximum de la matrice donne S_i en relation avec P_j .
- Etape 3 : Puisque P_j ne représente ni "*", ni " Θ ", l'ensemble des valeurs de la colonne P_j passe à 0, ce qui signifie qu'aucune des observations restantes ne pourra être de nouveau en relation avec P_j .
- Etape 4 : Finalement, l'ensemble des valeurs de la ligne contenant S_i passe aussi à 0, afin que l'observation S_i ne soit plus associée à aucun autre élément de la carte.

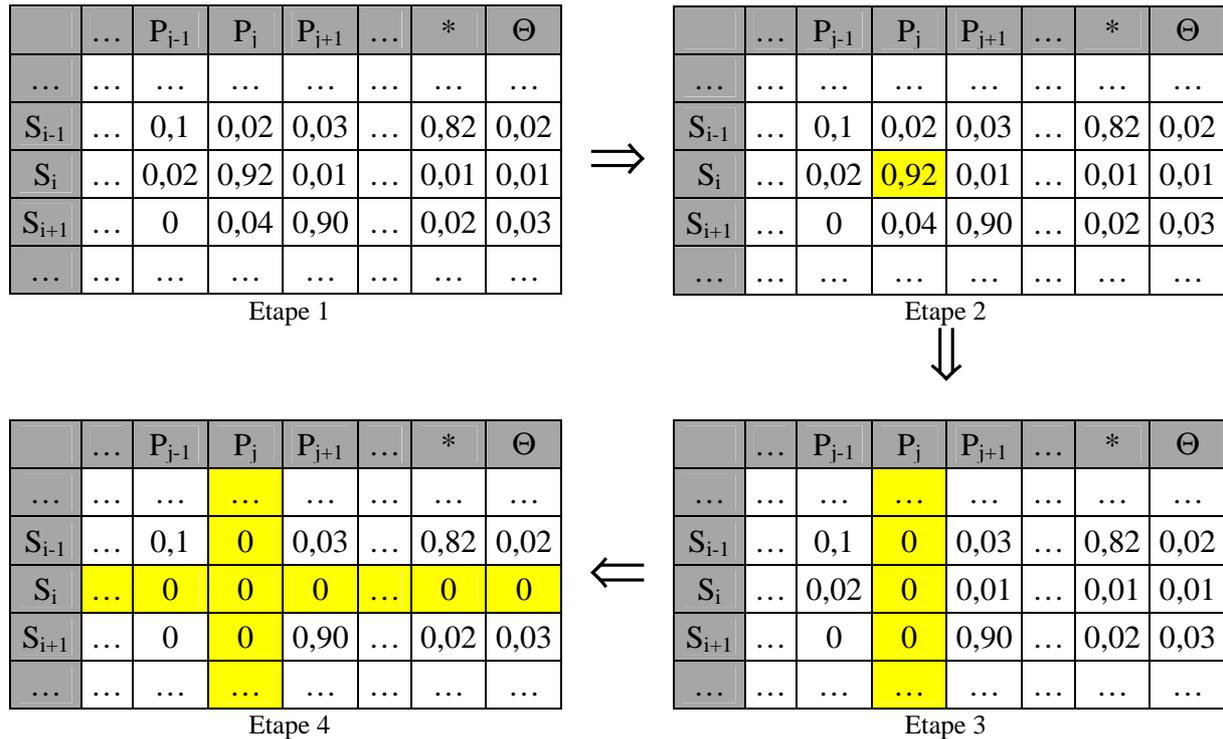


Figure 3-6 : Exemple d'itérations de l'algorithme de décision.

Une fois cette étape terminée, nous obtenons une table de mise en correspondance contenant pour chaque observation le point qui est associé, ou un champ vide si c'est un nouveau point. Cette table sera transmise à la phase suivante qui consiste à mettre à jour la carte.

3.3. Mise à jour incrémentale de la carte de l'environnement

La phase de mise à jour incrémentale d'une carte de l'environnement est générique quelles que soient les primitives sensorielles traitées. Elle se compose de deux cas distincts :

- Fusionner une observation avec une primitive de la carte,
- Intégrer et initialiser une observation en tant que nouvelle primitive de la carte.

Comme nous l'avons précédemment mentionné, nous connaissons grâce à la table de mise en correspondance les points qui sont à fusionner et les nouveaux qui sont à intégrer. Il nous faut trouver alors une méthode de mise à jour, au sens de l'estimation de paramètres, qui devra être gérée de manière itérative.

Plusieurs solutions sont possibles à ce niveau : nous pouvons considérer les points comme des primitives "directement fusionnables" ou bien, comme dans le cadre de décision, se servir des deux angles utilisés dans le calcul du point.

Nous avons retenu cette dernière solution basée sur les angles pour plusieurs raisons :

- Tout d'abord, elle s'inscrit dans la continuité de l'étape de décision, où nous avons considéré les deux angles comme deux sources distinctes d'informations,

- Ensuite, le fait qu'un point soit caractérisé par deux angles nous permet, comme dans la phase précédente, de tolérer qu'une des deux valeurs soit erronée (Figure 3-2), sans entacher obligatoirement le processus de mise à jour.

A ce stade, comme nous l'avons précisé dans l'état de l'art, l'estimation des paramètres des primitives géométriques de la carte est généralement réalisée à l'aide d'outils tels que le filtrage de Kalman. Notre choix s'est porté ici sur un formalisme plus simple basé sur la minimisation de critère quadratique : il s'agit des Moindres Carrés Récursifs (M.C.R.).

Nous allons dans un premier temps faire un bref rappel sur l'utilisation de la méthode des moindres carrés récursifs, puis nous expliquerons comment nous l'avons utilisé lors de la mise à jour de la carte.

3.3.1 Rappel sur les M.C.R.

Nous allons chercher à estimer, au sens des moindres carrés, le vecteur position $[x_b \ y_b]^T$ caractérisant géométriquement un amer vertical. Ce problème d'estimation peut être résolu de différentes manières. Nous avons choisi de travailler avec la méthode des moindres carrés et plus précisément avec la forme récursive des moindres carrés puisque les observations ne sont pas toutes disponibles à un instant k . Cette méthode est bien adaptée au problème que nous avons à traiter : elle respecte la contrainte incrémentale du processus de mise à jour.

De manière générale, dans le cas multivariable, la méthode des moindres carrés récursifs s'applique lorsque l'équation de mesure est donnée par :

$$y_i = a_1 x_{1i} + a_2 x_{2i} + \dots + a_n x_{ni} + b_i \quad [3-12]$$

où (x_1, \dots, x_n) représentent n variables connues, (a_1, \dots, a_n) les paramètres inconnus à estimer et y l'observation entachée d'un bruit b .

Soit sous la forme vectorielle :

$$y_i = \underline{x}_i^T \underline{a} + b_i \quad [3-13]$$

Le problème est alors de trouver, à partir de k observations y_i , le meilleur estimé du vecteur \underline{a} , au sens d'un critère d'écart quadratique C :

$$C = \sum_{i=1}^k (y_i - \underline{x}_i^T \hat{\underline{a}}_k)^2 \quad [3-14]$$

Les équations de récurrence des moindres carrés (décrites dans [LAB88]) pour le cas multivariable, sont données par les formules :

$$\begin{aligned} \hat{\underline{a}}_k &= \hat{\underline{a}}_{k-1} + K_k (y_k - \underline{x}_k^T \hat{\underline{a}}_{k-1}) \\ P_k &= P_{k-1} - K_k \underline{x}_k^T P_{k-1} \\ K_k &= P_{k-1} \underline{x}_k (1 + \underline{x}_k^T P_{k-1} \underline{x}_k)^{-1} \end{aligned} \quad [3-15]$$

Nous pouvons remarquer que cette formulation récursive est très proche de celle fournie par le filtrage de Kalman, puisque l'estimée à l'instant k est égale à l'estimée à l'instant $k-1$ plus une correction, qui, dans ce cas, dépend de la distance de la droite estimée à la nouvelle observation y_k . Nous retrouvons ainsi, tout comme dans le filtrage de Kalman, le terme d'innovation, ici $(y_k - \underline{x}_k^T \hat{a}_{k-1})$, qui représente l'apport d'information de l'observation y_k .

P_k est le poids attaché à la valeur instantanée du gradient de C à l'instant k . Il dépend des estimations précédentes et décroît constamment avec k . Finalement, plus nous effectuons d'estimations, meilleures sont ces estimations, et moins nous attachons d'importance aux écarts entre les paramètres estimés et l'observation.

Les équations récurrentes des moindres carrés récursifs sont intéressantes d'un point de vue temps de traitement : elles ne font pas intervenir d'inversion de matrices, puisque l'expression $(1 + \underline{x}_k^T P_{k-1} \underline{x}_k)$ est un scalaire.

La méthode des moindres carrés récursifs sera donc utilisée pour l'estimation des coordonnées (x,y) liées à chaque amer observé. Deux points restent à traiter :

- Calculer la position d'un amer observé dans le repère du monde à partir des données sensorielles disponibles,
- Déterminer les valeurs initiales a_0 et P_0 pour un nouvel amer.

3.3.2 Mise à jour des coordonnées d'un amer

Les données qui vont permettre d'estimer les positions des balises à une acquisition k sont d'une part, les positions du robot jusqu'à cette acquisition $(x r_k, y r_k, \theta r_k)$ et d'autre part, les angles des amers b_i dans le repère du robot.

A partir du moment où deux observations concernant le même amer sont disponibles, ses coordonnées $(x b^i, y b^i)$ dans le repère de la carte de l'environnement peuvent être directement obtenues avec l'équation suivante (Figure 3-7) :

$$\tan (\theta r_k + \phi_k^i) = \frac{y r_k - y b^i}{x r_k - x b^i} \quad [3-16]$$

Ce système d'équations sera surdéterminé. L'estimation de $(x b^i, y b^i)$ sera réalisée de façon incrémentale tout au long du déplacement du robot.

Il nous faut maintenant écrire l'équation sous une forme qui va nous permettre d'appliquer les équations de récurrence que nous avons développées précédemment. L'équation s'écrit, pour une série de données obtenues à une acquisition k :

$$(x r_k - x b^i) \tan(\alpha^i) = (y r_k - y b^i) \quad [3-17]$$

avec $\alpha^i = \theta r_k + \phi^i$ l'angle que fait l'amer b_i par rapport au robot dans le repère de l'environnement, ce qui nous donne finalement :

$$xb_i \tan(\alpha^i) - yb^i = xr_k \tan(\alpha^i) - yr_k \quad [3-18]$$

Soit sous forme matricielle :

$$\begin{bmatrix} \tan(\alpha^i) & -1 \end{bmatrix} \begin{bmatrix} xb_i \\ yb_i \end{bmatrix} = xr_k \tan(\alpha^i) - yr_k \quad [3-19]$$

d'où l'expression :

$$\underline{x}_i^T \underline{a}_i = y_i \quad [3-20]$$

avec : $\underline{x}_i^T = \begin{bmatrix} \tan(\alpha^i) & -1 \end{bmatrix}$

$$\underline{a}_i = \begin{bmatrix} xb_i \\ yb_i \end{bmatrix}$$

et $y_i = xr \tan(\alpha^i) - yr$

Comme nous avons deux observations pour une même position du robot, nous appliquerons deux fois cette équation par acquisition (Figure 3-7).

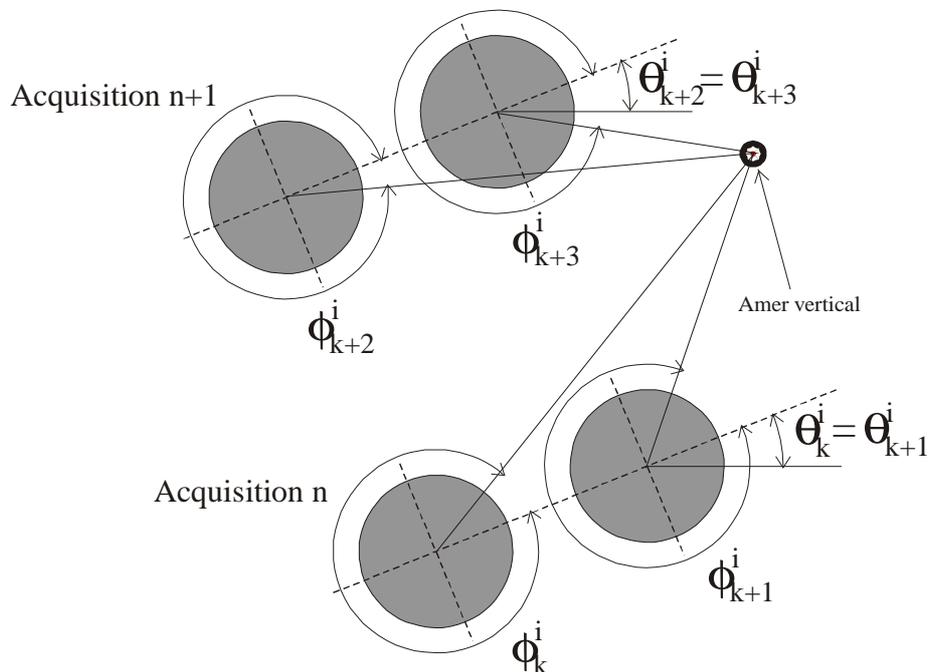


Figure 3-7 : Estimation de la position d'un amer à partir de plusieurs observations stéréoscopiques.

3.3.3 Phase d'initialisation de l'estimation de position d'une balise

L'algorithme des moindres carrés récursifs suppose de fixer des valeurs initiales pour les paramètres a_0 et P_0 . Le processus itératif peut, en fait, être initialisé directement à l'observation k_0 . Si nous choisissons une valeur de P_{k_0} suffisamment grande, à partir d'un certain nombre d'observations, l'estimation est indépendante de la condition initiale a_{k_0} . Cette condition initiale peut être choisie arbitrairement, elle est généralement égale à zéro.

Plutôt que de prendre des conditions initiales arbitraires, nous avons choisi de déterminer directement un a_0 qui sera le plus proche possible de l'estimation finale. Pour cela, nous initialisons les valeurs de a_0 et de P_0 à partir du moment où deux observations concernant une même balise sont disponibles. Or comme nous avons intrinsèquement deux observations pour une position donnée du robot, nous pouvons nous servir directement de la première acquisition pour initialiser ces paramètres.

Nous avons donc deux observations angulaires (ϕ_{k-1}^i, ϕ_k^i) associées aux deux estimations de position du robot (plus exactement les positions du capteur au moment de l'acquisition) $(x_{r_{k-1}}, y_{r_{k-1}}, \theta_{r_{k-1}})$ et $(x_{r_k}, y_{r_k}, \theta_{r_k})$, qui nous permettent d'obtenir une estimation initiale des coordonnées $(x_{b_0}^i, y_{b_0}^i)$ par résolution du système :

$$\tan(\theta_{r_{k-1}} + \phi_{k-1}^i) = \frac{y_{r_{k-1}} - y_{b_0}^i}{x_{r_{k-1}} - x_{b_0}^i} \quad [3-21]$$

$$\tan(\theta_{r_k} + \phi_k^i) = \frac{y_{r_k} - y_{b_0}^i}{x_{r_k} - x_{b_0}^i} \quad [3-22]$$

Ce qui nous donne finalement :

$$x_{b_0}^i = \frac{\tan(\theta_{r_{k-1}} + \phi_{k-1}^i)x_{r_{k-1}} - y_{r_{k-1}} - \tan(\theta_{r_k} + \phi_k^i)x_{r_k} + y_{r_k}}{\tan(\theta_{r_{k-1}} + \phi_{k-1}^i) - \tan(\theta_{r_k} + \phi_k^i)} \quad [3-23]$$

$$y_{b_0}^i = \tan(\theta_{r_{k-1}} + \phi_{k-1}^i)x_{b_0}^i - \tan(\theta_{r_{k-1}} + \phi_{k-1}^i)x_{r_{k-1}} + y_{r_{k-1}} \quad [3-24]$$

avec $\theta_{r_{k-1}} = \theta_{r_k}$

La matrice P_0 qui caractérise à un coefficient près la matrice de covariance est initialisée de manière à majorer très largement l'erreur qui entache l'estimation initiale. La matrice P_0 sera donc initialisée avec des termes diagonaux importants. Nous pouvons mentionner que la détermination de la position d'une balise sur deux observations d'angle d'azimut fournit dans certains cas de figure un résultat qui est très éloigné de la position théorique. Ces cas de figure correspondent notamment à deux observations angulaires bruitées très proches l'une de l'autre (problème générique du calcul par triangulation).

La méthode d'initialisation permet finalement d'obtenir une convergence plus rapide vers l'estimation finale de la position plutôt qu'en initialisant a_0 avec des valeurs arbitraires.

La première acquisition faite par le robot, au début de sa trajectoire, est utilisée pour initialiser l'ensemble des amers verticaux, et nous donne donc une carte partielle et erronée de l'environnement, sur laquelle nous allons baser la localisation suivante.

Nous présenterons les résultats expérimentaux de notre méthode de localisation et de construction de carte simultanée dans le paragraphe 3.5. Le paragraphe suivant est consacré au post traitement permettant de gérer la notion de liens existant entre deux points consécutifs estimés. Cette information nous est fournie par l'appariement de secteurs de niveaux de gris (cf. chapitre 2).

3.3.4 Traitement des liens entre amers verticaux

Nous avons donc décidé, non seulement de modéliser les amers verticaux de l'environnement, mais aussi les amers horizontaux, c'est à dire les liens qui unissent les point de la carte. Pour cela, nous allons utiliser les informations propres à nos primitives sensorielles (cf. chapitre 2).

Rappelons que lorsque deux secteurs de niveaux de gris sont appariés, cela signifie que les deux amers verticaux sont séparés par une paroi verticale (mur, radiateur, ...) ou alors du "vide" (couloir) (cf. chapitre 2).

La représentation de ces liens est gérée par un graphe, avec pour chaque nœud (les points de l'environnement), une liste de voisins droits et gauches. Nous détaillerons dans la suite de ce paragraphe l'utilité de faire la distinction entre ces deux types de voisins.

Toutefois, la cohérence des deux cas de figure distincts précédents n'est pas toujours respectée. En effet, les secteurs ne sont pas tous appariés, ce qui provoque des discontinuités au niveau de la modélisation de l'environnement. Pour pallier à ce problème, nous modéliserons la carte de l'environnement par un polygone, ainsi nous ajoutons dans un premier temps, un lien entre deux points consécutifs qui ne sont pas liés (Figure 3-8).

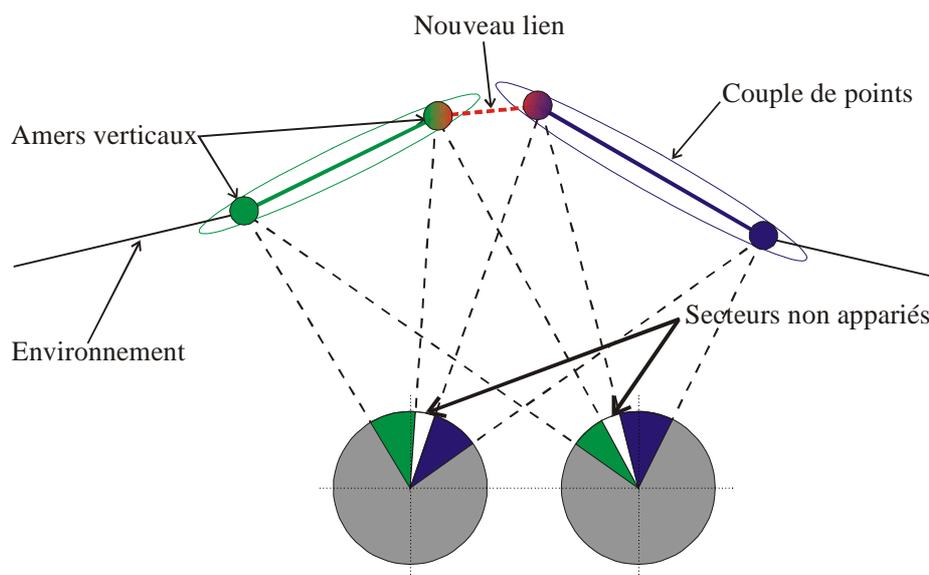


Figure 3-8 : Ajout d'un lien entre deux amers verticaux non « connectés ».

Précisons à ce niveau que cette représentation n'est pas forcément fidèle à la réalité puisque des amers verticaux de l'environnement ont pu ne pas être détectés (Figure 3-9). Toutefois, la représentation "étendue" faisant apparaître des liens est la plus précise que nous puissions obtenir avec les acquisitions réalisées. Elle évoluera au fur et à mesure du déplacement du robot, et l'algorithme que nous présentons permet de supprimer et d'ajouter des liens.

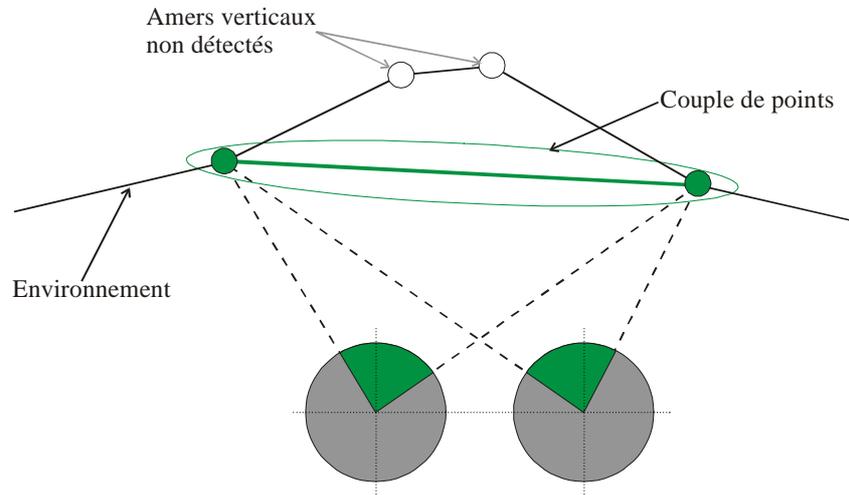


Figure 3-9 : Exemple d'amers verticaux non détectés.

Par rapport à la problématique de mise à jour incrémentale des liens, plusieurs cas doivent être distingués. Le premier cas (le plus simple) consiste à fusionner deux points observés et liés, à deux points de la carte, eux aussi liés. Il n'y a alors aucune ambiguïté possible, et le lien reste inchangé dans la carte de l'environnement (Figure 3-10).

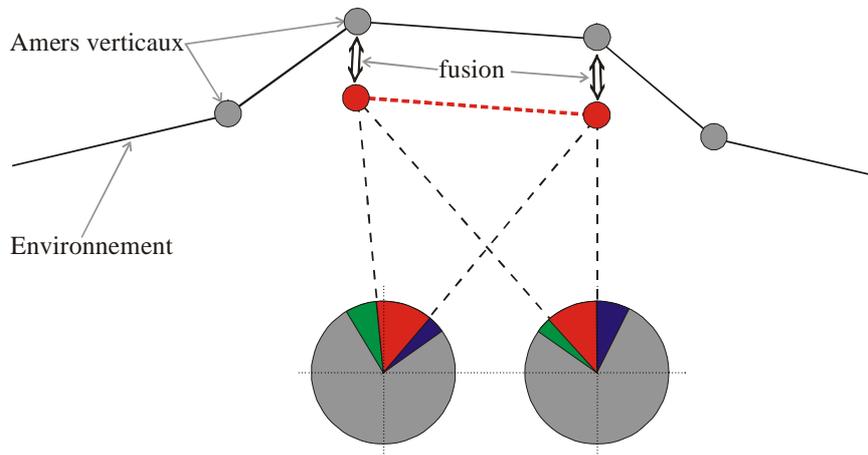


Figure 3-10 : Premier cas de fusion d'amers verticaux.

Le deuxième est déclenché lorsqu'un ou plusieurs points de la carte n'ont pas été détectés par le capteur (Figure 3-11). La stratégie est ici de ne pas modifier le graphe des liens de la carte et donc de fusionner les deux points observés mais pas le "lien sensoriel" les caractérisant. Le problème est similaire pour un point observé qui n'appartient pas à la carte.

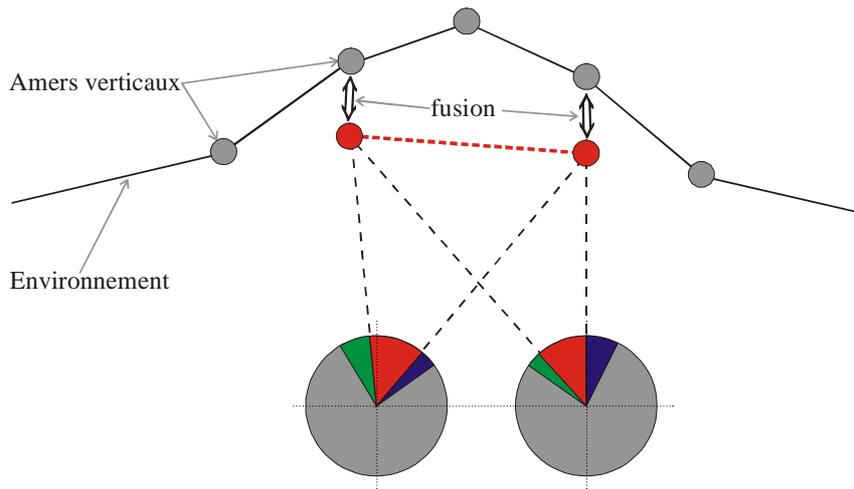


Figure 3-11 : Deuxième cas de fusion d'amers verticaux.

Enfin, le dernier cas est celui qui est le plus sujet à ambiguïté. Il s'agit du cas où il faut fusionner deux points observés, qui ne sont pas directement liés, avec deux points de la carte qui ne sont pas non plus directement liés (Figure 3-12).

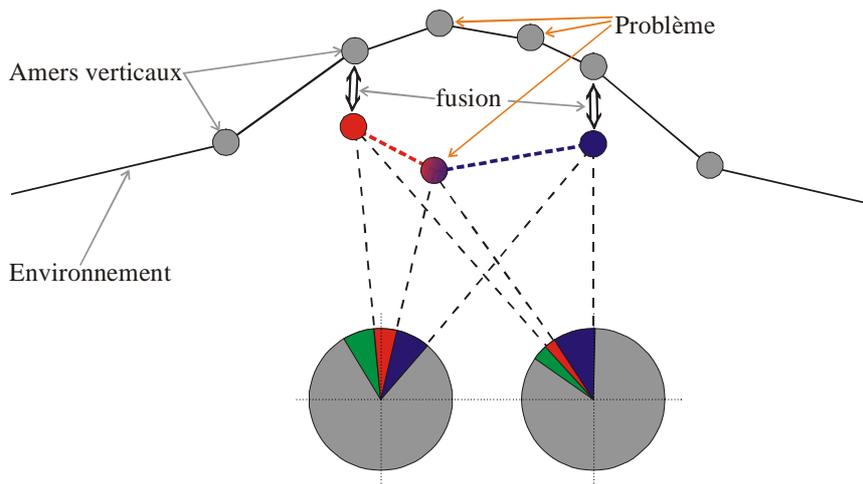


Figure 3-12 : Troisième cas de fusion d'amers verticaux.

Nous avons adopté une stratégie consistant à ne rien fusionner par rapport à l'acquisition considérée, mais de lever les ambiguïtés par rapport aux acquisitions suivantes. Cette stratégie a pour conséquence d'aboutir à une représentation "non cohérente" puisqu'il est impossible d'avoir plusieurs chaînes de liens entre deux amers d'un environnement réel (Figure 3-13). Ce problème est issu d'une erreur commise soit dans la phase de calcul des primitives, soit dans la phase de fusion.

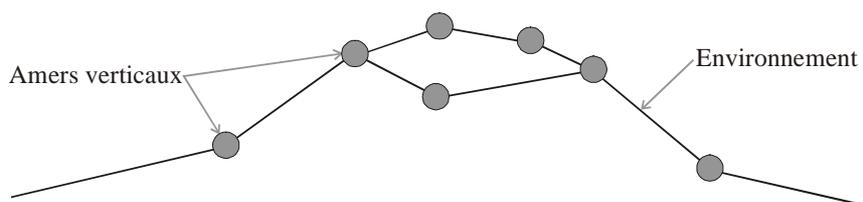


Figure 3-13 : Exemples d'amers verticaux avec plusieurs liens.

Cette aberration momentanée au niveau du graphe des liens se corrigera avec les acquisitions suivantes. Comme nous le détaillerons dans le paragraphe suivant, un filtrage sera réalisé pour effacer des données non pertinentes. Ce filtrage consiste à maintenir la cohérence du graphe des liens au cours du temps.

3.4. Traitements post-modélisation

Nous avons pu constater expérimentalement qu'il arrive fréquemment qu'un point soit intégré à la carte alors qu'il n'existe pas réellement dans l'environnement. Ce problème a plusieurs origines possibles :

- La phase de segmentation n'a pas pu fournir des résultats cohérents sur les deux images (par exemple segmentation du bord gauche d'un montant de porte, puis droit),
- La phase d'appariement a associé deux secteurs qui n'auraient pas dû l'être,
- La phase d'estimation de la configuration du robot fournit un résultat erroné,
- La phase de décision au niveau du processus de fusion a induit des intégrations en surnombre,
- Un cumul de ces sources d'erreur.

Il existe donc une multitude de raisons pour lesquelles la modélisation est incohérente. Toutefois, des constatations empiriques peuvent être exploitées pour amoindrir les sources d'erreur :

- Nous avons constaté expérimentalement que ces problèmes ne sont pas fréquents,
- Les erreurs sont aléatoires, impliquant qu'un point erroné a peu de chance de réapparaître une deuxième fois,
- Les points erronés le sont en général fortement, ce qui favorise leur repérage : l'occurrence liée à l'observation est un facteur fortement discriminant.

Nous proposons une solution dans le paragraphe suivant qui permet d'optimiser le maintien de la cohérence de la carte.

3.4.1 *Maintien de la cohérence par filtrage*

Nous avons intégré un algorithme de filtrage que nous appliquons une fois la mise à jour des points et du graphe des liens réalisée. Cet algorithme est basé sur la gestion d'une pondération liée à l'occurrence des observations. Il se décline de la manière suivante :

- Quand un nouveau point est inséré dans la carte, il a un poids de 1 (nombre de fois où il a été observé), et il est daté à 0. La date représente ici le nombre de cycles de localisation/modélisation depuis la création du point,
- Après chaque étape de mise à jour, l'ensemble des dates pour tous les points de la carte est incrémenté de 1,
- Lorsqu'un point étiqueté avec un poids inférieur à un seuil prédéfini b , n'a plus été observé depuis un temps t , alors son poids est décrémenté de 1,
- Finalement, les points avec un poids de 0 sont effacés de la carte.

Le problème essentiel de cet algorithme est lié à l'initialisation des valeurs de b et de t : Ces deux valeurs influencent de manière significative le nombre de points de la carte. Une valeur de b trop petite laisse beaucoup de points aberrants alors qu'une valeur trop grande a tendance à supprimer des points valides. Quant au paramètre t , il influe évidemment sur la persistance des points erronés au sein de la carte. Etant donné qu'ils peuvent être utilisés non seulement pour la localisation mais aussi lors de l'étape de mise à jour, un temps relativement court supprimera les points avant même qu'ils puissent être confirmés ou infirmés par de nouvelles observations. Nous avons fixé expérimentalement ces valeurs à $b=2$ et $t=2$.

Un dernier point reste alors à résoudre : la suppression d'un point de la carte et les conséquences sur les liens existants.

3.4.2 Gestion des liens dans le cas d'une suppression d'un point de la carte

Nous avons vu dans la partie précédente que les liens représentent en fait des parois verticales probables qui peuvent exister entre les amers verticaux de l'environnement.

Dans les cas les plus courants, la suppression d'un point ne pose pas de problèmes. En effet, la majorité de ceux-ci n'ayant que deux voisins, il suffit de relier ces deux voisins entre eux et de supprimer le point (Figure 3-14).

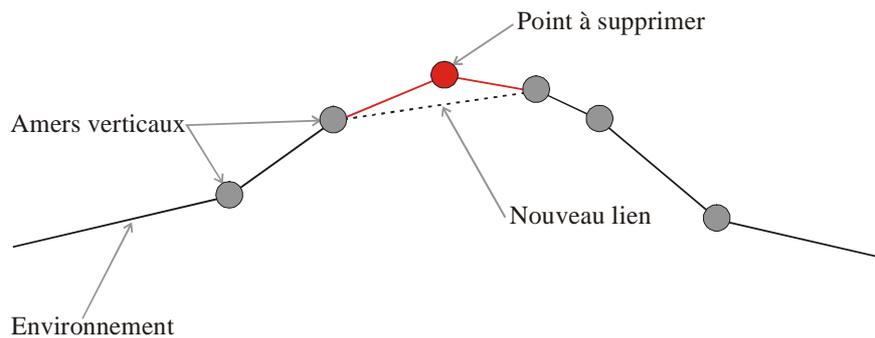


Figure 3-14 : Suppression d'un point de la carte avec ajout d'un lien.

Le problème se pose quand un point a plus de deux liens (troisième cas d'intégration de lien de l'étape précédente). Il faut alors prendre en compte la suppression de données redondantes à l'intérieur du graphe (Figure 3-15). Nous devons alors parcourir le graphe représentant l'environnement à partir des voisins gauches du point à supprimer, et rechercher les voisins droits de ce même point. Ceci nous donnera l'information sur les liens éventuels à ajouter ou à enlever après la suppression d'un nœud.

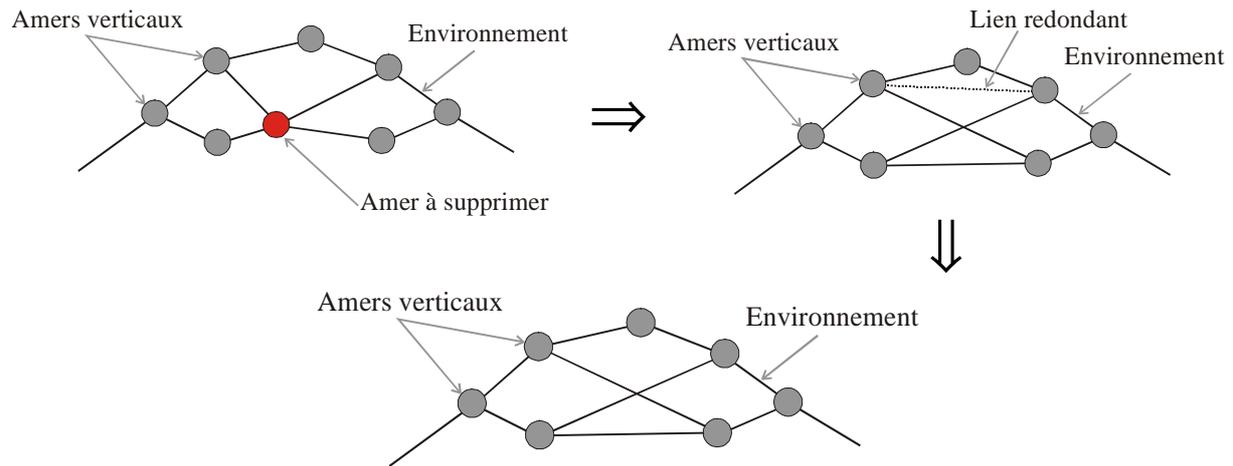


Figure 3-15 : Suppression d'un point de la carte avec liens multiples.

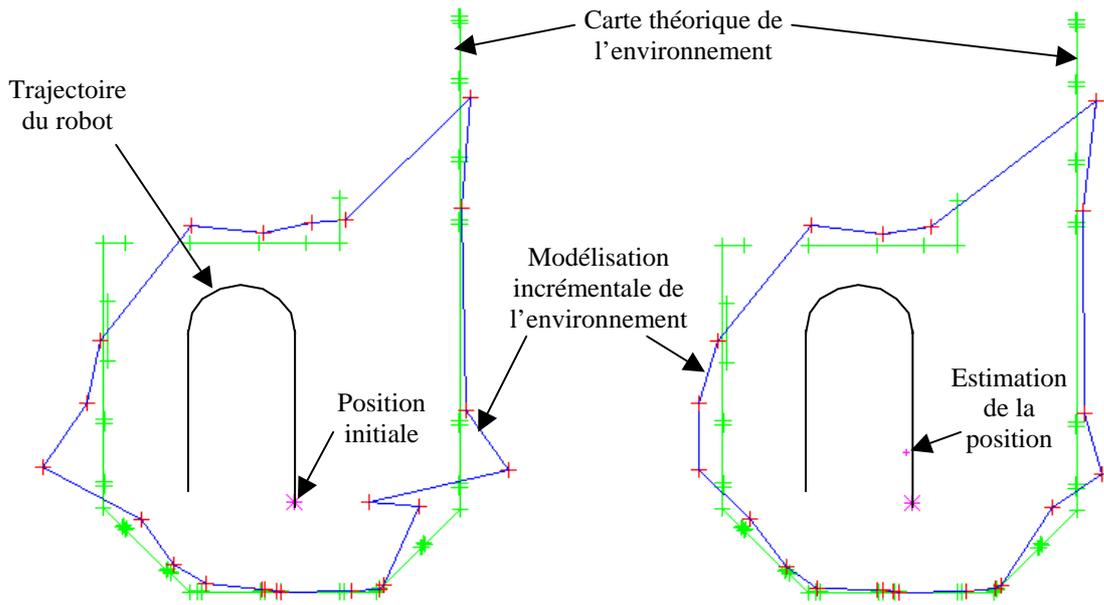
Finalement, le module de construction de carte proposé permet de gérer d'une part l'estimation des paramètres des primitives, mais aussi les liens existants entre chacune de ces primitives. Nous allons maintenant présenter les différents résultats expérimentaux que nous avons réalisés dans deux environnements différents.

3.5. Résultats expérimentaux

Afin de valider notre approche, nous avons testé notre méthode de localisation et de modélisation incrémentale dans deux types d'environnement structuré différents. La première série de tests a été réalisée dans l'environnement fermé et structuré du chapitre précédent. Il s'agit là d'un hall de notre laboratoire qui dessert des bureaux. La deuxième a été réalisée dans un couloir fortement symétrique de grande dimension (couloir de 45 mètres de long sur 3,5 mètres de large). L'essentiel des amers sont de type portes et piliers.

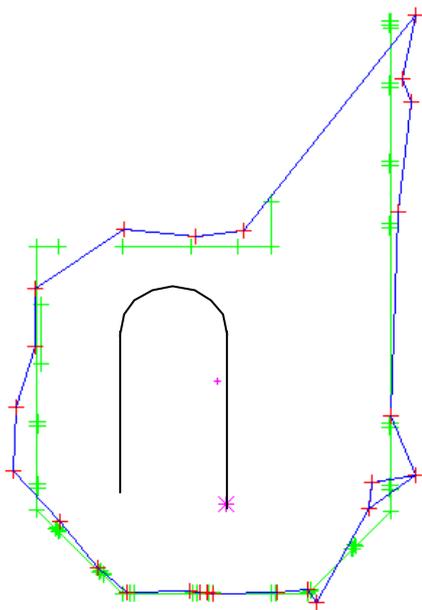
3.5.1 Tests en environnement fermé et structuré sur de petits trajets

Cette première expérimentation est basée sur huit acquisitions stéréoscopiques (seize images), réalisées sur un aller et retour du robot. Le contexte de cette première série d'acquisitions était, dans un premier temps, de valider les choix établis pour la mise à jour de la carte. Il nous a permis en outre de nous assurer que la localisation restait cohérente avec une absence de connaissance *a priori* (Figure 3-16).

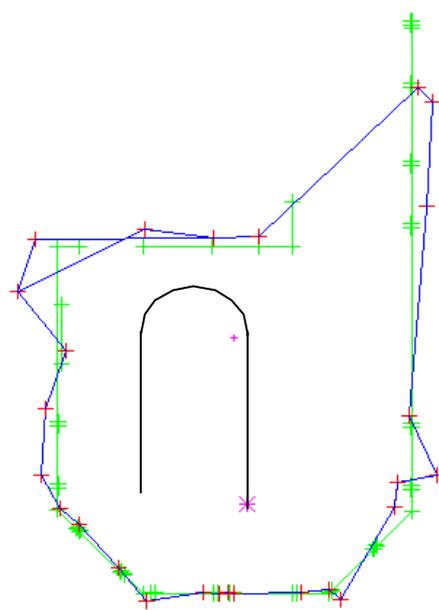


Acquisition 1

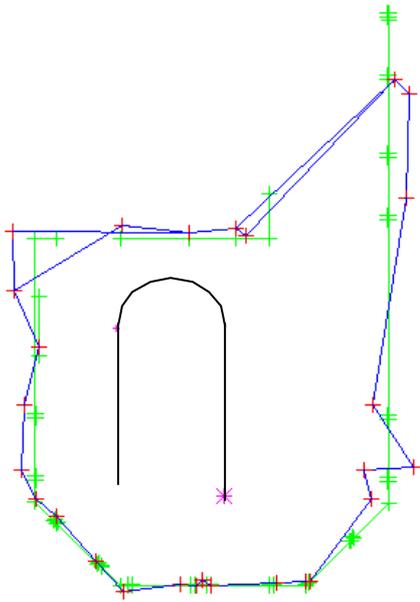
Acquisition 2



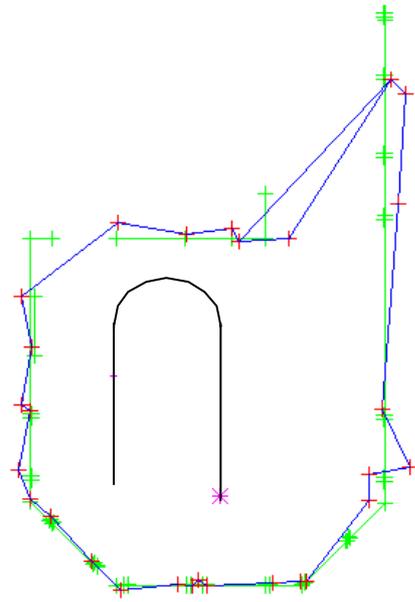
Acquisition 3



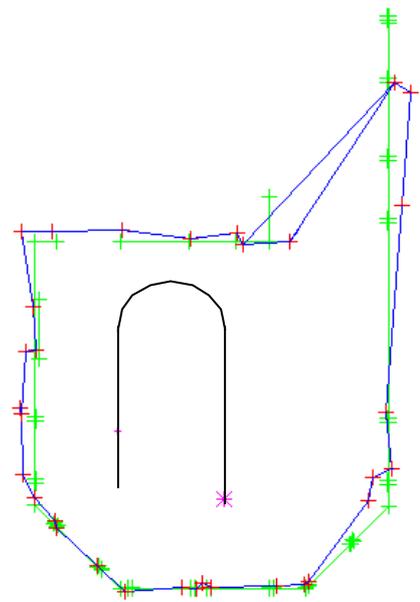
Acquisition 4



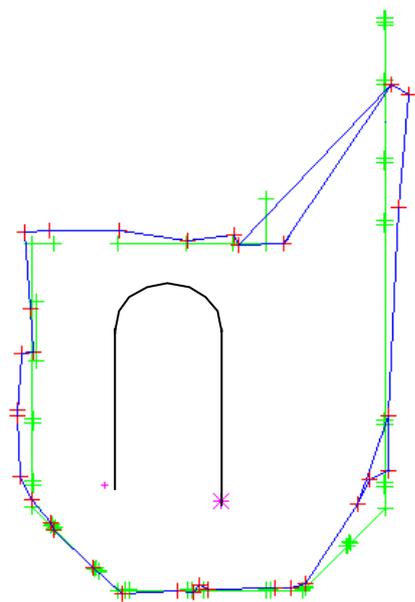
Acquisition 5



Acquisition 6



Acquisition 7



Acquisition 8

Figure 3-16 : Résultats de la modélisation de l'environnement avec huit acquisitions stéréoscopiques.

Nous pouvons constater dans un premier temps que le robot a réussi à se localiser de façon robuste sur la totalité des acquisitions et ceci sans connaissance *a priori* sur l'environnement. Cette constatation était fondamentale pour pouvoir modéliser correctement l'environnement. De façon plus approfondie, nous pouvons qualitativement affirmer que l'ensemble des positions estimées sont proches des positions réelles : la précision est importante.

D'autre part, nous pouvons remarquer que l'environnement est correctement modélisé puisque la quasi totalité des amers verticaux a été caractérisée avec précision par un point.

Ces constatations positives ont amené à tester notre méthode de construction de carte sur un environnement de plus grande dimension. Nous constatons qu'il n'y a pas « d'explosion combinatoire » au niveau de l'intégration de ces points, et qu'il n'y a donc pas de problème de surnombre. Enfin, nous avons des liens entre les primitives qui correspondent bien à une « réalité terrain ».

3.5.2 Résultats en environnement de type couloir sur des trajets importants

Le deuxième environnement dans lequel nous avons effectué des acquisitions est un couloir d'environ 45 mètres de long sur 3,5 mètres de large, et dans lequel les murs sont composés de panneaux et de piliers (Figure 3-17 et Figure 3-18). Le couloir est donc constitué d'éléments répétitifs et souvent symétriques. Ce dernier point compliquera les phases de localisation/modélisation et permettra de mettre en évidence la robustesse de notre approche. Enfin, le test s'effectuant sur une large distance, des problèmes de dérives cumulatives mis en évidence par Fox risquent d'apparaître. Les dérives cumulatives sont liées à l'absence totale de connaissance de l'environnement.

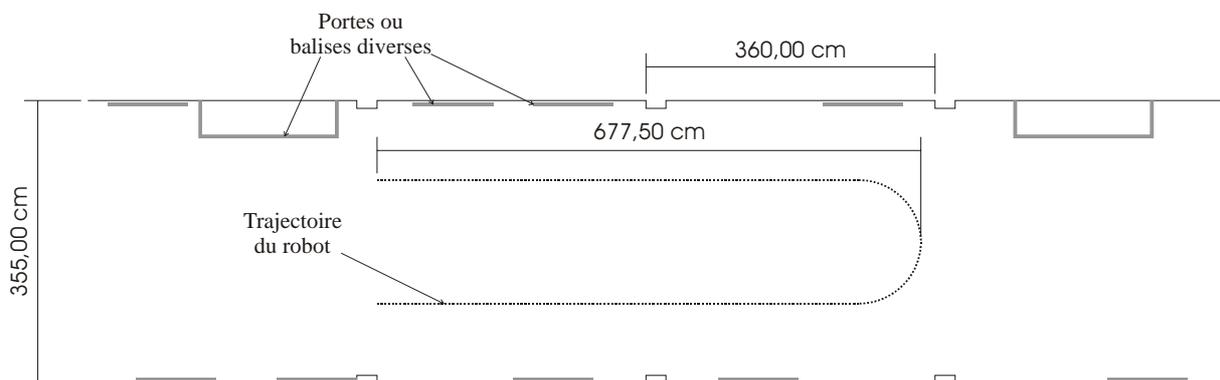


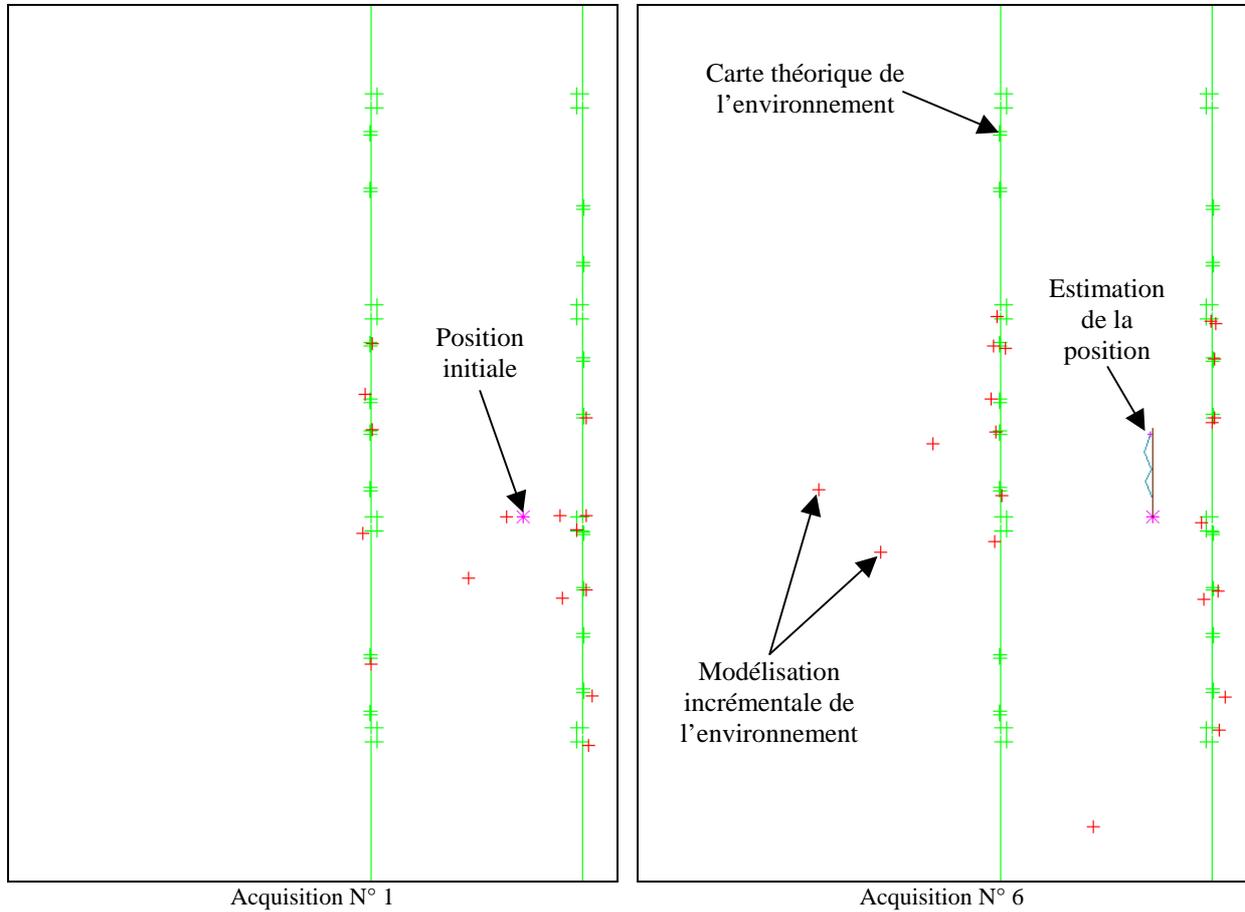
Figure 3-17 : Deuxième environnement d'expérimentation.

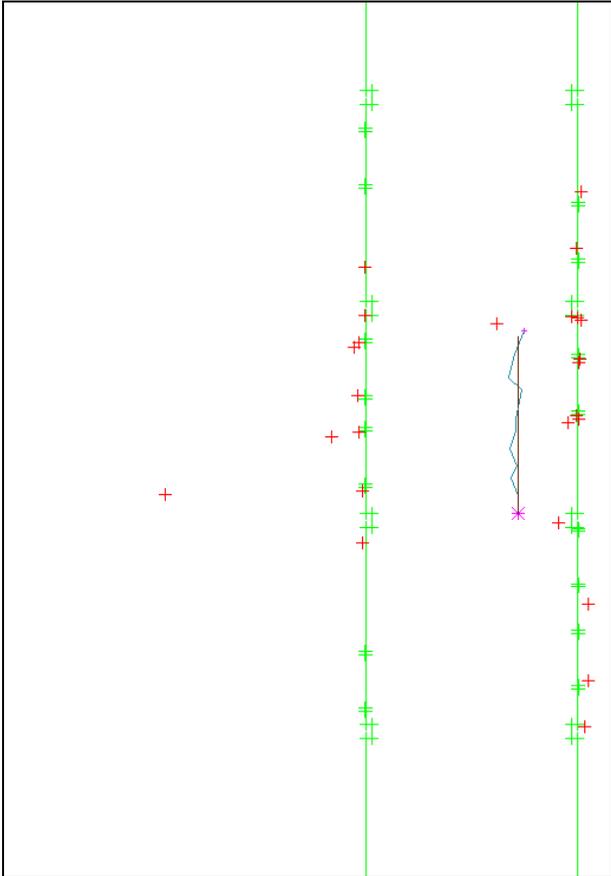


Figure 3-18 : Photos du deuxième environnement d'expérimentation.

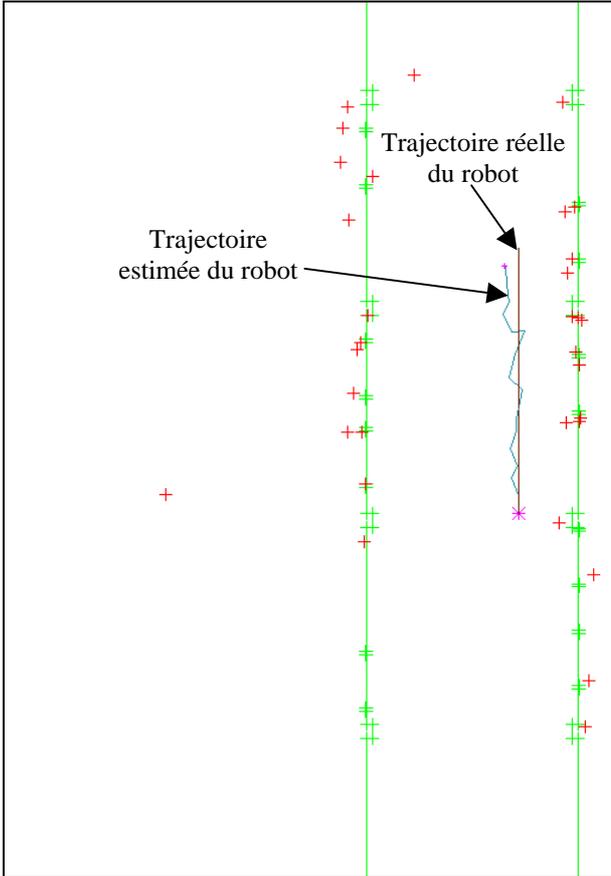
Cette fois-ci, nous avons réalisé 45 acquisitions dans cet environnement, la trajectoire du robot est toujours du type aller et retour, sur une distance totale d'environ quatorze mètres (Figure 3-17). Sur les deux portions de ligne droite, une acquisition est prise tous les 30

centimètres, et sur la portion de courbe tous les 60 centimètres. L'axe passant par les deux images est toujours perpendiculaire à la trajectoire. L'odométrie est utilisée afin d'éviter l'explosion combinatoire de l'algorithme de localisation, mais aussi pour s'assurer que la position recherchée reste cohérente avec la précédente.

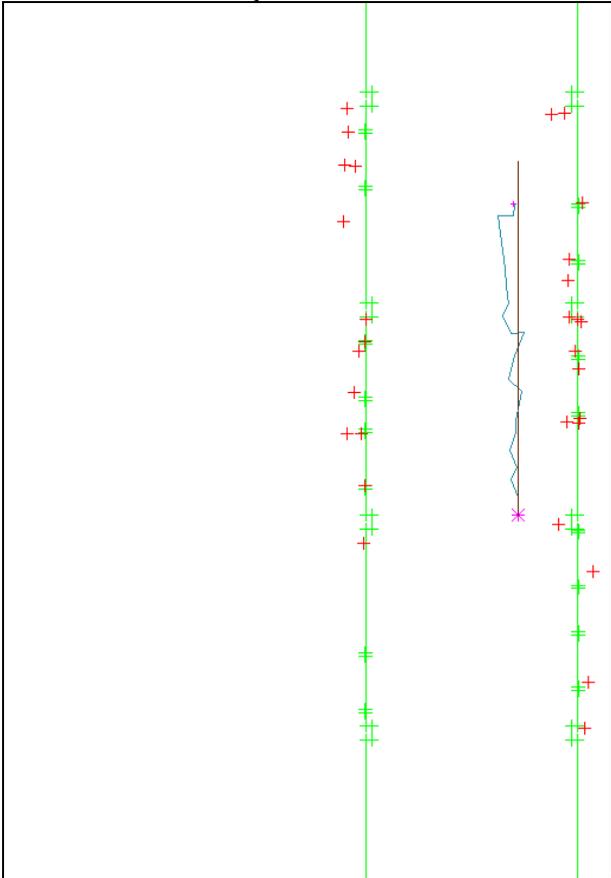




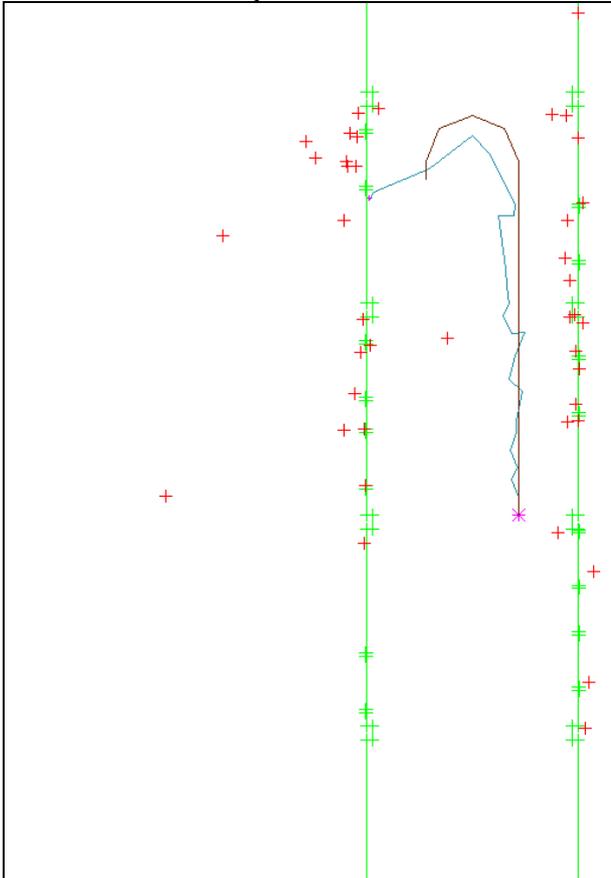
Acquisition N° 11



Acquisition N° 16



Acquisition N° 21



Acquisition N° 26

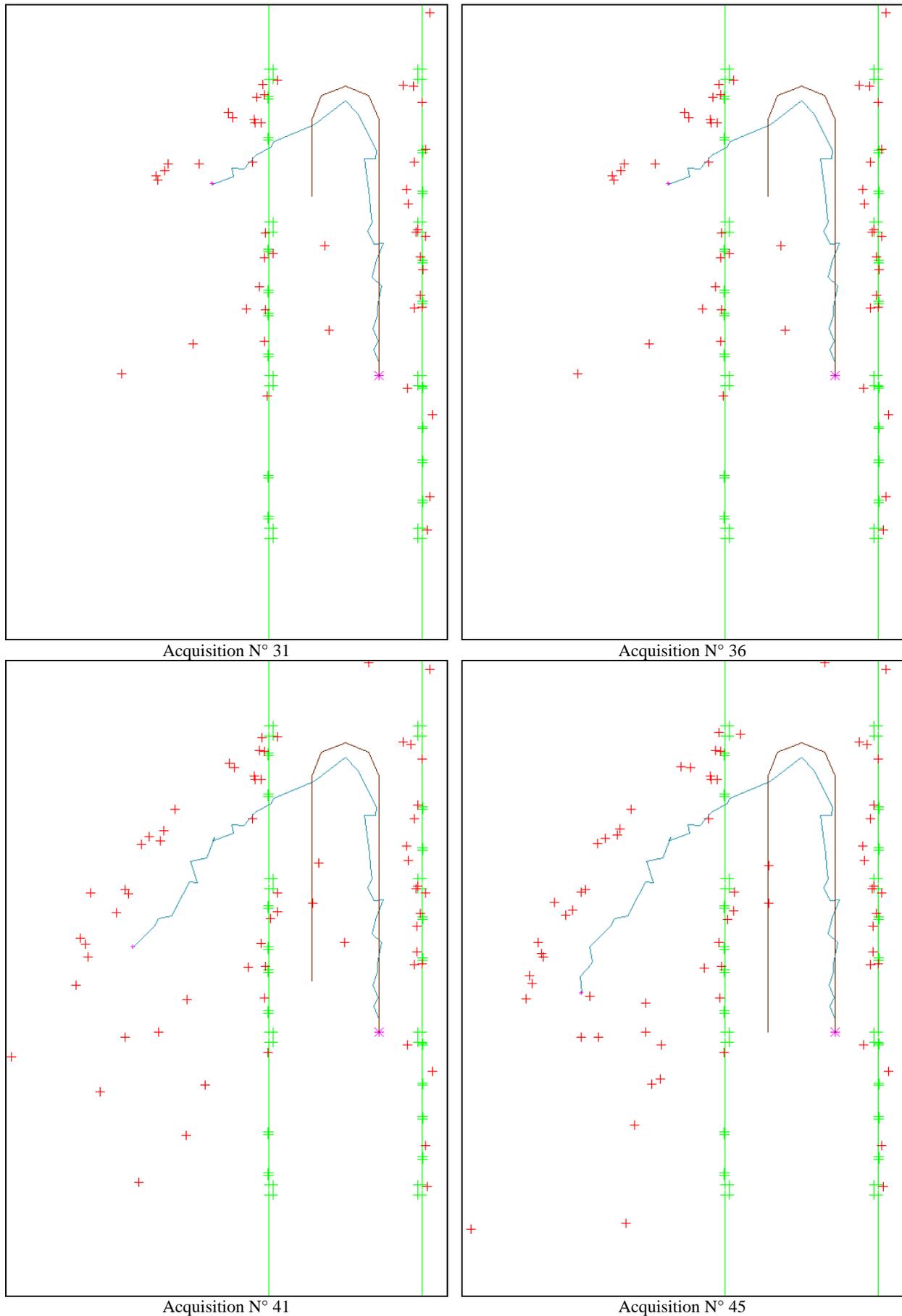


Figure 3-19 : Résultats de la modélisation de l'environnement sur 45 acquisitions stéréoscopiques.

Nous pouvons noter par rapport à cette deuxième série d'acquisitions stéréoscopiques que sur l'intégralité de la partie rectiligne de la trajectoire, le processus de localisation/modélisation permet d'une part de localiser le robot avec une précision importante et d'autre part d'obtenir une représentation robuste de l'environnement très proche de la réalité terrain, ceci même au bout de la 20^{ème} acquisition (soit une distance parcourue d'environ 7 mètres par rapport à la position initiale du robot). Nous constatons néanmoins que le processus simultané de localisation/modélisation diverge au bout d'un certain temps de navigation du robot. En effet, à partir de la 21^{ème} acquisition, qui correspond à l'amorce du virage, la position du robot calculée commence à s'éloigner de la position réelle du robot. Il paraît évident qu'à partir de cet instant le phénomène d'erreurs cumulatives tendra à faire diverger le système.

Nous pouvons tenter d'expliquer le problème de divergence par rapport aux points suivants :

- Le milieu d'évolution est défavorable puisqu'il comporte des éléments répétitifs. En outre il s'agit d'un couloir, donc comportant des amers verticaux uniquement sur deux cotés, ce qui limite obligatoirement la phase de localisation (répartition non homogène des amers). Enfin, la trajectoire relativement longue implique que le robot ne peut se recalculer sur des éléments qu'il a « modélisé » sur peu d'acquisitions,
- Le capteur est disposé perpendiculairement au sens de déplacement du robot, ce qui diminue la précision de la modélisation. En effet, cette précision obtenue sur les amers modélisés est meilleure lorsqu'ils sont détectés devant le capteur, c'est à dire perpendiculaire à l'axe passant par les deux images. Ce problème est lié aux configurations angulaires défavorables,
- L'estimation de position n'étant pas réalisée au sens des moindres carrés (cf. chapitre 2), ceci induit une erreur systématique qui deviendra cumulative tout au long du processus de modélisation incrémentale. Une erreur de calcul sur la position entraîne obligatoirement des erreurs sur l'estimation des coordonnées des points de la carte.

Les deux premiers points discutés ici sont uniquement liés aux conditions d'expérimentations alors que le troisième concerne la méthode de localisation.

Nous constatons que le maintien de la cohérence de la carte est optimisé : nous n'avons pas d'intégration en surnombre de primitives au sein de la carte. Néanmoins, nous avons mis en évidence le problème quasi inévitable de dérive qui apparaît sur des distances importantes, propre lui aussi à ce type de paradigme.

L'amélioration de ces résultats devra passer par trois points qui n'ont pas été intégrés dans notre paradigme :

- La modélisation des erreurs de localisation, ainsi que celles liées aux primitives estimées,
- L'interaction entre les erreurs de localisation et celles de modélisation,
- La prise en compte des erreurs sur les primitives estimées dans le processus de mise en correspondance.

Dans les chapitres suivants, nous proposons une approche permettant d'intégrer tous ces points.

3.6. Conclusion

Nous avons dans ce chapitre développé une méthode de modélisation de l'environnement, sans connaissances *a priori*, basée sur l'utilisation de la théorie de Dempster-Shafer pour la mise en correspondance et sur les M.C.R. pour l'estimation des paramètres des primitives.

Dans un premier temps, notre algorithme consiste à calculer quelles sont les possibilités d'appariement entre les amers observés et ceux de la carte. Nous appliquons ensuite une méthode de décision. Les critères utilisés dans cette étape décisionnelle sont uniquement basés sur les angles d'azimut des amers verticaux.

Dans un deuxième temps, nous utilisons les moindres carrés récursifs pour mettre à jour les coordonnées d'un point de la carte avec une observation qui lui est associée. Les paramètres estimés par cette méthode sont les coordonnées des amers. Nous obtenons finalement une carte à l'instant n résultant de la fusion de la carte à l'instant $n-1$ et du modèle sensoriel stéréoscopique obtenu à l'instant n .

Nous avons constaté de façon expérimentale que, dans un environnement structuré, la construction incrémentale de la carte se fait correctement et que la majorité des amers est correctement modélisée. Par contre, dans un environnement structuré, symétrique et sur des distances importantes, la méthode montre ses limites surtout vis à vis du phénomène intrinsèque au processus de localisation/modélisation en environnement inconnu : la dérive induite par les erreurs cumulatives.

Nous avons identifié un certain nombre de points à prendre en compte dans notre paradigme pour réduire ce problème de dérive. Il s'agit notamment d'intégrer la notion de quantification d'erreur au niveau de l'estimation des paramètres des primitives de la carte. L'interaction entre ces deux quantificateurs d'imprécision sera un facteur prépondérant à prendre en compte. Dans cette optique, à la différence des solutions classiquement utilisées dans la littérature pour gérer l'imprécision qui sont de type probabiliste, nous avons porté notre choix sur des méthodes ensemblistes. C'est ce que nous allons développer dans les deux prochains chapitres.

Chapitre 4

Localisation d'un robot mobile par inversion ensembliste

Sommaire partiel

4.1.	INTRODUCTION	96
4.2.	RAPPELS SUR LA THEORIE ENSEMBLISTE.....	97
4.2.1	<i>Définition d'un problème d'inversion ensembliste.....</i>	97
4.2.2	<i>Synthèse sur l'analyse par intervalles.....</i>	98
4.2.2.1.	Opérations sur les intervalles.....	98
4.2.2.2.	Fonctions d'inclusion	99
4.2.3	<i>L'algorithme SIVIA.....</i>	101
4.3.	LOCALISATION D'UN ROBOT MOBILE PAR L'UTILISATION DE SIVIA.....	105
4.3.1	<i>Définition du problème.....</i>	105
4.3.2	<i>Caractérisation du modèle d'erreur.....</i>	106
4.3.2.1.	Premier modèle d'erreur pour la localisation.	106
4.3.2.2.	Deuxième modèle d'erreur pour la localisation.	108
4.3.3	<i>Algorithme de localisation.....</i>	110
4.3.4	<i>Comment utiliser l'information odométrique ?.....</i>	112
4.4.	RESULTATS EXPERIMENTAUX	114
4.4.1	<i>Résultats expérimentaux utilisant le premier modèle d'erreur.....</i>	114
4.4.2	<i>Résultats expérimentaux utilisant le second modèle d'erreur.....</i>	116
4.4.3	<i>Comparaison des deux modèles d'erreur.....</i>	117
4.5.	CONCLUSION	118

4.1. Introduction

Nous avons présenté dans les chapitres précédents une démarche permettant au robot de construire une carte de l'environnement au fur et à mesure des acquisitions. La possibilité de générer une carte de l'environnement reposait sur un algorithme de localisation robuste prenant en compte un critère d'optimisation lié à l'erreur en distance entre les points.

Nous avons pu constater que la méthode précédente ne permettait ni d'intégrer la notion d'erreur sur l'estimation de position du robot, ni sur l'estimation de positions des amers verticaux. Cette notion d'imprécision, calculée à partir des données sensorielles, sera bien sûr utilisée dans la phase de modélisation.

L'absence de prise en compte de l'imprécision rend incomplète la démarche de localisation/modélisation puisqu'elle ne permet pas de traiter un facteur prépondérant : l'influence de l'erreur d'estimation de position du robot sur l'estimation des paramètres des amers verticaux.

Il y a en effet une interaction évidente entre les erreurs commises sur la position du robot et celles introduites par le calcul des positions des amers. C'est cette interaction qui, dans le processus de construction incrémental, est à l'origine des erreurs cumulatives.

Nous avons donc voulu proposer une alternative permettant de prendre en compte la notion d'imprécision dès l'étape de localisation. Plusieurs formalismes sont alors applicables pour modéliser cette imprécision :

- Statistique : Une loi de probabilité est définie et les points sont considérés comme des variables aléatoires vérifiant cette loi. Cette première solution consiste à gérer une modélisation statistique de l'imprécision sur l'estimation de position et sur l'estimation des paramètres des amers verticaux. Le problème de "fusion statistique" dans le temps, des estimations et de leurs imprécisions associées est alors posé. Une solution, désormais plus que classique pour gérer cette "fusion temporelle", est d'utiliser le filtrage de Kalman. La quantification des imprécisions est alors gérée avec la notion de matrice de covariance. Nous avons souhaité nous démarquer de ce type d'approche très largement utilisée en robotique mobile.
- Analytique : "L'ensemble solution" est calculé analytiquement à partir de la résolution d'équations et/ou d'inéquations. Par rapport à cette approche, nous avons commencé à développer une méthode, basée sur la présence d'un ensemble d'équations, fonction des paramètres possibles d'erreur. Nous n'avons pas développé cette solution qui nous obligeait à gérer des systèmes d'équations beaucoup trop complexes pour ce que nous voulions faire.
- Garantie : "Un ensemble solution" simple est calculé et permet de contenir de manière garantie l'ensemble solution. Nous avons finalement opté pour cette dernière solution, dont l'analyse par intervalle fait partie. Cette méthode nous permet d'avoir un ensemble qui contient obligatoirement la solution, et ce en manipulant des primitives relativement simples telles que des pavés par exemple.

Ce chapitre se découpera donc en trois parties. Dans un premier temps nous exposerons la théorie mathématique concernant l'analyse par intervalles, et détaillerons dans ce cadre la notion de fonctions d'inclusion et l'algorithme SIVIA. Nous expliquerons ensuite comment nous utiliserons l'inversion ensembliste pour résoudre l'unique problème de la localisation du robot. Enfin, nous discuterons divers résultats expérimentaux et nous exposerons en quoi la méthodologie utilisée pour la localisation sera utilisable pour la modélisation.

4.2. Rappels sur la théorie ensembliste

4.2.1 Définition d'un problème d'inversion ensembliste

Il est possible de constater que de nombreux problèmes ne sont pas définis comme la recherche d'une solution $y=f(x)$ pour x connu, mais plutôt comme la recherche des antécédents de y par la fonction réciproque de f . On parle alors d'un problème d'inversion ensembliste.

Nous pouvons le définir de la façon suivante :

Soient : ✓ une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}^p$
 ✓ Y un sous-ensemble fermé de \mathbb{R}^p

Le problème de l'inversion ensembliste consiste à caractériser l'ensemble X image réciproque de Y par f (Figure 4-1). Cet ensemble X , appelé ensemble solution, est défini par :

$$X = f^{-1}(Y) = \{x \in \mathbb{R}^n \mid f(x) \in Y\} \quad [4-1]$$

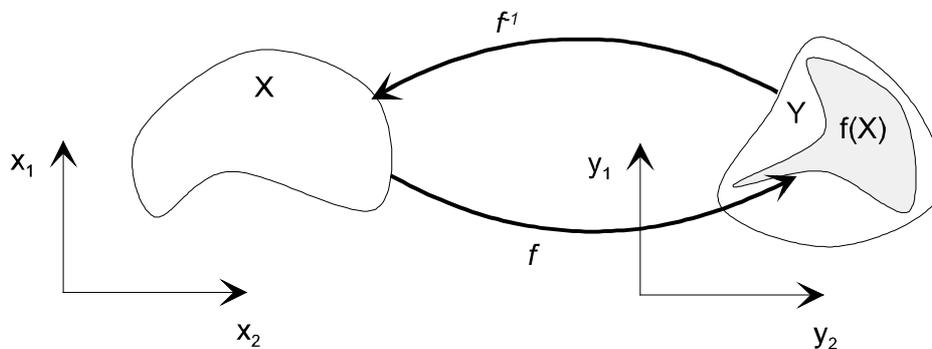


Figure 4-1 : Illustration d'un problème d'inversion ensembliste.

L'image directe $f(X)$ de l'ensemble X par la fonction f est définie par :

$$f(X) = \{f(x) \mid x \in X\} \quad [4-2]$$

Nous avons donc, d'après les propriétés élémentaires de la théorie des ensembles :

$$f(X) \subset Y \quad [4-3]$$

Ainsi, les problèmes ensemblistes admettent pour solution non pas un nombre ou un vecteur, mais un ensemble solution.

Un algorithme d'inversion ensembliste, utilisant l'analyse par intervalles, a été développé par L. Jaulin durant ses travaux de thèse [JAU94]. C'est cet algorithme que nous utiliserons après en avoir rappelé le fonctionnement.

4.2.2 Synthèse sur l'analyse par intervalles

Lorsque des calculs numériques sont réalisés, des erreurs provenant principalement de données imprécises peuvent apparaître et se propager, altérant ainsi le résultat final du calcul. Cette constatation est à l'origine du calcul par intervalles : tout nombre imprécis est représenté par un intervalle le contenant de façon garantie. Nous pouvons donc considérer un intervalle soit comme un ensemble infini de réels, soit comme la représentation d'un réel incertain.

4.2.2.1. Opérations sur les intervalles

Un **intervalle** $[x]$ est défini par une paire ordonnée de nombres réels correspondant à ses extrémités et décrit un ensemble contenant une infinité de nombres réels. C'est un sous-ensemble connexe fermé et borné :

$$[x] = [x^-, x^+] = \{x \in \mathbb{R} \mid x^- \leq x \leq x^+\} \quad [4-4]$$

où x^- et x^+ sont respectivement les bornes inférieures et supérieures. L'ensemble des intervalles réels sera noté $\mathbb{II} \mathbb{R}$.

Un intervalle est dit dégénéré si $x^- = x^+$; on a alors $[x] \in \mathbb{R} \subset \mathbb{II} \mathbb{R}$. Les intervalles étant des ensembles, les notions d'égalité, d'appartenance, d'inclusion stricte et large ainsi que l'intersection sont définies. Toutefois, la réunion de deux intervalles n'est pas obligatoirement un intervalle, nous définissons alors l'union convexe qui correspond au plus petit intervalle contenant l'union des deux intervalles.

$$[x] \cup [y] = [\min(x^-, y^-), \max(x^+, y^+)] \quad [4-5]$$

Nous pouvons caractériser un intervalle par :

- son centre $m([x]) = \frac{x^- + x^+}{2}$, [4-6]

- sa longueur $w([x]) = x^+ - x^-$, [4-7]

- son rayon $r([x]) = \frac{x^+ - x^-}{2}$. [4-8]

Les opérations arithmétiques de base telles que l'addition, la soustraction, la multiplication et la division ont été étendues aux intervalles par R.E. Moore en 1979 [MOO79]. Les intervalles peuvent ainsi être manipulés comme des nombres.

Soient $[x]$ et $[y]$ deux intervalles. Nous montrons facilement que :

$$\begin{aligned}
 [x] + [y] &= [x^- + y^-, x^+ + y^+] \\
 -[x] &= [-x^+, -x^-] \\
 [x] - [y] &= [x^- - y^+, x^+ - y^-] \\
 [x] \times [y] &= [\min(x^- \times y^-, x^- \times y^+, x^+ \times y^-, x^+ \times y^+), \max(x^- \times y^-, x^- \times y^+, x^+ \times y^-, x^+ \times y^+)] \\
 1 / [x] &= [1/x^+, 1/x^-] \text{ si } 0 \notin [x] \\
 [x] / [y] &= [x] \times 1/[y]
 \end{aligned}
 \tag{4-9}$$

La multiplication et l'addition sont toutes deux associatives et commutatives, toutefois la distributivité de la multiplication par rapport à l'addition n'est pas toujours vérifiée :

$$[x] \times ([y] + [z]) \subseteq [x] \times [y] + [x] \times [z]
 \tag{4-10}$$

Il est à noter que $[x]^2 \neq [x] \times [x]$: par exemple si $[x] = [-1,1]$, alors $[x]^2 = [0,1]$ et $[x] \times [x] = [-1,1]$. Dans le premier cas c'est l'intervalle complet qui est au carré, alors que dans le deuxième cas nous considérons les variations des deux intervalles comme indépendantes.

De plus, si $0 \in [x]$, l'ensemble $\{ 1/x \mid x \in [x] \}$ ne peut pas être représenté par un intervalle tel que nous l'avons défini. Dans ce cas nous pourrions utiliser la notion d'intervalle étendu :

$$\overline{II \mathbb{R}} = II \mathbb{R} \cup \{[-\infty, r] \mid r \in \mathbb{R}\} \cup \{[l, +\infty] \mid l \in \mathbb{R}\} \cup \{[-\infty, +\infty]\}
 \tag{4-11}$$

Le problème est ici que l'ensemble solution peut être composé de plusieurs intervalles, et nécessitera donc un traitement particulier.

Nous pouvons étendre la notion d'intervalle à des vecteurs, on parle alors de pavés, ou encore de vecteurs d'intervalles. Un pavé $[X] \in II \mathbb{R}^n$, de dimension n s'écrit sous la forme :

$$[X] = \begin{pmatrix} [x]_1 \\ \dots \\ [x]_n \end{pmatrix}
 \tag{4-12}$$

La longueur $w([X])$ du pavé sera alors :

$$w([X]) = \max_{i=1, \dots, n} w([x]_i)
 \tag{4-13}$$

De la même façon, l'ensemble des matrices d'intervalles de dimension $n \times m$ est noté $II \mathbb{R}^{n \times m}$.

4.2.2.2. Fonctions d'inclusion

Il est souvent difficile de calculer l'image d'un intervalle $[x]$ par une fonction. En effet, il faut souvent passer par des calculs complexes. Par contre, nous pouvons aisément construire une fonction d'inclusion dont l'image contient à coup sûr l'image directe de cet intervalle. Le même raisonnement peut être tenu pour un pavé.

Soient une fonction :

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^p \quad [4-14]$$

et $[X]$ un pavé de \mathbb{R}^n , la fonction :

$$f^I: \mathbb{R}^n \rightarrow \mathbb{R}^p \quad [4-15]$$

est une **fonction d'inclusion** de f si et seulement si elle vérifie (Figure 4-2) :

$$\forall [X] \in \mathbb{R}^n \quad f([X]) \subset f^I([X]) \quad [4-16]$$

Une fonction d'inclusion f^I est dite monotone si elle satisfait la condition suivante :

$$[x] \subset [y] \Rightarrow f^I([x]) \subset f^I([y]) \quad [4-17]$$

Elle sera convergente si elle vérifie :

$$w([x]) \rightarrow 0 \Rightarrow w(f^I([x])) \rightarrow 0 \quad [4-18]$$

Pour une même fonction f , il existe une infinité de fonctions d'inclusions. Parmi cet ensemble, il en existe une seule qui soit minimale au sens de l'inclusion : c'est la **fonction d'inclusion minimale** notée f^{I*} . Le pavé $f^{I*}([X])$ est le plus petit pavé de \mathbb{R}^p qui contient $f([X])$, c'est donc le pavé enveloppe de cet ensemble, cf. Figure 4-2.

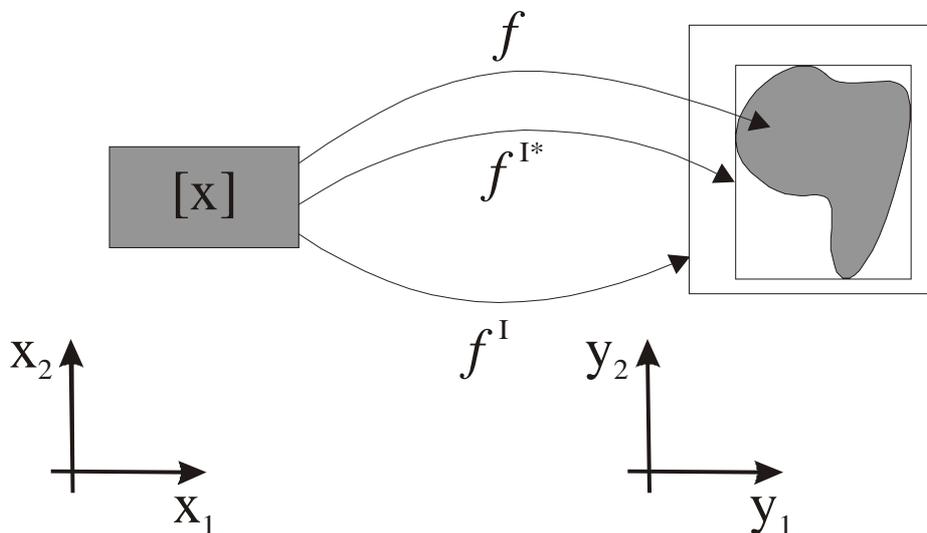


Figure 4-2 : Illustration de la notion de fonction d'inclusion.

Lorsque la fonction f est composée de fonctions élémentaires (+, -, cos, ...), nous pouvons construire une **fonction d'extension naturelle** obtenue en remplaçant les valeurs réelles par les intervalles correspondants. En général, cette fonction d'inclusion n'est pas minimale (elle est même plutôt pessimiste).

Par exemple si nous considérons plusieurs formulations de la même fonction [KIE99] :

$$\begin{aligned}
 f_1(x) &= x(x + I) \\
 f_2(x) &= x \times x + x \\
 f_3(x) &= x^2 + x \\
 f_4(x) &= \left(x + \frac{1}{2}\right)^2 - \frac{1}{4}
 \end{aligned}
 \tag{4-19}$$

En prenant comme intervalle $[x] = [-I, I]$, on obtient :

$$\begin{aligned}
 f_1([x]) &= [x]([x] + 1) = [-2, 2] \\
 f_2([x]) &= [x] \times [x] + [x] = [-2, 2] \\
 f_3([x]) &= [x]^2 + [x] = [-1, 2] \\
 f_4([x]) &= \left([x] + \frac{1}{2}\right)^2 - \frac{1}{4} = \left[-\frac{1}{4}, 2\right]
 \end{aligned}
 \tag{4-20}$$

Nous pouvons constater que plus une variable apparaît dans une expression, plus la fonction d'extension naturelle sera pessimiste, en effet, chaque occurrence d'une variable est traitée comme une nouvelle variable indépendante. Nous pourrions donc l'améliorer en manipulant son expression formelle.

Voici quelques exemples de fonctions d'inclusions usuelles :

$$\begin{aligned}
 \sqrt{[x]} &= \left[\sqrt{x^-}, \sqrt{x^+} \right], \text{ si } x^- \geq 0 \\
 \exp([x]) &= [\exp(x^-), \exp(x^+)] \\
 \tan([x]) &= [\tan(x^-), \tan(x^+)] \text{ si } [x] \subseteq [-\pi/2, \pi/2]
 \end{aligned}
 \tag{4-21}$$

4.2.3 L'algorithme SIVIA

L'algorithme SIVIA (*Set Inversion Via Interval Analysis*) a été développé par Luc Jaulin et Eric Walter [JAU94] [JAU97]. Il permet de caractériser de façon approchée mais garantie, par l'intermédiaire de sous pavages, la solution d'un problème d'inversion ensembliste.

L'algorithme consiste, en partant d'un pavé initial, à effectuer un découpage de ce pavé en deux sous pavés. Ils sont ensuite tous deux examinés pour savoir s'ils peuvent être gardés ou rejetés. Si un pavé n'est pas valable, il est supprimé. Dans le cas contraire il est divisé de nouveaux en deux jusqu'à obtenir un pavé d'une taille inférieure à la précision souhaitée.

On cherche donc à caractériser l'ensemble solution χ de γ par la fonction réciproque f^{-1} , ce qui donne $\chi = f^{-1}(\gamma)$. Cet ensemble va être encadré par deux sous pavages χ^- et χ^+ , tel que $\chi^- \subset \chi \subset \chi^+$ (Figure 4-3).

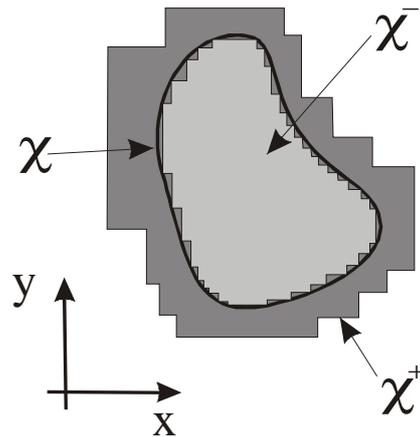


Figure 4-3 : Encadrement de l'ensemble solution du problème d'inversion ensembliste.

Nous dirons que si $[X] \subset \chi$ le pavé est acceptable, et que si $[X] \cap \chi = \emptyset$ le pavé est inacceptable, sinon la pavé est ambigu. L'analyse par intervalle définit deux conditions permettant de décider de l'acceptabilité d'un pavé :

- Si $f^1([X]) \subset \gamma$ Alors $[X] \subset \chi$, $[X]$ est acceptable
- Si $f^1([X]) \cap \gamma = \emptyset$ Alors $[X] \cap \chi = \emptyset$ $[X]$ est inacceptable
- Sinon le pavé est indéterminé.

Tout pavé indéterminé n'est pas nécessairement ambigu, mais peut être acceptable ou inacceptable. Par contre, tout pavé ambigu est nécessairement indéterminé (Figure 4-4).

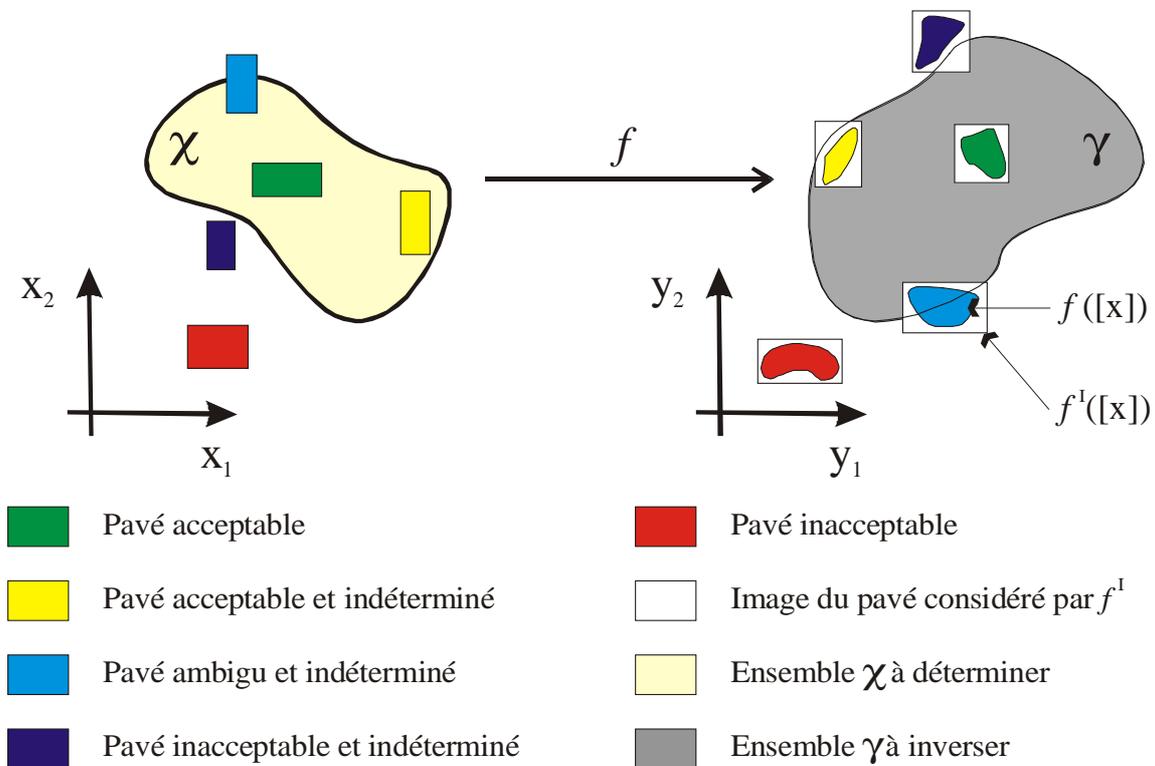


Figure 4-4 : Conditions d'acceptabilité d'un pavé.

En appliquant cet algorithme, nous obtenons trois sous pavages χ^- , χ^{ind} et χ^{ext} .

- χ^- est l'ensemble des pavés inclus dans χ .
- χ^{ind} est l'ensemble des pavés indéterminés.
- χ^{ext} est l'ensemble des pavés pour lesquels l'intersection avec l'ensemble solution est vide.

avec :

$$\chi^- \subset \chi \subset (\chi^{ind} \cup \chi^-), \quad [4-22]$$

Nous avons donc les équivalences suivantes :

- $\chi^{ext} = \overline{\chi^+}$.
 - $\chi^{ind} \cup \chi^- = \chi^+$.
- [4-23]

Le domaine de recherche initial est fixé suffisamment grand pour contenir de façon garantie l'ensemble χ . En outre, nous fixons une taille minimale ε qui représente la précision fixée en dessous de laquelle un pavé indéterminé n'est plus découpé.

Ainsi, tout pavé qui est indéterminé est de nouveau découpé, jusqu'à ce que sa taille devienne inférieure à ε , ce qui améliore la précision du sous pavage.

Dans la suite de notre travail, l'ensemble à caractériser est défini par un test booléen, qui prend la valeur vrai (1) si le pavé est acceptable, la valeur faux (0) si le pavé est inacceptable et la valeur indéterminé ([0,1]) si aucune de ces conditions n'est vérifiée. Ainsi nous définissons χ comme :

$$\chi = \{x \in \mathbb{R}^n \mid t(x) = 1\} \quad [4-24]$$

La fonction t est une fonction permettant de calculer simplement si un ensemble de paramètres sont compatibles entre eux. Nous pourrions donc redéfinir χ comme $\chi = t^{-1}(1)$.

Nous présentons ci-après l'algorithme de SIVIA récursif, utilisant les tests booléens [JAU94] :

```

SIVIA()
Entrées :
  t1 : fonction d'inclusion du test booléen
  [X] : Pavé à traiter ([X] = [X]0 lors de l'appel de SIVIA)
  ε : Précision du pavage
  χ- : Ensemble des pavés acceptables (χ- = ∅ au début)
  χind : Ensemble des pavés ambigus (χind = ∅ au début)
  χext : Ensemble des pavés inacceptables (χext = ∅ au début)

Sorties :
  χ- : Ensemble des pavés acceptables
  χind : Ensemble des pavés ambigus
  χext : Ensemble des pavés inacceptables

Si w([X]) < ε alors
  χind = χind + [X]
    
```

```

Sortir
finsi
Si  $t^I([X]) = \text{vrai}$  alors
   $\chi^- = \chi^- + [X]$ 
  Sortir
finsi
Si  $t^I([X]) = \text{faux}$  alors
   $\chi^{\text{ext}} = \chi^{\text{ext}} + [X]$ 
  Sortir
finsi

( $[X]_1, [X]_2$ ) = Découpage( $[X]$ )
SIVIA( $t^I, [X]_1, \varepsilon, \chi^-, \chi^{\text{ind}}$ )
SIVIA( $t^I, [X]_2, \varepsilon, \chi^-, \chi^{\text{ind}}$ )

fin SIVIA

```

Cet algorithme SIVIA fait appel à une fonction `Découpage()` permettant de récupérer deux pavés à partir d'un seul. En effet, comme nous l'avons vu précédemment, un pavé indéterminé est découpé en deux sous pavés pour être de nouveau examiné par l'algorithme. Cette fonction découpe le pavé le long du plus grand plan de symétrie. Il peut donc être utilisé quelque soit la dimension n du pavé (c'est à dire le nombre d'intervalles composants le pavé), à partir du moment où le facteur d'échelle entre les différentes dimensions du pavage est respecté.

L'utilisation de SIVIA et de la théorie ensembliste dans un programme nécessite un certain nombre de nouvelles structures et de types, comme par exemple le type booléen étendu pouvant prendre la valeur vrai, faux et indéterminé. Toutefois, nous ne détaillerons pas plus l'ensemble de ce travail, qui malgré sa complexité, sort du cadre de la problématique de cette thèse. Pour plus de détails, on pourra se référer à [KIE99]. Le programme SIVIA utilisé, ainsi que la majorité des structures de données sont disponible sur le site de L. Jaulin¹.

Nous pouvons remarquer par rapport à la description de SIVIA qu'une fonction $t^I()$ est utilisée, qui est la fonction d'inclusion du test booléen que nous avons introduit précédemment. Utiliser une fonction booléenne n'enlève rien au fait qu'il faut également l'approcher au mieux par une fonction d'inclusion.

Pour clôturer cette synthèse, nous allons présenter un exemple d'approximation par SIVIA d'une inéquation qui est un anneau définie comme $1 \leq \sqrt{x^2 + y^2} \leq 2$. Nous avons utilisé comme initialisation un pavé initial $[X]_0 = \left(\begin{array}{c} [-3,3] \\ [-3,3] \end{array} \right)$ et fixé la taille minimum d'un intervalle à $\varepsilon = 0,02$.

Nous pouvons retrouver sur la Figure 4-5, l'ensemble des trois sous pavages :

- En blanc : les pavés inacceptables,
- En vert : les pavés acceptables,
- En rouge : Les pavés indéterminés.

¹ <http://www.istia.univ-angers.fr/~jaulin/>

Nous pouvons constater également que lorsqu'un pavé n'est pas indéterminé, il est classé automatiquement comme acceptable ou inacceptable et n'est pas découpé : ceci se traduit par différentes tailles de pavés.

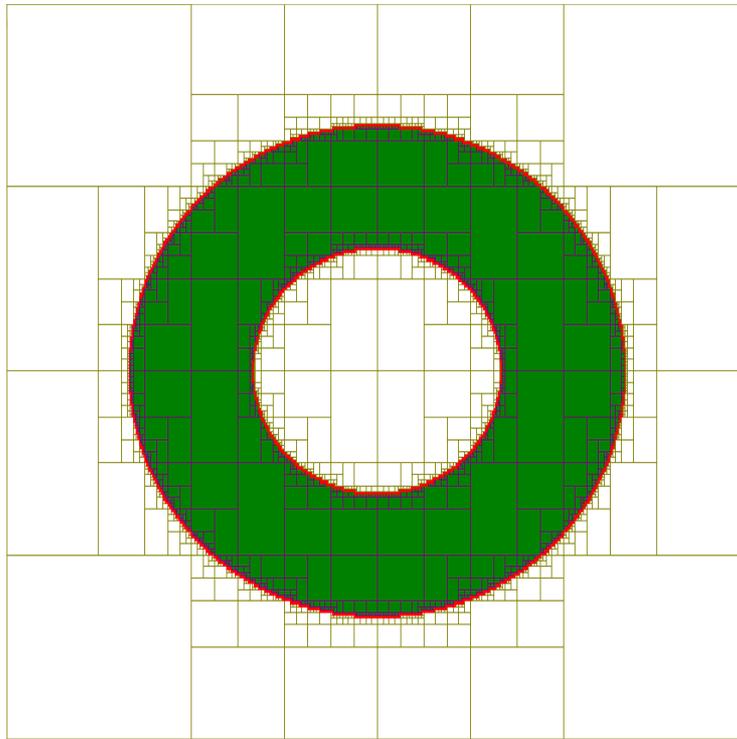


Figure 4-5 : Exemple de fonctionnement de SIVIA sur la résolution d'une inéquation.

4.3. Localisation d'un robot mobile par l'utilisation de SIVIA

4.3.1 Définition du problème

Nous avons vu précédemment que l'utilisation de SIVIA concernait les problèmes où, en connaissant l'image d'un ensemble par une fonction, nous avons à caractériser la source. La localisation d'un robot mobile entre tout à fait dans ce cadre. En effet, à partir d'un ensemble de mesures (d_1, \dots, d_n) , connaissant la méthode permettant de les obtenir à partir d'une position donnée, Nous cherchons à calculer la configuration du robot (x_r, y_r, θ_r) :

$$\begin{pmatrix} x_r \\ y_r \\ \theta_r \end{pmatrix} \xrightarrow{f} \begin{pmatrix} d_1 \\ \dots \\ d_n \end{pmatrix} \quad [4-25]$$

La résolution de ce problème consiste alors à caractériser l'image contenant l'ensemble des mesures par la fonction réciproque f^{-1} .

La localisation d'un robot mobile par la théorie ensembliste à bien sûr déjà été réalisée, notamment avec des capteurs télémétriques [LEV98] [KIE00]. Le robot est alors placé dans

un environnement connu, et l'ensemble de ses acquisitions télémétriques fournissent des intervalles de mesures, qui sont utilisés par l'algorithme SIVIA pour déterminer un pavage représentant la position du robot.

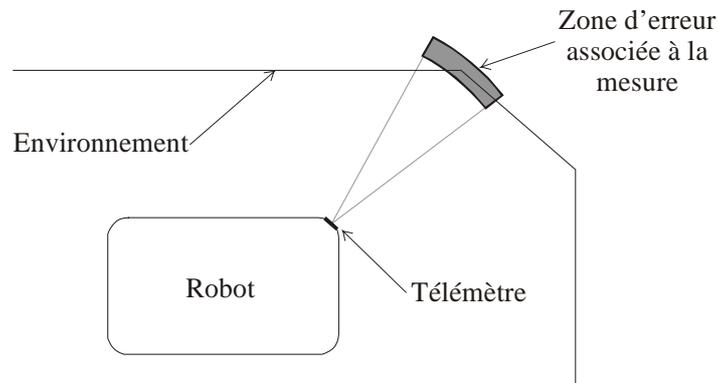


Figure 4-6 : Cône d'émission d'un capteur télémétrique.

Dans ces méthodes, le point prépondérant est celui de la mise en correspondance des données issues du capteur avec un amers de l'environnement. Un capteur télémétrique est caractérisé par un cône d'émission, représentant la zone possible dans laquelle un amers a pu être détecté (Figure 4-6). Le problème consiste alors à rechercher, pour un pavé donné qui représente la position du robot, l'ensemble des capteurs compatibles avec cette mesure. Cette étape revient en fait à caractériser la fonction booléenne t présentée précédemment, ou plus précisément sa fonction d'inclusion t^1 .

Notre problème diffère légèrement de cette approche. En effet, notre capteur fonctionne comme un goniomètre, c'est à dire que les données sensorielles représentent les angles d'observations des amers verticaux de l'environnement; elles ne peuvent donc pas être mises en correspondance avec d'autres éléments de la carte (comme les amers horizontaux). Ce constat est un avantage car il diminue nécessairement le nombre de possibilités d'appariement.

La première étape permettant de mettre en place notre algorithme de localisation par inversion ensembliste est de déterminer un modèle d'erreur lié aux données issues de notre capteur omnidirectionnel. Nous présentons dans le paragraphe suivant deux modèles d'erreur compatibles avec nos données sensorielles.

4.3.2 Caractérisation du modèle d'erreur

4.3.2.1. Premier modèle d'erreur pour la localisation.

La première idée que nous avons testée est intuitive et elle est basée sur l'utilisation des deux angles d'observations d'un amers, fournis par les deux images. Nous avons déjà utilisé ce paramètre dans le chapitre précédent comme critère de fusion : il nous a paru légitime de l'utiliser dans ce cadre.

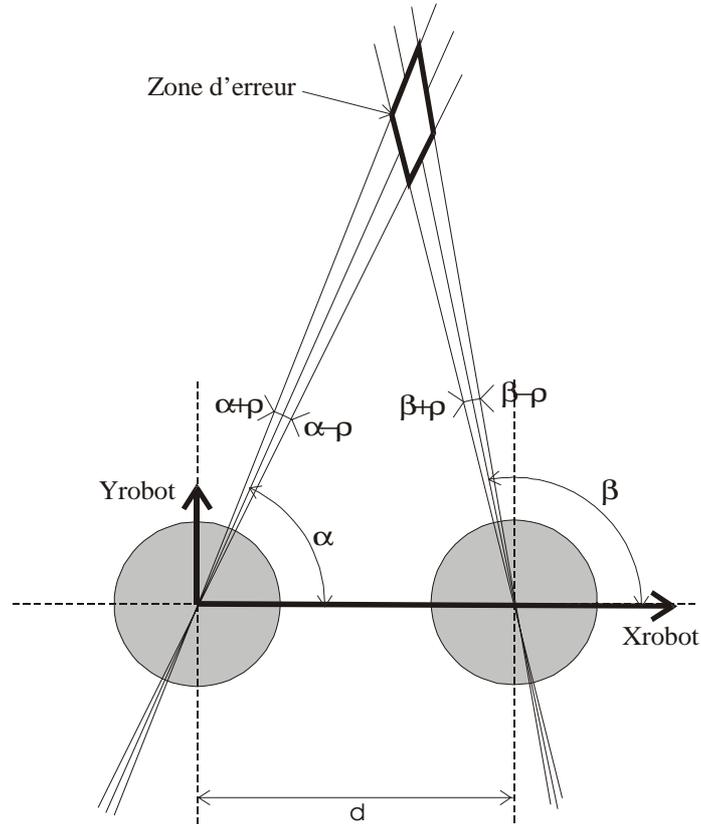


Figure 4-7 : Première approche permettant de caractériser l'erreur sur les données sensorielles.

Nous considérons à ce niveau que chaque angle d'observation d'un amer est entaché d'une erreur maximale ρ (Figure 4-7).

Si (x_r, y_r, θ_r) représente la position du robot et d la distance séparant les deux cônes, le calcul des coordonnées d'un point i de la carte dans le repère du robot est le suivant :

$$\begin{cases} x''_{si} = \frac{d \times \tan(\beta_i)}{\tan(\beta_i) - \tan(\alpha_i)} \\ y''_{si} = \frac{d \times \tan(\beta_i) \times \tan(\alpha_i)}{\tan(\beta_i) - \tan(\alpha_i)} \end{cases} \quad [4-26]$$

Pour se placer dans le repère du monde, il faut d'abord appliquer une rotation dans le repère du robot égale à l'orientation θ_r du robot :

$$\begin{pmatrix} x'_{si} \\ y'_{si} \end{pmatrix} = \begin{pmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{pmatrix} \times \begin{pmatrix} x''_{si} \\ y''_{si} \end{pmatrix} \quad [4-27]$$

Puis un changement de repère du robot vers le repère du monde :

$$\begin{cases} x_{si} = x'_{si} + x_r \\ y_{si} = y'_{si} + y_r \end{cases} \quad [4-28]$$

Soit :

$$\begin{pmatrix} x_{si} \\ y_{si} \end{pmatrix} = \begin{pmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{pmatrix} \times \begin{pmatrix} \frac{d \times \tan(\beta_i)}{\tan(\beta_i) - \tan(\alpha_i)} \\ \frac{d \times \tan(\beta_i) \times \tan(\alpha_i)}{\tan(\beta_i) - \tan(\alpha_i)} \end{pmatrix} + \begin{pmatrix} x_r \\ y_r \end{pmatrix} \quad [4-29]$$

Pour introduire le calcul par intervalle, nous posons $[\alpha_i] = [\alpha_i - \rho, \alpha_i + \rho]$ et $[\beta_i] = [\beta_i - \rho, \beta_i + \rho]$ et nous utilisons les fonctions d'inclusion $+$, $-$, \times , \div , $\cos()$, $\sin()$ et $\tan()$ relatives aux intervalles, ce qui nous permet d'obtenir la fonction d'inclusion suivante :

$$\begin{aligned} [S_i] &= \begin{pmatrix} [x_{si}] \\ [y_{si}] \end{pmatrix} \\ &= \begin{pmatrix} \cos([\theta_r]) & -\sin([\theta_r]) \\ \sin([\theta_r]) & \cos([\theta_r]) \end{pmatrix} \times \begin{pmatrix} \frac{d \times \tan([\beta_i])}{\tan([\beta_i]) - \tan([\alpha_i])} \\ \frac{d \times \tan([\beta_i]) \times \tan([\alpha_i])}{\tan([\beta_i]) - \tan([\alpha_i])} \end{pmatrix} + \begin{pmatrix} [x_r] \\ [y_r] \end{pmatrix} \\ &= f'([\alpha_i], [\beta_i], [x_r], [y_r], [\theta_r]) \end{aligned} \quad [4-30]$$

Où $[x_r]$, $[y_r]$ et $[\theta_r]$ représentent les intervalles respectifs de x_r , y_r et θ_r utilisés par SIVIA pour déterminer la position du robot et $[S_i]$ représente le pavé associé à la $i^{\text{ème}}$ mesure. C'est cette fonction d'inclusion qui nous permettra d'utiliser l'algorithme SIVIA.

4.3.2.2. Deuxième modèle d'erreur pour la localisation.

Partant de la constatation que d'un point de vue paramétrique, les données que nous gérons sont de même nature que des données télémétriques, nous avons extrapolé le modèle d'erreur utilisé par M. Kieffer à notre problème. Il est caractérisé par une erreur en distance et une erreur en angle (Figure 4-6). Nous avons testé cette modélisation pour la comparer avec notre première méthode.

Nous considérons à ce niveau que notre capteur nous fournit les positions des amers verticaux de l'environnement entachés d'une erreur en angle et d'une erreur en distance. Ceci forme un cône d'émission, semblable à celui obtenu avec un capteur télémétrique. La pointe de ce cône est située au milieu des deux images, et sur l'axe passant par leur centre. Connaissant les deux angles α et β , nous avons les coordonnées de l'amer, ce qui nous permet de calculer φ , l'angle d'observation de l'amer et l la distance mesurée (Figure 4-8).

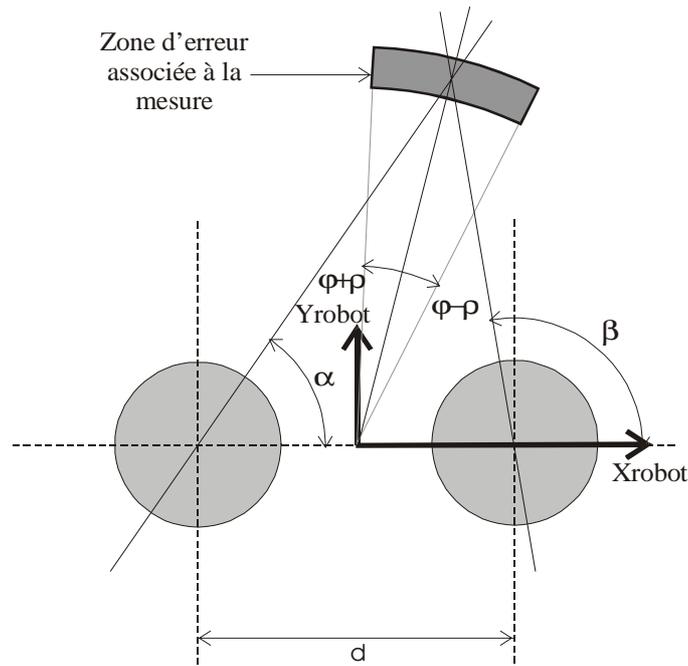


Figure 4-8 : Deuxième approche pour la modélisation de l'erreur.

Si (x_r, y_r, θ_r) représente la position du robot, l la distance mesurée et φ l'angle mesuré alors le calcul des coordonnées d'un point i de la carte se calcule avec les formules suivantes :

$$\begin{cases} x''_{si} = l_i \times \cos(\varphi_i) \\ y''_{si} = l_i \times \sin(\varphi_i) \end{cases} \quad [4-31]$$

Nous appliquons ensuite une rotation dans le repère du robot égale à l'orientation θ_r du robot (comme précédemment) :

$$\begin{pmatrix} x'_{si} \\ y'_{si} \end{pmatrix} = \begin{pmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{pmatrix} \times \begin{pmatrix} x''_{si} \\ y''_{si} \end{pmatrix} \quad [4-32]$$

Puis un changement de repère du robot vers le repère du monde :

$$\begin{cases} x_{si} = x'_{si} + x_r \\ y_{si} = y'_{si} + y_r \end{cases} \quad [4-33]$$

Soit :

$$\begin{pmatrix} x_{si} \\ y_{si} \end{pmatrix} = \begin{pmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{pmatrix} \times \begin{pmatrix} l_i \times \cos(\varphi_i) \\ l_i \times \sin(\varphi_i) \end{pmatrix} + \begin{pmatrix} x_r \\ y_r \end{pmatrix} \quad [4-34]$$

En posant $[l_i] = [l_i - \varepsilon, l_i + \varepsilon]$ et $[\varphi_i] = [\varphi_i - \rho, \varphi_i + \rho]$ et en utilisant les fonctions d'inclusions $+$, $-$, \times , \div , $\cos()$ et $\sin()$ relatives aux intervalles, nous obtenons la fonction d'inclusion suivante :

$$\begin{aligned}
[S_i] &= \begin{pmatrix} [x_{si}] \\ [y_{si}] \end{pmatrix} \\
&= \begin{pmatrix} \cos([\theta_r]) & -\sin([\theta_r]) \\ \sin([\theta_r]) & \cos([\theta_r]) \end{pmatrix} \times \begin{pmatrix} [l_i] \times \cos([\varphi_i]) \\ [l_i] \times \sin([\varphi_i]) \end{pmatrix} + \begin{pmatrix} [x_r] \\ [y_r] \end{pmatrix} \\
&= f^i([l_i], [\varphi_i], [x_r], [y_r], [\theta_r])
\end{aligned} \tag{4-35}$$

De la même façon qu'avec le modèle d'erreur précédent, c'est cette fonction d'inclusion qui sera utilisée avec l'algorithme SIVIA.

4.3.3 Algorithme de localisation

Nous présentons dans ce paragraphe l'algorithme "au sens ensembliste" de mise en correspondance des données sensorielles avec les données de la carte, qui est basé sur la comparaison de chaque observation avec chaque point de la carte.

Tout d'abord, une fois déterminé le pavé $[S_i]$ correspondant à une donnée sensorielle par l'une des deux méthodes précédentes, il faut tester si un des points de la carte se trouve dans ce pavé, l'algorithme de cette étape est donc le suivant :

```

Fonction test_validite_pave([Si])
valide=faux
Pour chaque point Pj de la carte
  Si Pj ∈ [Si] alors valide=vrai
fin pour
renvoyer valide

```

Avec P_j représentant le point j de la carte et $[S_i]$ représentant la pavé associé à la $i^{\text{ème}}$ donnée sensorielle.

Finalement, pour obtenir notre fonction d'inclusion booléenne, nous permettant d'obtenir un test global de validité, nous appliquons cet algorithme à l'ensemble des données sensorielles :

```

Fonction test_position_robot([xr],[yr],[θr])
valide=vrai
Pour chaque données sensorielle i
  Calculer le pavé [Si]=fi([xr],[yr],[θr],...)
  Si test_validite_pave([Si])=faux alors
    valide=faux
  Fin si
fin pour
renvoyer valide

```

Cet algorithme est illustré par la Figure 4-9, où nous pouvons constater que chaque observation est bien issue d'une donnée réelle existant dans la carte de l'environnement.

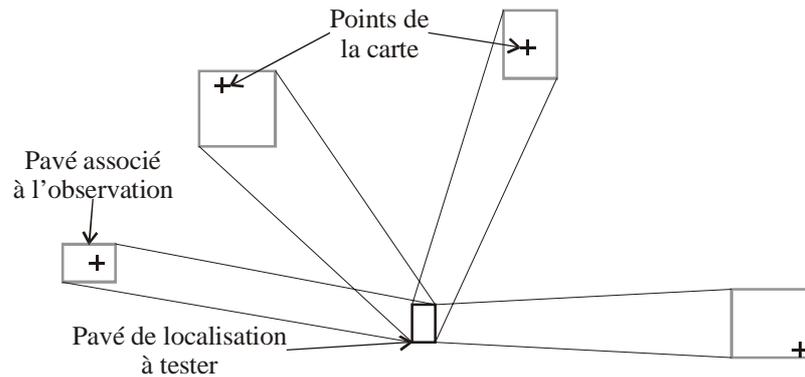


Figure 4-9 : Illustration du fonctionnement de l'algorithme de localisation.

Cette situation représente uniquement le cas où il n'y a pas de données aberrantes. En effet, l'algorithme teste tous les pavés associés aux données sensorielles un par un, et si un seul n'est pas valide, la position du robot ne l'est pas non plus. Il est bien évident que cette situation pose de nombreux problèmes car il est courant d'avoir plusieurs données aberrantes par acquisition.

La solution que nous avons choisie pour traiter ce problème est la même que celle adoptée par M. Kieffer [KIE99], elle consiste à appliquer l'algorithme en considérant qu'il n'y a pas de données aberrantes : si aucune solution n'est trouvée, nous réitérons l'algorithme avec une donnée aberrante, puis deux, et ainsi de suite. Cette solution nous permet d'obtenir un résultat quelque soit le ratio "données aberrantes/données valides".

Nous présentons ici l'algorithme modifié prenant en compte les données aberrantes, le nombre de données aberrantes est représenté par la variable **dab** :

```

Fonction test_position_robot2([xr],[yr],[θr],dab)
non_valide=0
Pour chaque donnée sensorielle i
    Calculer le pavé [Si]=f-1([xr],[yr],[θr],...)
    Si test_validite_pave([Si])=faux alors
        non_valide=non_valide+1
    fin si
fin pour
Si non_valide ≤ dab alors
    renvoyer vrai
fin si
renvoyer faux

```

Par rapport au test de validité d'une donnée sensorielle, une autre solution aurait été de calculer les données sensorielles théoriques à partir de la position du robot, puis de comparer ces données avec celles issues du capteur. L'avantage de cette alternative est de permettre d'obtenir un pavage intérieur et extérieur de la position du robot. Nous n'avons pas retenu cette solution car dans le cadre de son utilisation pour la modélisation de l'environnement, le calcul d'une donnée théorique sera basé non plus sur un point de la carte connu à l'avance, mais sur un ensemble de pavés représentant l'ensemble des positions possibles de l'amer. Cette alternative serait donc non généralisable à la suite de notre étude. Tout ceci sera détaillé dans le prochain chapitre.

Une conséquence immédiate résultant de ce choix est que les pavés seront toujours découpés jusqu'à la taille minimale définie par l'erreur. Nous aurons alors uniquement

l'approximation extérieure de la position du robot, il s'agit néanmoins de l'unique donnée qui nous intéresse pour les traitements ultérieurs que nous aurons à mettre en œuvre. La Figure 4-10 présente un exemple de représentation tridimensionnelle du pavage extérieur d'une localisation du robot effectuée avec la méthode précédente.

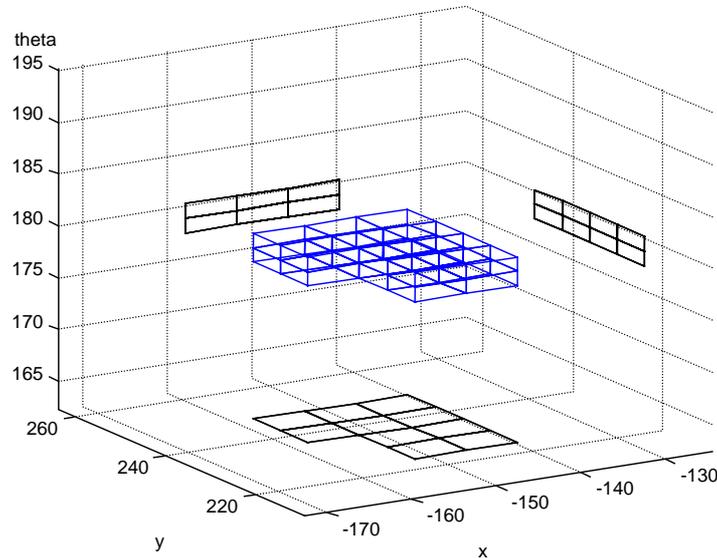


Figure 4-10 : Exemple de pavage extérieur de la position du robot.

4.3.4 Comment utiliser l'information odométrique ?

Dans ce paragraphe, nous étudions la possibilité d'utiliser l'information odométrique tout comme nous l'avons fait avec la méthode proposée dans le Chapitre 3, mais cette fois ci, en utilisant un formalisme "ensembliste",.

Lors de l'utilisation de SIVIA, la première étape est la recherche de solution à partir d'un pavé reçu en argument, qui doit contenir la position réelle du robot. Une première solution est d'initialiser ce pavé avec les dimensions de l'environnement. Le problème est que d'une part le temps de traitement s'en trouve augmenté, d'autre part dans des environnements relativement symétriques, la solution peut ne pas être unique, et même ne pas contenir la position réelle du robot.

En partant de ces constatations, nous avons décidé d'utiliser l'information odométrique pour affiner la recherche de solutions. Une première méthode pour utiliser cette information est de se baser sur le même principe que le filtrage de Kalman, en utilisant des phases successives de prédictions/corrections [KIE98]. Un état est prédit à partir de l'état à l'instant t et de l'information odométrique, un second état compatible avec les mesures est généré et l'intersection de ces deux ensembles donne l'état à l'instant $t+1$ (Figure 4-11).

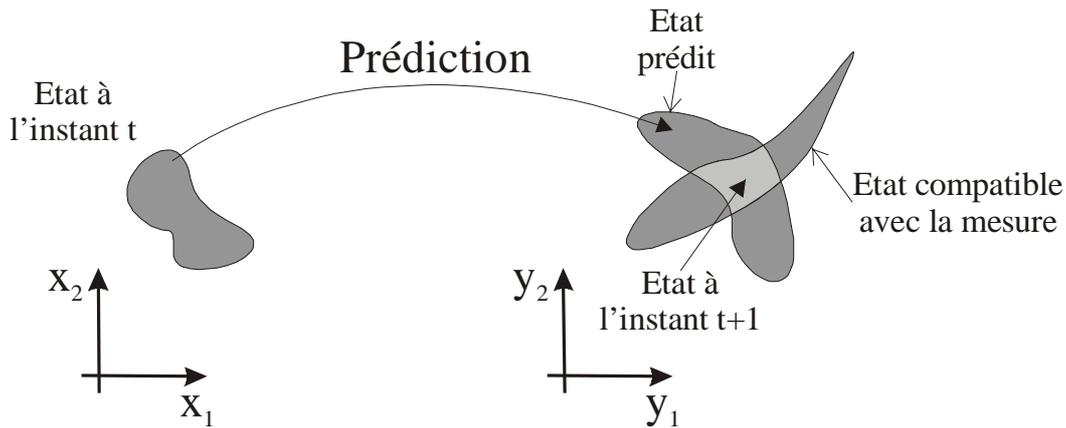


Figure 4-11 : Utilisation de l'odométrie par estimation d'état.

Cette méthode fonctionne mais nécessite des algorithmes spécifiques utilisant des sous pavages et des arbres binaires pour calculer l'état prédit. De plus, des versions modifiées de SIVIA doivent être utilisées pour tenir compte de ces particularités. Pour plus de détails le lecteur pourra se référer à [KIE99].

L'utilisation d'une phase de prédiction aussi précise que possible est très utile quand le nombre de pavés est relativement important, comme dans le cas de l'utilisation de capteurs télémétriques. Dans notre cas, les données sensorielles représentent les amers verticaux de l'environnement, par conséquent l'imprécision sera plus faible et le nombre de pavés relativement peu élevé (comme nous pourrons le constater dans les résultats expérimentaux).

Nous avons donc décidé d'utiliser l'odométrie uniquement pour initialiser le pavé initial P_0 qui est utilisé pour débiter la recherche des positions réelles du robot. A partir d'un sous pavage résultant d'une localisation, nous calculons le pavé englobant minimal, puis, ce pavé est augmenté de l'erreur odométrique maximum, fonction de la distance parcourue (Figure 4-12).

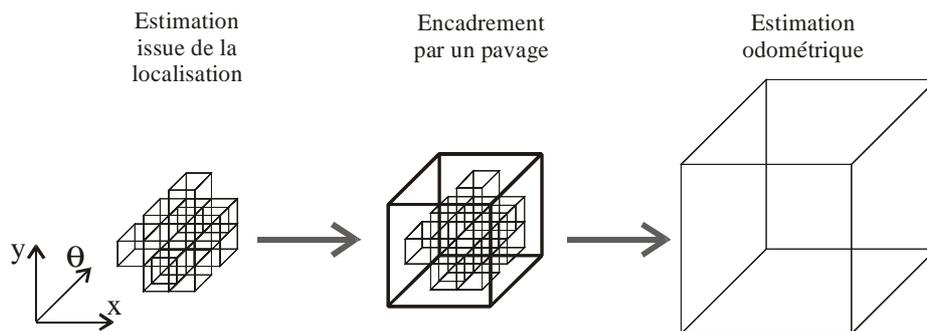


Figure 4-12 : Méthode odométrique utilisée.

Contrairement à la méthode de fusion des informations odométriques avec les données observées, dans notre cas aucun pavé résultant de la phase de localisation n'est supprimé de l'ensemble solution. Cette méthode est uniquement une phase d'initialisation, et comme nous majorons l'erreur odométrique, elle permet simplement à la phase de localisation de ne pas tester les éventuels résultats incompatibles avec la position actuelle du robot.

4.4. Résultats expérimentaux

Nous présentons dans ce paragraphe l'ensemble des résultats expérimentaux obtenus avec notre méthode de localisation. Nous avons utilisé les mêmes séries d'acquisitions que dans les chapitres précédents, et nous avons bien sûr comparé les deux méthodes de modélisation de l'erreur associée à une donnée issue du capteur. Nous discuterons les résultats obtenus avec ces deux méthodes et nous présenterons le choix que nous avons effectué pour la suite.

4.4.1 Résultats expérimentaux utilisant le premier modèle d'erreur

Nous nous servons ici du domaine d'erreur obtenu avec les deux angles d'azimut. Les résultats obtenus dans le premier environnement type hall sont présentés sur la Figure 4-13 pour les huit acquisitions dont sept localisations. Nous pouvons constater que seuls deux pavages n'englobent pas la position réelle du robot, toutefois, la distance séparant ces positions au pavé le plus proche n'est que de 6 cm pour la première et 2 cm pour la seconde. La représentation de la configuration du robot est ici bidimensionnelle (abscisse et ordonnée), en effet, la représentation de l'information angulaire n'est pas facilement réalisable sur cette carte, de plus l'intervalle correspondant à l'angle estimé englobe systématiquement l'angle réel. Ceci est une caractéristique de notre capteur qui, fonctionnant comme un goniomètre, donne une bonne précision angulaire.

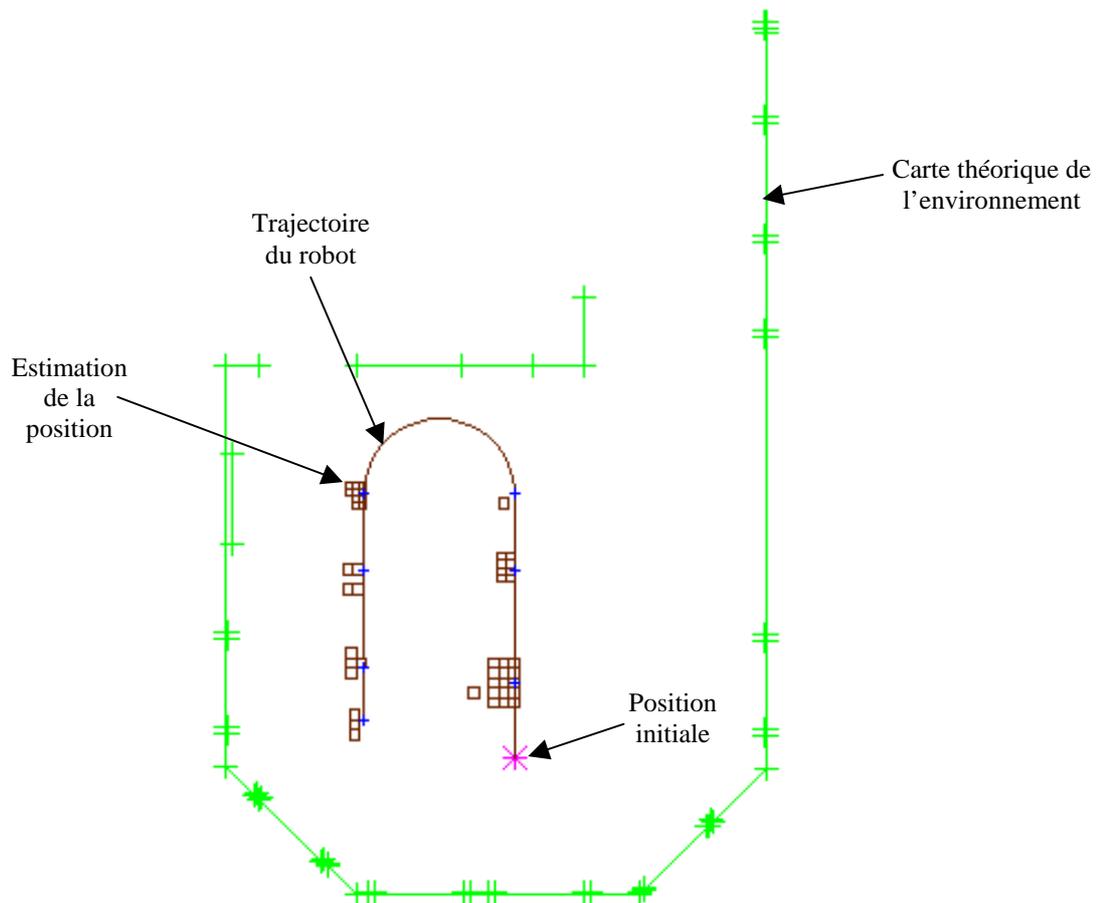


Figure 4-13 : Résultat de la localisation dans le premier environnement à l'aide de la 1^{ère} quantification d'erreur.

Nous pouvons constater que la localisation est robuste puisqu'une solution optimale est systématiquement trouvée. La quantification de l'erreur par le pavage est cohérente sur la quasi totalité des acquisitions.

La série d'acquisitions réalisée dans le deuxième environnement type couloir comporte 45 acquisitions soit 44 localisations. L'ensemble des estimations de position est donné en Figure 4-14. Etant donné qu'une acquisition est faite tous les trente centimètres, nous n'avons représenté sur cette carte qu'un résultat sur deux pour garder une certaine lisibilité, ce sont celles dont les positions réelles sont représentées par des croix roses. De plus, un pavage sur deux est de couleur bleu pour ne pas le confondre avec le suivant et le précédent. Nous pouvons constater que sur 44 localisations, 29 d'entre elles donnent un pavage contenant la position réelle du robot. Pour les 15 autres l'écart moyen entre la position du robot et le pavé le plus proche est inférieur à 7 centimètres. Nous pouvons considérer que ces écarts sont minimes, étant donné que la position réelle du robot a été déterminée avec une précision qui est de l'ordre de 5 cm et qui correspond à un relevé au mètre ruban, il en va de même pour l'ensemble des dimensions de l'environnement.

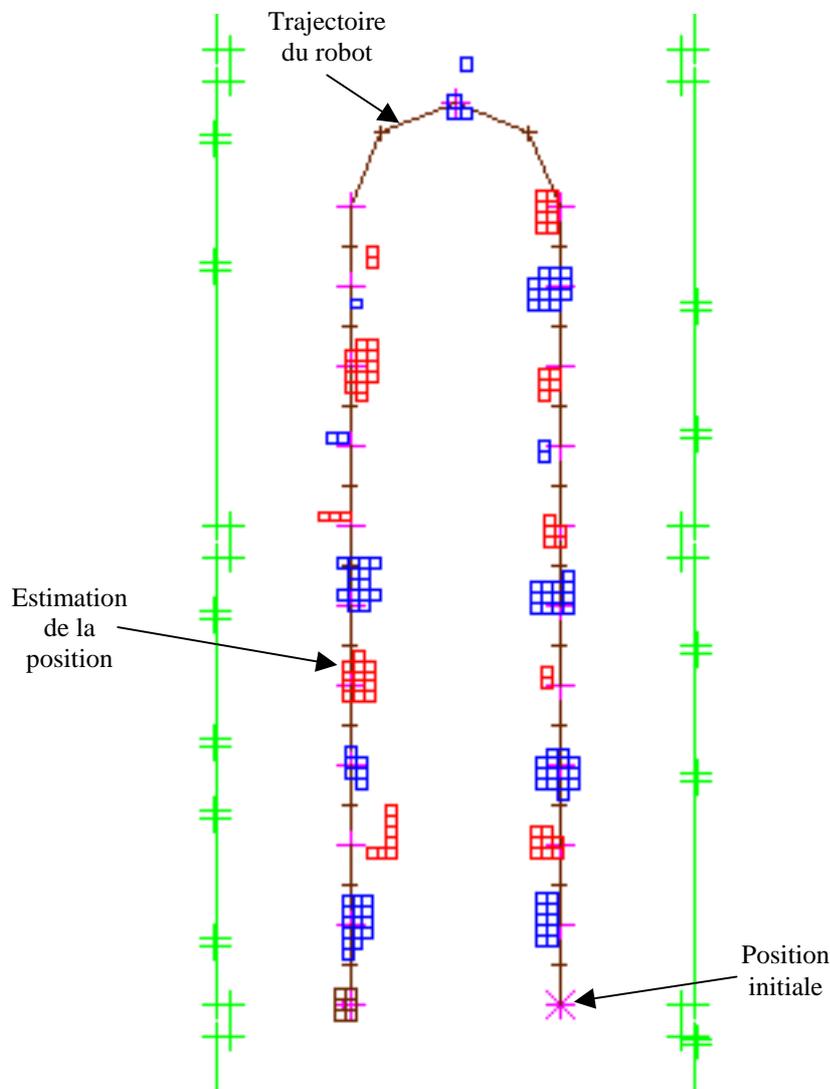


Figure 4-14 : Résultat de la localisation dans le deuxième environnement à l'aide de la 1^{ère} quantification d'erreur.

4.4.2 Résultats expérimentaux utilisant le second modèle d'erreur

Nous avons, comme précédemment, estimé la position du robot à partir des mêmes acquisitions mais en intégrant le deuxième modèle d'erreur au niveau SIVIA.

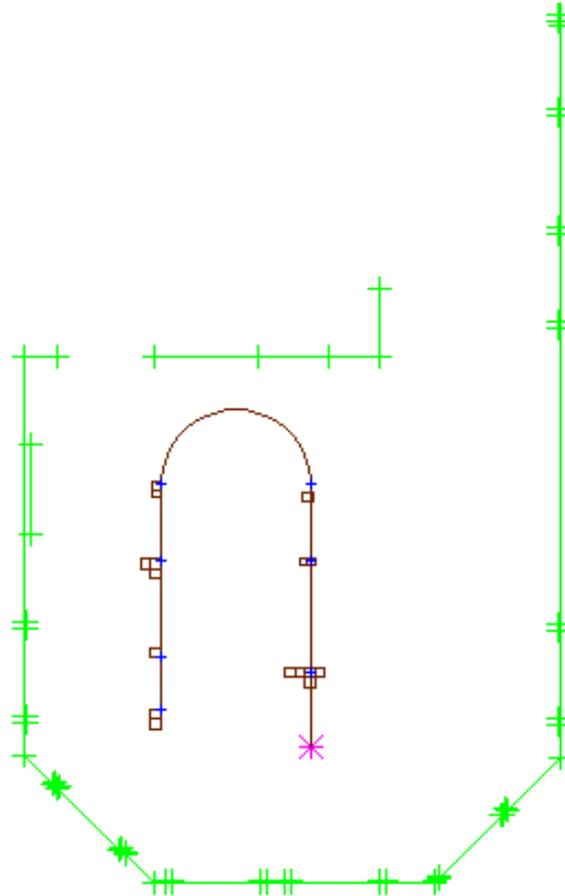


Figure 4-15 : Résultats de localisation dans le premier environnement à l'aide de la 2^{ème} quantification d'erreur.

Dans l'environnement de type hall, une seule localisation donne un pavage ne contenant pas la position réelle du robot. La position réelle du robot se trouvant alors à environ 7 cm du pavé le plus proche (Figure 4-15). Il est intéressant de remarquer que cette erreur était déjà apparue sur la même position avec la méthode précédente de quantification et ce avec un ordre de grandeur à peu près identique.

Nous pouvons remarquer également que le nombre de pavés qui approximent chaque position du robot est moins important qu'avec la précédente méthode tout en englobant la position réelle. Nous pouvons donc constater expérimentalement que l'estimation de l'erreur de localisation est plus optimale au sens de la minimalisation du domaine englobant la position qu'avec le premier modèle d'erreur. Ceci pourra donc être intéressant pour une modélisation ultérieure.

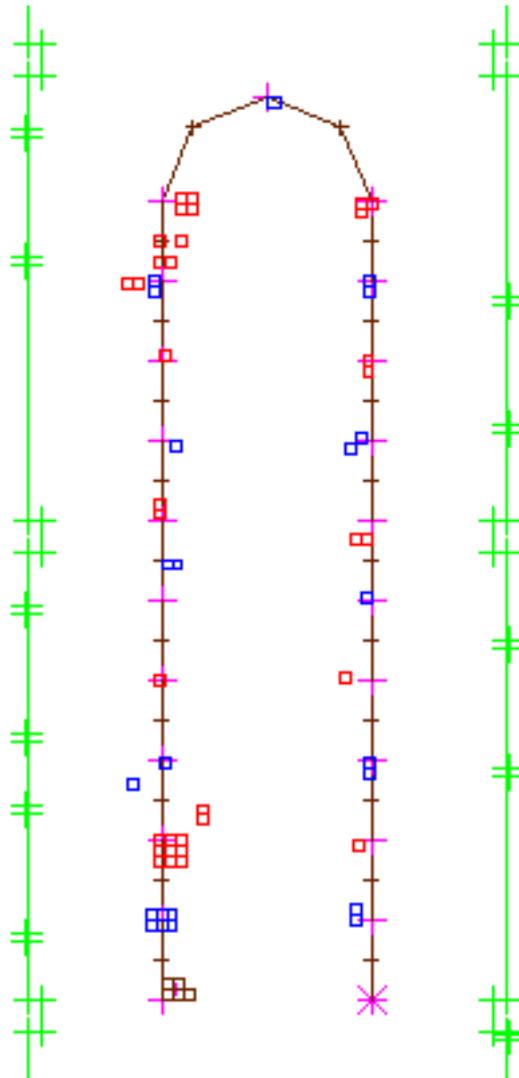


Figure 4-16 : Résultats de localisation dans le premier environnement à l'aide de la 2^{ème} quantification d'erreur.

Un récapitulatif de la série d'acquisitions réalisée dans le deuxième environnement est donné en Figure 4-16. Comme précédemment nous n'avons représenté qu'un résultat de localisation sur deux. Nous pouvons constater que sur 44 localisations, 25 d'entre elles donnent un pavage contenant la position réelle du robot. Pour les 19 autres l'écart moyen entre la position du robot et le pavé le plus proche est également inférieur à 7 centimètres. Nous pouvons également constater que la taille du pavage est plus petite avec cette deuxième méthode qu'avec la première, ce qui implique une meilleure précision.

4.4.3 Comparaison des deux modèles d'erreur

En analysant les résultats expérimentaux donnés précédemment, nous pouvons constater que les deux solutions sont similaires d'un point de vue estimation de position. Toutefois, la deuxième méthode de modélisation de l'erreur présente des avantages par rapport à la première :

- Le nombre de pavés résultant d'une localisation est inférieur dans la deuxième méthode, ce qui diminue également l'imprécision sur un amer observé lors d'une

modélisation. En effet, concernant les acquisitions réalisées dans le deuxième environnement, le nombre de pavés moyen pour chaque localisation est de l'ordre de 20 pour la première méthode contre 10 pour la seconde.

- En utilisant la première méthode, nous pouvons remarquer que plus l'angle d'observation d'un amer s'éloigne de la verticale du capteur (c'est-à-dire s'éloigne de 90 degrés et de -90 degrés), plus la taille du pavé associé à cet amer va augmenter jusqu'à devenir pour certaines configurations angulaires, peu réaliste et donc peu cohérente. Ceci compliquera obligatoirement la phase de modélisation.

En se basant sur ces constatations, nous avons porté notre choix sur la deuxième méthode de quantification de l'imprécision pour suivre notre étude. Nous travaillerons donc, pour l'étape de construction de carte, avec le modèle qui se rapproche le plus de celui adopté pour les données télémétriques. L'algorithme de localisation que nous avons présenté dans ce chapitre sera couplé à un algorithme de mise à jour de carte dans le chapitre suivant.

4.5. Conclusion

Nous avons dans ce chapitre rappelé brièvement les bases de l'analyse par intervalle, ainsi que le principe de l'algorithme SIVIA.

Nous avons appliqué cette théorie pour localiser notre robot, avec les données sensorielles omnidirectionnelles obtenues lors des acquisitions. Pour répondre au mieux à la problématique générale que nous nous sommes fixée, à savoir localiser et modéliser l'environnement de façon incrémentale, nous avons testé deux modèles d'erreur associés aux mesures : un premier basé sur les deux angles d'observation de l'amer, et un identique au modèle utilisé avec des données télémétriques et qui prend en compte une erreur en distance et une erreur en angle.

Les différents résultats expérimentaux obtenus dans deux environnements différents nous ont permis dans un premier temps de valider deux algorithmes de localisation, déclinés par rapport aux deux quantifications d'erreur utilisées.

En comparant les deux types de résultats obtenus, nous avons observé que le deuxième modèle d'erreur, similaire à un modèle d'erreur télémétrique, se révèle mieux adapté que le premier à une démarche de localisation et modélisation simultanées.

Nous présentons dans le chapitre suivant, une application de l'approche ensembliste au problème de localisation et modélisation incrémentale de l'environnement sans connaissance *a priori*. La quantification de l'imprécision qui entache l'estimation de position pourra être propagée et donc utilisée par rapport à l'estimation des paramètres caractérisant chacun des amers verticaux cartographiés.

Chapitre 5

Une approche ensembliste au problème de mise à jour incrémentale de la carte

Sommaire partiel

5.1.	INTRODUCTION	120
5.2.	REPRESENTATION DES AMERS DANS LA CARTE DE L'ENVIRONNEMENT	121
5.3.	ALGORITHME DE LOCALISATION MODIFIE.....	124
5.4.	METHODE DE DECISION POUR LA MISE EN CORRESPONDANCE	125
5.4.1	<i>Détermination de la croyance accordée à chaque association.....</i>	<i>125</i>
5.4.2	<i>Algorithme de décision</i>	<i>130</i>
5.5.	MISE A JOUR INCREMENTALE DE LA CARTE DE L'ENVIRONNEMENT	131
5.6.	RESULTATS EXPERIMENTAUX	133
5.6.1	<i>Tests en environnement structuré de type hall</i>	<i>133</i>
5.6.2	<i>Résultats en environnement de type couloir sur des trajets importants</i>	<i>136</i>
5.7.	CONCLUSION	142

5.1. Introduction

L'objectif de ce chapitre est de proposer une solution palliant au problème soulevé dans le chapitre 3, c'est-à-dire l'absence de prise en compte des erreurs commises d'une part sur l'estimation de position du robot et d'autre part sur l'estimation de position des balises. Le chapitre précédent a mis en évidence l'intérêt d'utiliser une approche ensembliste pour quantifier de façon cohérente l'imprécision sur la localisation. Nous allons appliquer dans ce chapitre la même approche pour quantifier l'imprécision sur l'estimation de position du robot au cours du déplacement.

En nous appuyant sur l'algorithme de localisation précédemment présenté, nous prendrons en compte dans ce chapitre un facteur prépondérant dans la modélisation de l'environnement : l'interaction entre les erreurs de localisation et celles liées à l'estimation de position des balises. Ainsi, le paradigme de localisation et de construction incrémentale de l'environnement se trouvera optimisé par rapport à celui présenté dans le chapitre 3. Des tests ont été ainsi réalisés sur les mêmes environnements et ont montré la robustesse de notre approche sur des distances importantes. En effet, nous montrerons que les phénomènes d'erreurs cumulatives, inhérents à l'absence de connaissance *a priori* ont pu être minimisés, voire supprimés dans certains cas.

Dans un premier temps nous présenterons l'algorithme de localisation, basé sur celui qui a été détaillé dans le précédent chapitre, mais modifié pour tenir compte des primitives utilisées (pavés) représentant les amers verticaux de l'environnement.

Ensuite nous exposerons la méthode de décision utilisée pour savoir si une observation doit être fusionnée avec un élément de la carte, ou si elle amène à la création d'un nouvel élément.

Puis nous verrons comment nous avons résolu le problème de la mise à jour incrémentale de la carte de l'environnement, nous exposerons notamment le principe de notre méthode basée sur les intersections de pavés.

Finalement nous clôturerons ce chapitre par des résultats expérimentaux réalisés dans les deux environnements de références utilisés dans le chapitre 3, et conclurons sur l'analyse de ces résultats.

5.2. Représentation des amers dans la carte de l'environnement

La représentation des données dans la carte est à la base du paradigme de localisation et modélisation simultanées. Nous avons vu dans le premier chapitre que les méthodes de modélisation peuvent être classées par rapport à la quantification choisie. Dans notre cas, il faut nous orienter vers une représentation d'amer compatible avec les informations ensemblistes, et facilement utilisable lors d'une mise à jour. Il semble à ce niveau que la seule solution possible soit une représentation par sous pavage.

Dans le chapitre précédent, nous avons testé deux modèles possibles concernant la représentation de l'erreur associée à chaque observation. La première est basée sur la fusion de deux erreurs en angles qui entachent les deux observations d'angle d'azimut. La deuxième est identique à celle utilisée sur les capteurs télémétriques, et correspond à la fusion d'une erreur en angle et d'une erreur en distance.

L'étude de ces deux représentations sur deux séries d'acquisitions a fait ressortir que la deuxième méthode donne une meilleure précision sur la localisation. C'est donc cette représentation qui sera utilisée dans notre paradigme de localisation et de modélisation de l'environnement.

Le résultat de l'étape de localisation étant un sous pavage $[[L]]$, nous pouvons calculer pour chacun des pavés $[L]_g$ le constituant, et pour chaque donnée sensorielle $[\varphi]_i$ et $[l]_i$, le pavé résultant $f'([L]_g, [\varphi]_i, [l]_i)$ grâce à la fonction d'inclusion déjà utilisée avec l'algorithme de localisation. Nous rappelons que l'équation du pavé associé à chaque mesure i est donnée par :

$$\begin{aligned}
 [S]_i &= \begin{pmatrix} [x_s]_i \\ [y_s]_i \end{pmatrix} \\
 &= \begin{pmatrix} \cos([\theta_r]) & -\sin([\theta_r]) \\ \sin([\theta_r]) & \cos([\theta_r]) \end{pmatrix} \times \begin{pmatrix} [l]_i \times \cos([\varphi]_i) \\ [l]_i \times \sin([\varphi]_i) \end{pmatrix} + \begin{pmatrix} [x_r] \\ [y_r] \end{pmatrix} \\
 &= f'([L]_g, [\varphi]_i, [x_r], [y_r], [\theta_r])
 \end{aligned} \tag{5-1}$$

Où $[x_r]$, $[y_r]$, $[\theta_r]$, $[\varphi]_i$ et $[l]_i$ représente les intervalles respectifs de x_r , y_r et θ_r la position du robot, φ_i l'angle d'observation de l'amer considéré par le centre du capteur et l_i la distance mesurée.

Si nous appliquons cette fonction d'inclusion à l'ensemble des pavés issus de l'étape de localisation, nous obtenons à la suite de ce traitement un ensemble de pavés correspondant à chaque mesure qui pourront avoir une intersection non vide entre eux et ne constitueront donc pas un sous pavage (Figure 5-1).

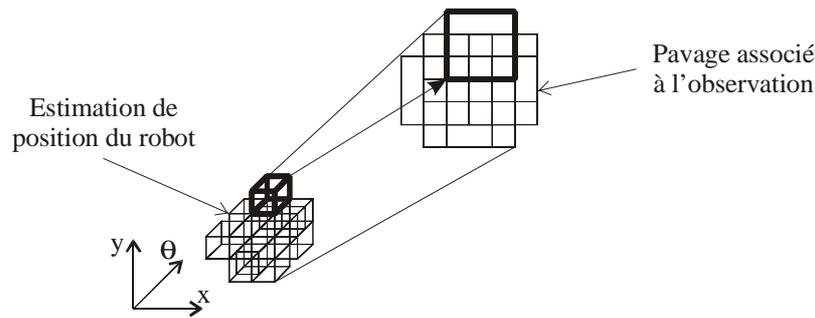


Figure 5-1 : Représentation de l'ensemble des pavés caractérisant une donnée sensorielle

Cette problématique a déjà été abordée par M. Kieffer dans [Kieffer 99]. En effet, pour calculer l'image d'un sous pavage, il a développé l'algorithme ImageSP qui se décompose en trois phases :

- **Hachage** : Calcule d'un sous pavage régulier $\llbracket A \rrbracket$ dont tous les pavés sont de taille inférieure à ϵ ,
- **Evaluation** : Calcule de l'image de chacun de ces pavés par la fonction d'inclusion f^I considérée,
- **Régularisation** : Approximation de l'union des pavés $f^I(\llbracket A \rrbracket)$ par un nouveau sous pavage $\llbracket B \rrbracket$.

Etant donné que le sous pavage obtenu lors de la phase de localisation du robot $\llbracket L \rrbracket$ est déjà constitué uniquement de pavés de taille inférieure à la précision souhaitée, la première étape n'est pas nécessaire. Nous pouvons directement calculer pour chacun de ces pavés, et pour chaque donnée sensorielle, le pavé résultant, grâce à la fonction d'inclusion précédente (Figure 5-1).

Enfin, la régularisation consiste à utiliser de nouveau l'algorithme SIVIA pour obtenir le sous pavage voulu. Cependant, cette représentation par sous pavage peut se décliner de différentes manières. La première approche consiste à représenter l'amer considéré par les deux ensembles habituellement utilisés dans la théorie ensembliste, c'est-à-dire l'approximation extérieure et intérieure paramétrant la primitive géométrique. Il nous faudra donc manipuler dans les traitements liés à la fusion et à l'intégration de données non seulement deux types différents de pavés, mais aussi des pavés de tailles différentes. En effet, l'utilisation de l'algorithme SIVIA, implique que lorsqu'un pavé convient il n'est plus bisecté, alors que le découpage continu s'il est incertain jusqu'à obtenir un pavé d'une taille égale ou inférieure à la précision fixée (Figure 5-2). Le problème de cette méthode est qu'elle nous oblige à appliquer de nouveau l'algorithme SIVIA sur l'ensemble des pavés certains lors de l'étape de fusion avec un autre ensemble. De plus, l'information de sous pavage extérieur et intérieur n'est pas utile dans notre cas puisque nous devons traiter l'ensemble des solutions possibles.

Une autre solution est d'utiliser un ensemble de pavés de tailles identiques, égales à la précision minimale fixée, qui caractérisent les deux ensembles précédents. L'avantage de cette représentation est qu'aucune bisection ne sera nécessaire lorsque nous aurons à traiter un ensemble de ce type. Les pavés seront soit acceptés, soit rejetés puisqu'ils sont tous de tailles inférieures à la précision souhaitée. Avec cette méthode, nous simplifions donc la représentation des données dans la carte, mais aussi les traitements qui seront appliqués dans les phases suivantes (Figure 5-2).

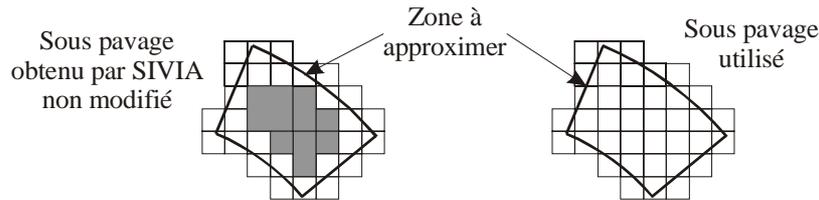


Figure 5-2 : Approximation d'un ensemble par les deux méthodes décrites précédemment

Il nous faut maintenant déterminer la fonction d'inclusion qui sera utilisée par l'algorithme SIVIA lors de l'ajout d'un nouvel amer dans la carte. Puisque nous voulons obtenir l'ensemble des pavés de tailles inférieures à la précision souhaitée, cette fonction ne devra renvoyer que deux valeurs : "faux" ou "indéterminé". En effet, une valeur "vrai" renvoyée par ce test d'inclusion stopperait immédiatement la bisection du pavé en cours.

Cette fonction d'inclusion a un rôle double :

- Elle sera utilisée pour initialiser la carte de l'environnement avec les données issues de la première acquisition,
- Elle sera aussi utilisée à chaque fois qu'un nouvel amer sera ajouté à la carte.

Ces deux cas de figure nous obligent à différencier deux utilisations de cette fonction d'inclusion. En effet, lors de la phase d'initialisation de la carte, la position du robot n'est pas un sous pavage mais bien une position, puisqu'elle représente l'origine de la carte. Par contre, lors de l'ajout d'un nouvel amer, la position du robot est définie par un ensemble de pavés, issus de la phase de localisation. C'est sur ce deuxième cas plus complexe que nous allons détailler notre fonction d'inclusion.

Il faut rappeler, à cette étape, que l'image directe d'une observation à partir d'un sous pavage issu de la phase de localisation fournit bien sûr un ensemble de pavés mais non nécessairement disjoints. Il nous faut donc l'approximer par un sous pavage plus facile à utiliser et à représenter dont les seules zones d'intersection soient les bornes des pavés. Pour calculer cet ensemble nous allons de nouveau utiliser l'algorithme SIVIA, qui, en partant d'un pavé initial nous fournira le sous pavage voulu. Ceci permettra d'approximer chaque nouvel amer à insérer dans la carte. Nous devons donc, avant d'appeler SIVIA, calculer un pavé initial. Ceci se fait aisément en calculant les minimums et maximums à partir d'une observation pour chaque pavé de localisation (Figure 5-3).

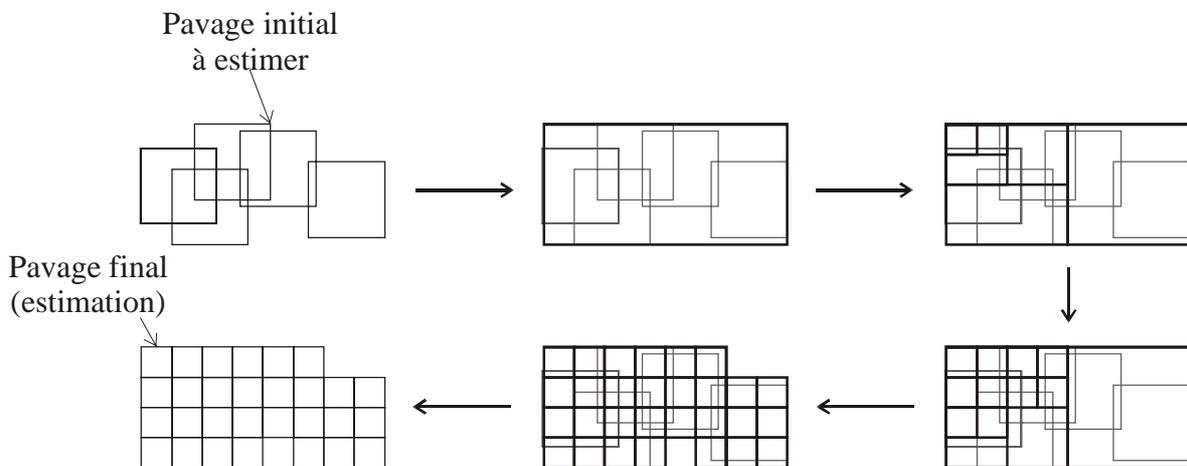


Figure 5-3 : Approximation d'un ensemble de pavés par l'utilisation de SIVIA

La fonction d'inclusion utilisée par SIVIA lors de cette étape se décline de la manière suivante. Nous calculons pour chaque pavé de localisation s'il existe une intersection entre l'observation considérée et le pavé à tester. Dès que l'intersection est non vide, la fonction retourne la valeur indéterminée. Ceci nous donne donc l'algorithme suivant :

```

Calculer le pavé [Observation] à partir du pavé [ $\phi$ ]i et [ $l$ ]i
Pour chaque pavé [ $L$ ]g résultat de la localisation
  Calculer [Observation]g à partir de [Observation] et [ $L$ ]g
  Si [Observation]g  $\cap$  [Pavé_recu]  $\neq \emptyset$ 
    retourner indetermine
  finsi
finpour
retourner faux

```

Dans le cas de l'initialisation de la carte, l'algorithme est réduit, car n'ayant aucun pavé représentant la localisation du robot, nous avons directement le sous pavage correspondant à l'observation.

5.3. Algorithme de localisation modifié

Nous présentons dans ce paragraphe un algorithme de localisation légèrement modifié par rapport à la version du chapitre 4 en ce sens que nous ne considérons plus que les amers sont connus mais estimés par une approche ensembliste.

Nous avons présenté dans le chapitre précédent un algorithme de localisation basé sur l'utilisation de SIVIA. Nous avons testé deux modèles d'erreur possible associés à la mesure, et nous avons finalement conclu que le plus robuste était celui basé sur le modèle télémétrique [Kieffer 99]. A chaque mesure nous associons donc une erreur en distance et une erreur en angle, formant ainsi une portion d'anneau.

Dans l'algorithme de localisation utilisant une carte connue de l'environnement, une fois déterminé le pavé correspondant à une donnée sensorielle, il nous reste à tester si un point de la carte se trouve dans ce pavé, grâce à la fonction `test_validite_pave()` détaillé dans le précédent chapitre.

Ici, étant donné que nous cherchons à estimer la position des amers verticaux de l'environnement par un sous pavage afin d'obtenir une information d'imprécision, un amer j de l'environnement n'est plus représenté par un point P_j , mais par un sous pavage constitué de n pavés, que nous noterons :

$$\llbracket P \rrbracket_j = \{ [T]_r / 1 \leq r \leq n \} \quad [5-2]$$

L'obtention de ce sous pavage a été détaillé dans le paragraphe précédent. La stratégie consiste alors à tester, pour chaque élément $\llbracket P \rrbracket_j$ de la carte s'il existe une intersection avec le pavé observé $[S]_i$:

```

Fonction test_validite_pave([S]i)
valide=faux
Pour chaque élément  $[[P]]_j$  de la carte
    Pour chaque pavé  $[T]_r$  de  $[[P]]_j$ 
        Si  $[T]_r \cap [S]_i \neq \emptyset$  alors valide=vrai
    fin pour
fin pour
renvoyer valide
    
```

Nous avons mis en évidence sur la Figure 5-4 les différentes possibilités pouvant être rencontrées dans le cadre d'une interaction pavé observé – sous pavage cartographié. Ces cas de figures n'étaient bien sûr pas gérés dans l'algorithme de localisation du chapitre précédent.

Nous pouvons constater sur la Figure 5-4 qu'il y a une intersection non vide pour l'observation 1 et 4, que l'observation 2 contient le sous pavage complet d'un élément de la carte, et que par contre l'observation 3 n'est en relation avec aucun élément de la carte.

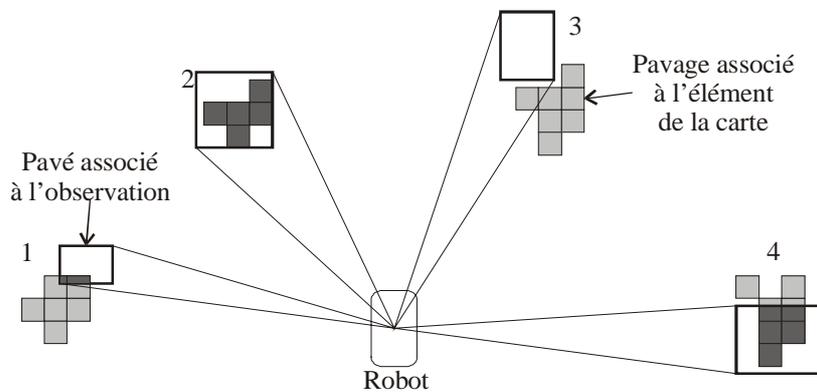


Figure 5-4 : Test d'intersection utilisé dans l'algorithme de localisation

Il s'agit de l'unique différence qui existe avec l'algorithme de localisation précédemment développé, tous les autres tests restant les mêmes. Nous avons donc conservé la partie du traitement concernant les données aberrantes, ainsi que l'intégration de l'information odométrique.

Dans une démarche similaire à celle exposée dans le paragraphe précédent, un pavé est toujours divisé s'il peut convenir, ce qui a pour conséquence de n'obtenir qu'un sous pavage extérieur correspondant à l'estimation de position ne contenant que des pavés de taille inférieure à la précision souhaitée.

5.4. Méthode de décision pour la mise en correspondance

5.4.1 Détermination de la croyance accordée à chaque association

Tout comme la méthode de localisation du chapitre 3, il faut déterminer quelles sont les informations à fusionner et quelles sont celles à ajouter à la carte en tant que nouvelles primitives. La méthode de décision utilisée ici sera similaire, et consistera à déterminer une

croissance pour chaque association, toujours en utilisant la théorie de Dempster-Shafer [Dempster 67] [Shafer 76]. Cette partie du traitement est cruciale et prépondérante dans le paradigme de localisation et de modélisation simultanée. En effet, c'est cette étape qui conditionnera le maintien de la cohérence de la carte de l'environnement. Une mauvaise décision entre nouvelle insertion ou fusion sera généralement à la base d'un surnombre de primitives cartographiées qui entraînera des erreurs cumulatives et donc une divergence de l'algorithme.

Nous avons au début de cette phase trois informations d'imprécisions disponibles qui seront utilisées :

- Une carte de l'environnement constituée de sous pavages, chacun représentant l'imprécision associée à l'amer modélisé,
- Un ensemble de données sensorielles caractérisées par des informations de type distance-angle sous forme d'intervalles, donnant l'imprécision sur la mesure,
- Un sous pavage résultant de l'étape de localisation, représentant l'imprécision associée à la position du robot.

La stratégie consistera à répercuter l'imprécision associée à la mesure sur l'ensemble des positions du robot, pour ensuite comparer cet ensemble aux sous pavages de la carte. Il s'agit là d'une phase importante de notre démarche de localisation/modélisation parce qu'elle permet de prendre en compte l'interaction entre erreur de mesure, erreur sur l'estimation et erreur sur les primitives cartographiées pour la mise en correspondance. Nous devons alors résoudre deux problèmes principaux :

- Comment caractériser et utiliser l'ensemble résultant de l'association de l'imprécision sur la localisation avec celle sur la mesure ?
- Quel critère de comparaison pouvons nous appliquer pour déterminer la croyance accordée à la fusion de cet ensemble avec un sous pavage de la carte ?

Ces deux questions étant intimement liées, la réponse à la première question dépend de celle de la deuxième. Pour savoir si une observation peut être ou non associée à une primitive cartographiée, il nous faut trouver un critère de comparaison entre les deux. Le critère qui nous a semblé le plus intuitif est l'intersection. En effet, plus l'ensemble associé à une observation englobe le sous pavage représentant un point de la carte, plus la certitude qu'il représente la même information est importante, impliquant qu'ils doivent être fusionnés. Ce critère sera détaillé dans la suite de ce chapitre.

Etant donné que les pavés d'une observation peuvent se chevaucher, plusieurs d'entre eux peuvent avoir une intersection non vide avec un des pavés représentant un point de la carte. Nous ne pouvons donc pas utiliser directement la notion d'intersection entre ces différents pavés pour le calcul du volume. En effet, si nous considérons trois ensembles A, B et C tels que :

$$A \cap B \cap C \neq \emptyset$$

alors nous avons l'inéquation suivante :

$$\text{Volume}(A \cap C) + \text{Volume}(B \cap C) > \text{Volume}(A \cap B \cap C) \quad [5-3]$$

qui signifie que le volume correspondant à l'intersection des ensembles A et B avec C est comptabilisé deux fois dans la partie gauche de l'inéquation.

La première solution à laquelle nous avons pensé pour calculer le volume d'intersection entre le sous pavage d'un amer de la carte et les pavés issus d'une observation, consistait à caractériser ce dernier ensemble par son contour afin d'obtenir un polygone. L'étape suivante

réside alors dans le calcul de l'intersection qui existe entre ce polygone et le sous pavage $\llbracket P \rrbracket_j$ (Figure 5-5).

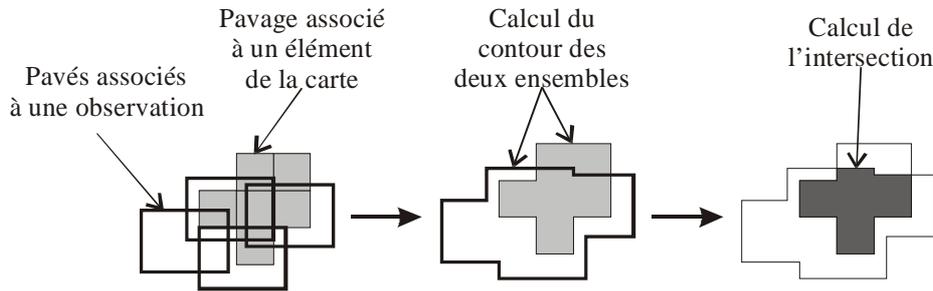


Figure 5-5 : Calcul du volume d'intersection par utilisation de contours

Cette méthode a le gros inconvénient de nécessiter l'utilisation d'intersections complexes. Cette complexité pourrait être évitée si les deux ensembles étaient des sous pavages. En effet, les éléments constituant des sous pavages étant des pavés, et le calcul d'intersections entre pavés étant relativement simple, il semble alors plus implicite de travailler avec ce type de primitives élémentaires.

La solution retenue est alors la même que lors de l'ajout d'une nouvelle observation dans la carte (cf. 5.2), à savoir calculer l'image du sous pavage issu de la localisation, et appliquer SIVIA pour obtenir un sous pavage associé à l'observation que nous noterons :

$$\llbracket S \rrbracket_i = \{ [K]_q / 1 \leq q \leq m \} \quad [5-4]$$

avec $1 \leq i \leq s$ et m représentant le nombre de pavés constituant le sous pavage.

Notre critère de comparaison sera donc basé sur la valeur :

$$\tau = (\text{Volume}(\llbracket P \rrbracket_j) - \text{Volume}(\llbracket P \rrbracket_j \cap \llbracket S \rrbracket_i)) \times 100 \quad [5-5]$$

qui représente le pourcentage de $\llbracket P \rrbracket_j$ inclu dans $\llbracket S \rrbracket_i$.

Le formalisme utilisé pour déterminer la croyance associée à une fusion est identique à celui utilisé dans le chapitre 3, c'est-à-dire basé sur la recherche du maximum de croyance par rapport à l'application des règles de Dempster-Shafer. Nous déterminons pour chaque couple associant une observation et un élément de la carte des jeux de masses corrélés à la fusion ou au contraire à la non fusion. Il nous faut alors définir notre cadre de discernement constitué, tout comme dans le chapitre 3, de deux éléments :

$$\Theta = \{ \text{OUI}, \text{NON} \}$$

Ce cadre de discernement correspond aux assertions :

- "OUI" il faut fusionner l'observation i avec l'élément j de la carte.
- "NON" il ne faut pas fusionner l'observation i avec l'élément j de la carte.

La robustesse de la décision, consistant à fusionner ou à initialiser une primitive, passera par le calcul et l'ajustement des jeux de masses. La première méthode que nous avons testée est basée sur l'utilisation d'un seuil ν , compris entre 0 et 100% (Figure 5-6) qui reste à

déterminer de façon précise. Les trois jeux de masses, pour le "OUI" le "NON" et l'ignorance étant calculés à partir de ce seuil.

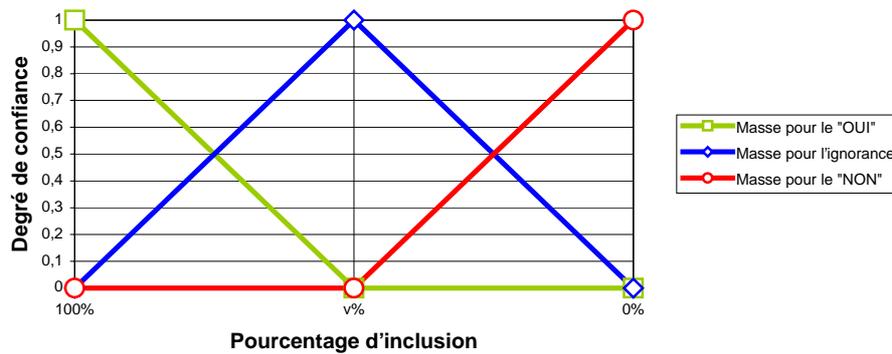


Figure 5-6 : Fonction de masse pour le pourcentage d'inclusion

Nous voulons que la croyance pour le "OUI" soit positive au dessus de ce seuil, et que la croyance pour le "NON" soit positive en dessous, ceci représentant le fait que si le pourcentage d'intersection des deux ensembles dépasse une certaine quantité v alors la fusion est probable. Les masses correspondent donc aux équations suivantes :

$$m(OUI) = \begin{cases} \frac{\tau - v}{100 - v} & \text{Si } v < \tau < 100 \\ 0 & \text{Sinon} \end{cases}$$

$$m(NON) = \begin{cases} 1 - \frac{\tau}{v} & \text{Si } 0 < \tau < v \\ 0 & \text{Sinon} \end{cases}$$

$$m(\Theta) = \begin{cases} 1 - m(OUI) & \text{Si } v < \tau < 100 \\ 1 - m(NON) & \text{Sinon} \end{cases}$$

Comme nous utilisons l'analyse par intervalle, l'observation d'un amer i indique que si sa position réelle existe elle est obligatoirement dans le sous pavage $\llbracket S \rrbracket_i$ associé à cette amer. Nous n'avons par contre aucune information sur la certitude de cette donnée. En outre, nous savons que les amers ne sont pas tous systématiquement observés au cours du déplacement du robot. Enfin, comme nous le verrons dans la suite de ce chapitre, si une observation et un élément de la carte représentent réellement la même information, le résultat de cette fusion sera obligatoirement l'intersection des deux ensembles.

Nous nous apercevons que la valeur du seuil v est déterminante dans la construction incrémentale de la carte. Plusieurs raisons justifient cette prépondérance :

- La précision de l'étape de localisation dépend du nombre d'éléments dans la carte de l'environnement : dans le chapitre 3 nous avons mis en évidence que plus le nombre d'amers erronés est important, plus nous augmentons le risque d'une mauvaise localisation. La stratégie consiste donc à insérer le moins souvent possible d'informations superflues,
- La précision de la localisation dépend également de la taille des sous pavages contenus dans cette carte, il faut donc chercher à faire diminuer rapidement le nombre de pavés constituant chaque amer,

- Malgré tout, une fusion erronée générera une représentation erronée de l'amer, ce qui aura des conséquences non seulement sur l'étape de localisation suivante, mais aussi sur la phase de modélisation qui en découlera.

L'importance de cette valeur sur la pertinence du processus de localisation/modélisation et parce que ce seuil est la seule valeur conditionnant la décision d'intégrer ou non une nouvelle donnée, il nous a semblé intéressant de tester différentes valeurs pour choisir celle qui sera la plus optimale. Nous constatons que lorsque ce seuil diminue, les observations sont fusionnées plus souvent, par contre lorsque ce seuil augmente, cela favorise la création de nouveaux éléments en surnombre dans la carte. Les meilleurs résultats obtenus, en terme de fusion et en terme de nombre d'amers insérés dans la carte fixent à 25% la valeur de v . Toutefois, nous voulons que la croyance pour le "OUI" soit toujours supérieure à 0,5 au dessus de 50% d'inclusion. En effet, si le sous pavage issu d'une observation englobe plus de 50% des pavés caractérisant un amer, nous considérons que la croyance pour la fusion doit être la plus élevée. Ceci nous a contraint à modifier légèrement les fonctions de masses pour obtenir finalement celles mentionnés sur la Figure 5-7.

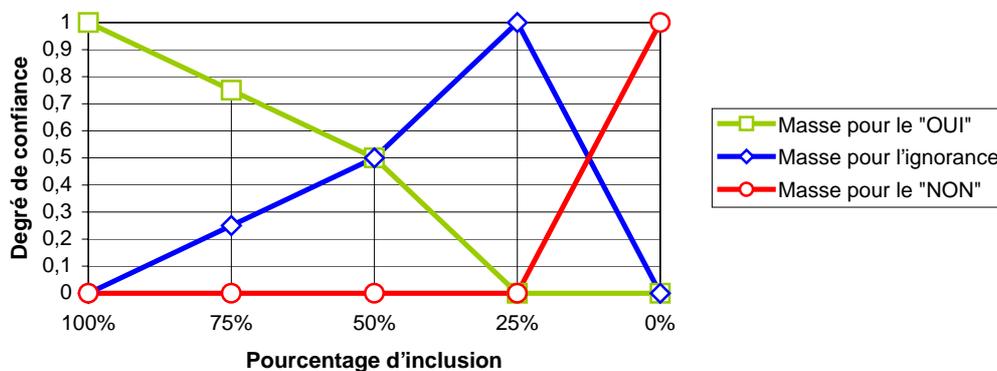


Figure 5-7 : Fonction de masse modifiées pour le pourcentage d'inclusion

Le calcul des différents jeux de masses est alors donné par les équations suivantes :

$$m(OUI) = \begin{cases} 0,01 \times v & \text{si } v > 50 \\ 0,02 \times v - 0,5 & \text{si } 25 < v < 50 \\ 0 & \text{sinon} \end{cases} \quad [5-6]$$

$$m(NON) = \begin{cases} -0,04 \times v + 1 & \text{si } v < 25 \\ 0 & \text{sinon} \end{cases} \quad [5-7]$$

$$m(\theta) = 1 - m(OUI) - m(NON) \quad [5-8]$$

Il reste maintenant à calculer le volume d'intersection qui existe entre chaque sous pavage $[[S]]_i$ issu d'une observation et chaque sous pavage $[[P]]_j$ représentant un amer dans la carte. L'algorithme que nous avons implémenté est le suivant :

```

Pour chaque observation  $[[S]]_i, 1 \leq i \leq s$ 
  Pour chaque amer  $[[P]]_j, 1 \leq j \leq p$ 
    volume_total = volume( $[[P]]_j$ )
    volume_partiel = 0
    Pour chaque pavé  $[P]_r \in [[P]]_j$ , tel que  $1 \leq r \leq n$ 
      Pour chaque pavé  $[S]_q \in [[S]]_i$ , tel que  $1 \leq q \leq m$ 
        Calculer volume_temp = volume d'intersection de  $[S]_q$  et  $[P]_r$ 
        volume_partiel = volume_partiel + volume_temp
      Fin pour
    Fin pour
     $v = \text{volume\_partiel} / \text{volume\_total}$ 
    Calcul des  $m_{i,j}(\text{OUI})$ ,  $m_{i,j}(\text{NON})$  et  $m_{i,j}(\theta)$  en fonction de  $v$ 
  Fin pour
Fin pour

```

Nous retrouvons ici nos fonctions de masses $m_{i,j}$, qui représentent les p triplets associés à chaque observation. Ce sont ces mêmes triplets qui seront utilisés dans la phase suivante concernant la décision.

5.4.2 Algorithme de décision

L'algorithme de décision que nous utilisons est basé sur le maximum de vraisemblance obtenu au sens de Dempster-Shafer. La phase précédente nous a permis de calculer pour chaque observation, p triplets correspondant à l'association avec chaque élément de la carte. Nous pouvons alors appliquer l'opérateur de Dempster-Shafer généralisé, afin d'obtenir une matrice de croyance de dimension $s \times (p+2)$ (cf. chapitre 3). L'algorithme de décision utilisé est semblable à celui utilisé dans le chapitre 3. Nous cherchons ainsi dans cette matrice l'élément ayant la valeur maximale. Si cet élément est trouvé c'est qu'une association est réalisée et nous passons ensuite l'ensemble des éléments restant de la ligne et de la colonne à 0 (cf. 3.2.3).

Néanmoins, nous avons dû modifier légèrement cet algorithme pour tenir compte des particularités des données traitées. En effet, contrairement à la méthode utilisée dans le chapitre 3, nous ne pouvons pas ici créer de nouvel élément dans la carte lorsque le poids sur le doute est maximum. En effet, dans ce cas le sous pavage correspondant influence de manière significative et négative la précision des phases de localisation suivantes. Il nous faut donc porter une attention particulière à la phase d'ajout de nouveaux amers dans la carte, pour que le nombre de primitives n'augmente pas de manière incohérente.

La modification apportée à l'algorithme de décision consiste à chercher le maximum du tableau sans prendre en compte l'ignorance, mais respectant la condition : $m_{i,j} < m_i(\theta)$. Si ce maximum met en correspondance une observation et un amer de la carte, la décision prise est la fusion de ces deux informations. La ligne ainsi que la colonne contenant cette valeur sont passées à 0, exactement comme dans le chapitre 3. S'il s'agit d'un nouveau point, là aussi la méthode est identique, le choix est porté sur la création d'un nouvel élément dans la carte et la ligne est passée à 0.

Une fois ce traitement effectué, nous avons un ensemble d'observations pour lesquelles aucune décision n'a été prise. Trois cas sont envisageables (Figure 5-8) :

- Si dans la ligne de la matrice contenant cette observation tous les éléments sont à 0, cela signifie que l'ignorance est dû à une croyance faible entre l'observation et un ou plusieurs éléments de la carte. En effet, ce cas de figure correspond à des amers de la carte qui étaient susceptibles d'être fusionnés avec cette observation mais qui ont déjà été appariés avec une plus grande croyance. Par conséquent, il ne peut s'agir ici que d'un nouvel élément que nous ajoutons à la carte (Figure 5-8-1).
- Si l'unique valeur différente de 0 (mise à part l'ignorance) est "*", c'est qu'il n'y a pas d'intersection suffisante entre l'observation et un élément de la carte pour les fusionner (<25%). Ici aussi, nous décidons de créer un nouveau sous pavage caractérisant ce nouvel amer (Figure 5-8-2).
- Enfin, s'il existe sur la ligne plusieurs valeurs différentes de 0, mettant en relation l'observation considérée et un ou plusieurs éléments de la carte, c'est que le volume d'intersection était compris entre 25% et 50%. Dans ce cas, nous ne pouvons pas fusionner ces informations, car comme nous le verrons dans la suite de ce chapitre, la fusion est basée sur l'intersection des deux ensembles. Il s'agit typiquement d'un cas de figure où nous risquons d'insérer un élément inexistant dans la carte de l'environnement. Nous avons posé préalablement l'hypothèse de travail consistant à minimiser le nombre d'insertions en ne prenant en compte que les mises en correspondances les plus robustes. En effet, l'insertion "non conditionnée" de primitives constitue un facteur prépondérant dans la dérive du processus de localisation/modélisation. Nous préférons dans ce cas ne pas tenir compte de cette observation pour un maintien robuste de la cohérence de la carte (Figure 5-8-3).

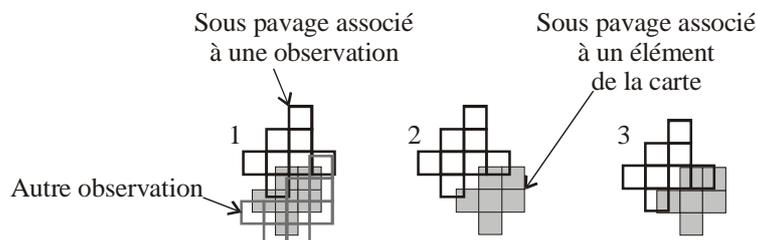


Figure 5-8 : Trois cas de figure possibles concernant l'ignorance sur la fusion

Cet algorithme nous donne finalement deux ensembles, le premier constitué des observations à fusionner avec un élément de la carte, et le deuxième constitué des nouveaux amers à ajouter. Le traitement et la gestion de ces deux ensembles vont être présentés dans la partie suivante de ce chapitre.

5.5. Mise à jour incrémentale de la carte de l'environnement

L'étape décisionnelle intégration/fusion précédente nous a fourni deux ensembles de points : le premier contenant ceux à fusionner et l'autre ceux à ajouter à la carte. L'intégration d'un nouvel élément à la carte de l'environnement a déjà été donnée au paragraphe 5.2.

La dernière étape restant à traiter est la fusion entre un élément de la carte et une observation. Ici, ces deux informations sont caractérisées par des ensembles, et comme nous nous trouvons dans un contexte à erreur bornée, la position réelle de l'amer appartient

obligatoirement aux deux ensembles. Le résultat de la fusion d'une observation avec un élément de la carte est donc l'intersection des deux ensembles.

A ce niveau nous devons résoudre un problème. En effet, chaque ensemble est caractérisé par un ensemble de pavés : celui représentant l'observation contient même des pavés pouvant se chevaucher. Le calcul de l'intersection revient alors à traiter le problème d'intersections multiples de rectangles disjoints. Il s'agit d'un problème non trivial.

Pour palier à cette difficulté, nous sommes partis de la constatation suivante : puisque la solution appartient aux deux ensembles, il semble intuitif d'en considérer un des deux, pour ensuite vérifier que chaque pavé le constituant appartient bien au deuxième ensemble. Si c'est la cas, le pavé est conservé, sinon, il est supprimé. Le sous pavage le plus adapté comme premier ensemble est alors celui représentant l'amer dans la carte, puisqu'il est uniquement constitué de pavés disjoints (Figure 5-9).

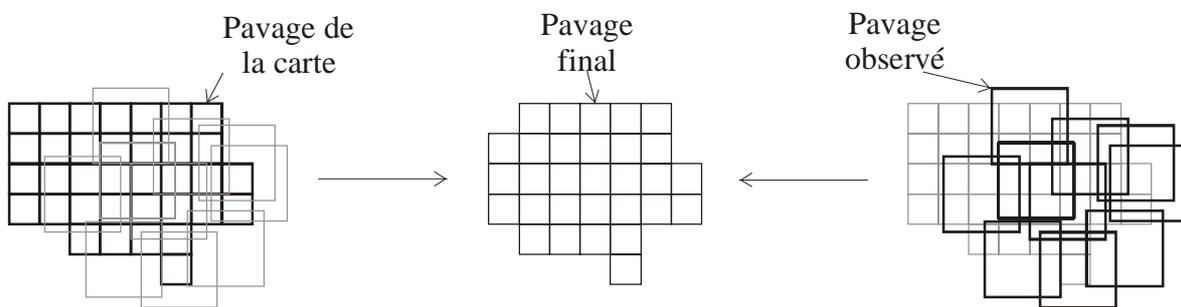


Figure 5-9 : Exemple de fusion entre observation et élément de la carte

Cette méthode donne lieu à l'algorithme suivant :

```

Pour chaque élément observé  $i$  à fusionner avec un élément  $j$  de la carte
  Pour chaque pavé  $[T]_r \in [[P]]_j$ 
    Pour chaque pavé  $[L]_g$  résultat de la localisation
      Calculer  $[K]_{g,i} = f^I([L]_g)$  à partir de l'observation  $i$ 
      Si  $[K]_{g,i} \cap [T]_r \neq \emptyset$  alors
        Le pavé  $[T]_r$  est noté "à garder"
      Finsi
    Finpour
  Si  $[T]_r$  n'est pas noté "à garder" alors supprimer  $[T]_r$ 
  Finpour
Finpour
    
```

Nous pouvons noter à ce niveau que notre méthode de fusion ne peut avoir pour résultat qu'une diminution du sous pavage représentant l'imprécision d'un amer dans la carte. Quelque soit l'ensemble associé à l'observation, il ne peut y avoir après fusion qu'un apport d'information, en ce sens que le sous pavage de l'amer ne pourra augmenter.

De la même manière que nous l'avons expliqué dans le chapitre 3, des informations sensorielles ne correspondant à rien dans l'environnement peuvent être intégrées à la carte. Nous appliquons ici la même méthode de gestion de ces données aberrantes, qui consiste à affecter une pondération sur chaque observation. Ces dernières sont utilisées pour supprimer de la carte les informations n'ayant plus été observées depuis un temps t , et qui ont un poids inférieur à b (cf. 3.4.1).

Afin de valider notre approche, nous présentons des résultats expérimentaux dans le paragraphe suivant obtenus dans les deux environnements de références, déjà utilisés dans le chapitre 4.

5.6. Résultats expérimentaux

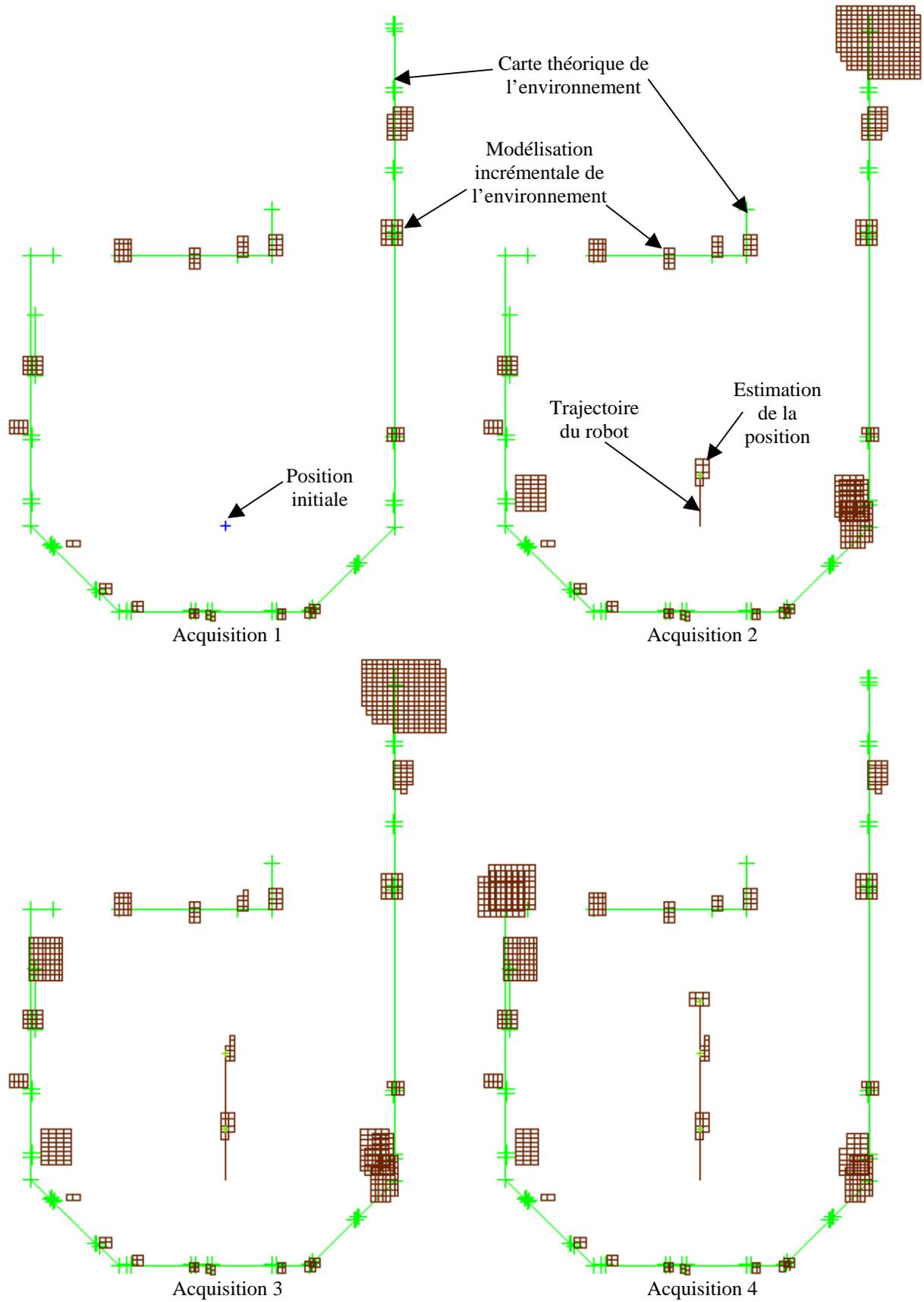
Dans le chapitre 3, nous avons présenté une méthode de gestion de liens entre les différents amers de la carte. Nous avons développé des algorithmes qui permettent de minimiser le nombre de liens dans la carte. Nous avons constaté expérimentalement que les liens obtenus après plusieurs acquisitions étaient tout à fait cohérents par rapport à la réalité terrain. Néanmoins, la modélisation des amers par des pavés ne se prêtant pas à la représentation des liens, nous n'intégreront pas ces derniers aux résultats expérimentaux. Toutefois, cette intégration de liens pourrait facilement être appliquée en considérant par exemple les centres des sous pavages comme étant le barycentre de l'ensemble des pavés.

Nous avons testé notre méthode de localisation et de modélisation incrémentale dans les deux types d'environnement structuré déjà présentés dans les précédents chapitres. Enfin, pour avoir des éléments de comparaison avec la première méthode de mise à jour de la carte, nous avons bien sûr utilisé les mêmes trajectoires et les mêmes acquisitions.

5.6.1 *Tests en environnement structuré de type hall*

Cette première série d'acquisitions nous a permis de valider notre paradigme de localisation et modélisation simultané dans un environnement de petite taille. Il nous permettra de comparer qualitativement les résultats obtenus, non seulement en termes de localisation avec ceux du chapitre précédent où l'environnement était parfaitement connu, mais aussi en termes de cohérence et de précision de la carte modélisée avec celle gérée dans le chapitre 3.

Comme nous l'avons déjà détaillé précédemment, certains éléments de la carte peuvent être supprimés au fur et à mesure des phases de localisation et de mise à jour. En effet, le filtrage que nous avons développé dans le chapitre 3, et que nous utilisons également ici, permet de ne garder dans la carte que les éléments ayant déjà été observés plusieurs fois.



Tout d'abord, et d'un point de vue général, le processus de localisation et de modélisation simultané donne des résultats cohérents en termes de précision et en termes de robustesse. D'un point de vue localisation, nous pouvons affirmer que l'absence de connaissance a priori n'a pas affecté l'étape d'estimation de la configuration du robot par une approche ensembliste. En effet, le sous pavage caractérisant l'imprécision sur la localisation est cohérent puisqu'il englobe dans la majorité des cas la position réelle du robot. La précision obtenue est bonne puisque le nombre de pavés pour chaque estimation de configuration est peu élevé et proche de celle obtenue dans le chapitre précédent où l'environnement était parfaitement connu. La cohérence de la phase de localisation est aussi mise en évidence par sa variation au cours du déplacement du robot : nous voyons que ce sous pavage est plus réduit au retour (c'est-à-dire après le demi tour) qu'à l'aller. Ceci s'explique par le fait que le robot utilise de manière précise et robuste les primitives cartographiques de l'environnement déjà modélisées au premier passage. Ce phénomène sera observable de manière plus significative sur des trajets plus importants dans notre deuxième environnement d'expérimentation.

D'un point de vue modélisation, et toujours par rapport à des constatations d'ordre général, nous pouvons affirmer que la carte générée est cohérente par rapport à la vérité terrain. Le nombre de primitives cartographiques intégrées est cohérent, prouvant ainsi la validité du processus de fusion et d'intégration. En termes de quantification de l'imprécision, et rappelons ici qu'il s'agit de l'extension que nous avons souhaité donner à la méthode présentée au chapitre 3, nous pouvons remarquer que l'approche ensembliste fournit des sous pavages qui englobent quasi systématiquement les positions réelles des éléments verticaux de l'environnement.

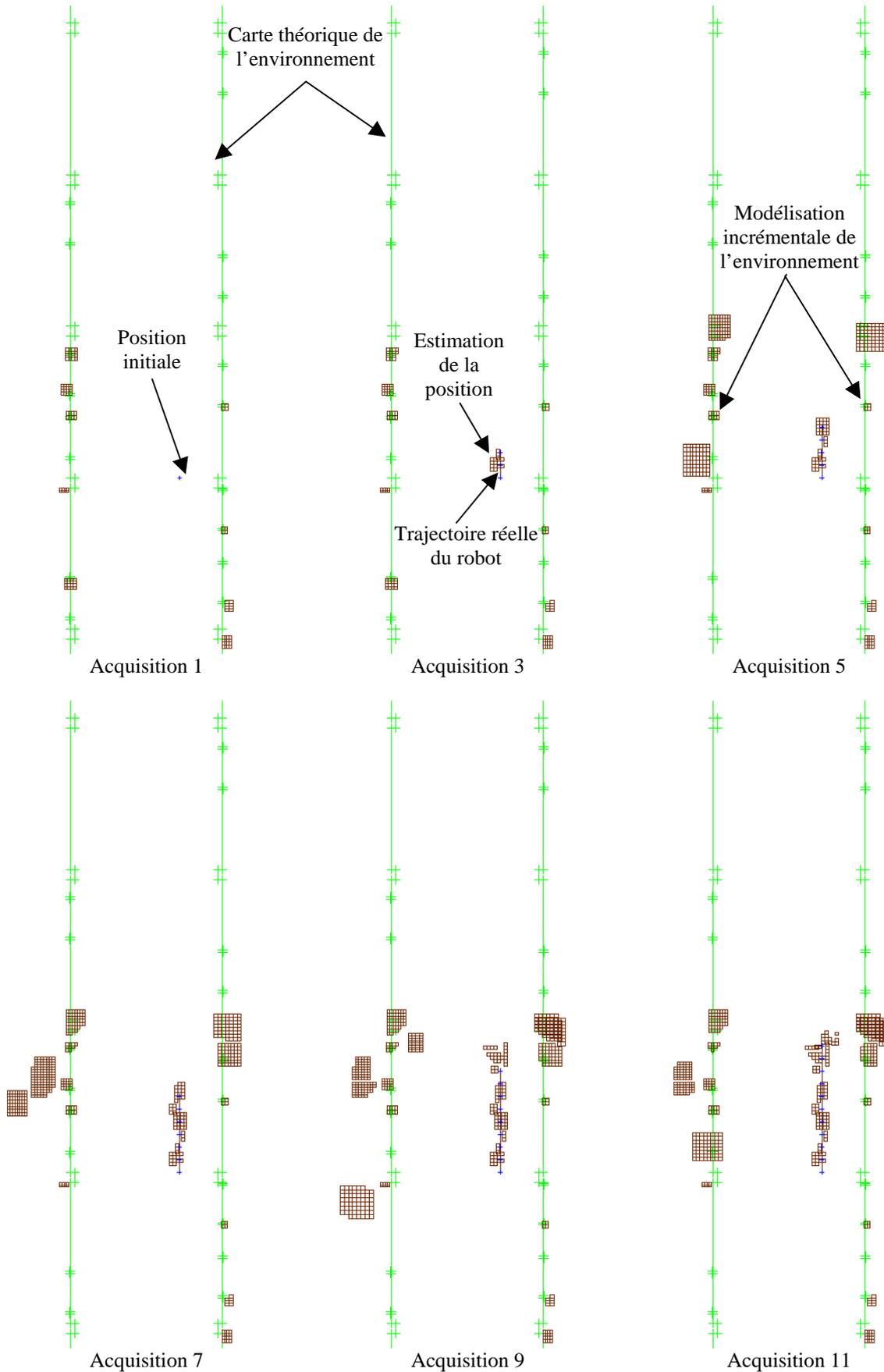
L'évolution des sous pavages au cours du processus incrémental de modélisation est robuste et cohérente. Deux points peuvent justifier ce constat : d'une part l'apport d'informations sensorielles s'accompagne bien d'une diminution de la taille du domaine d'erreur et d'une convergence du sous pavage vers la position réelle des amers. D'autre part l'interaction entre erreur de localisation et erreur de modélisation est prise en compte car plus la précision sur la localisation est importante, plus les sous pavages d'erreur sur les primitives cartographiques sont réduits de façon significative. Il s'agit là d'un facteur prépondérant qui permet de ne pas faire diverger le processus simultané de localisation et de modélisation au bout d'un certain nombre d'acquisitions. La dérive de ce paradigme incrémental doit cependant être testée sur des trajets importants sur un aller et retour. Ce test sur des environnements de grandes tailles est en effet important puisque notamment mis en avant dans des travaux tels que ceux de Dieter fox.

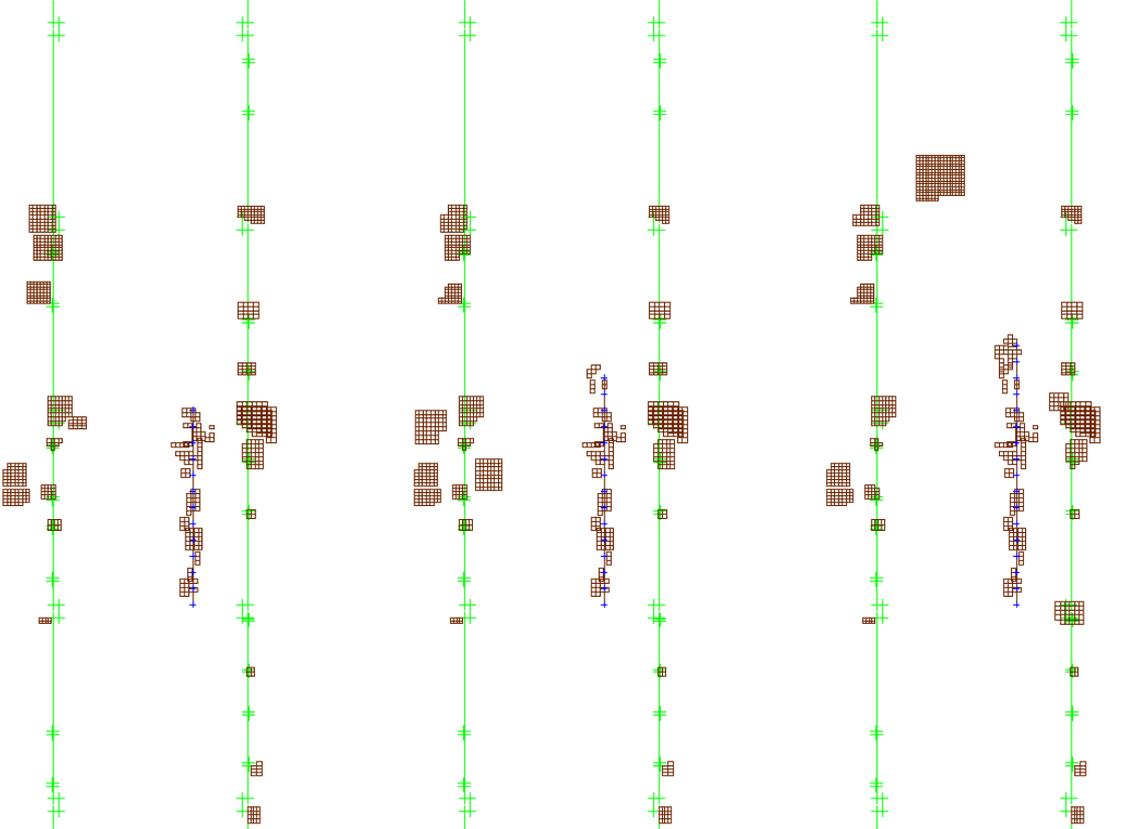
5.6.2 *Résultats en environnement de type couloir sur des trajets importants*

Nous pouvons rappeler ici que cette deuxième série de mesures a déjà été utilisée sur la méthode de localisation et de modélisation simultanée présentée dans le chapitre 3. Il est à noter que le processus incrémental de génération de carte au sens des moindres carrés avait divergé au bout de la 21^{ème} acquisition. Nous montrons que l'approche ensembliste permet d'obtenir de meilleurs résultats puisque même sur des distances importantes le processus ne diverge plus.

Cette deuxième série de mesures est toujours constituée de 45 acquisitions réalisées le long d'une trajectoire constituée d'un aller-retour. La distance parcourue est d'environ quatorze mètres. Là encore, nous utilisons l'odométrie afin de diminuer la taille du pavé de recherche

des positions possibles du robot. Nous rappelons à ce propos que l'erreur odométrique est maximisée afin de servir uniquement à l'initialisation de l'algorithme SIVIA.

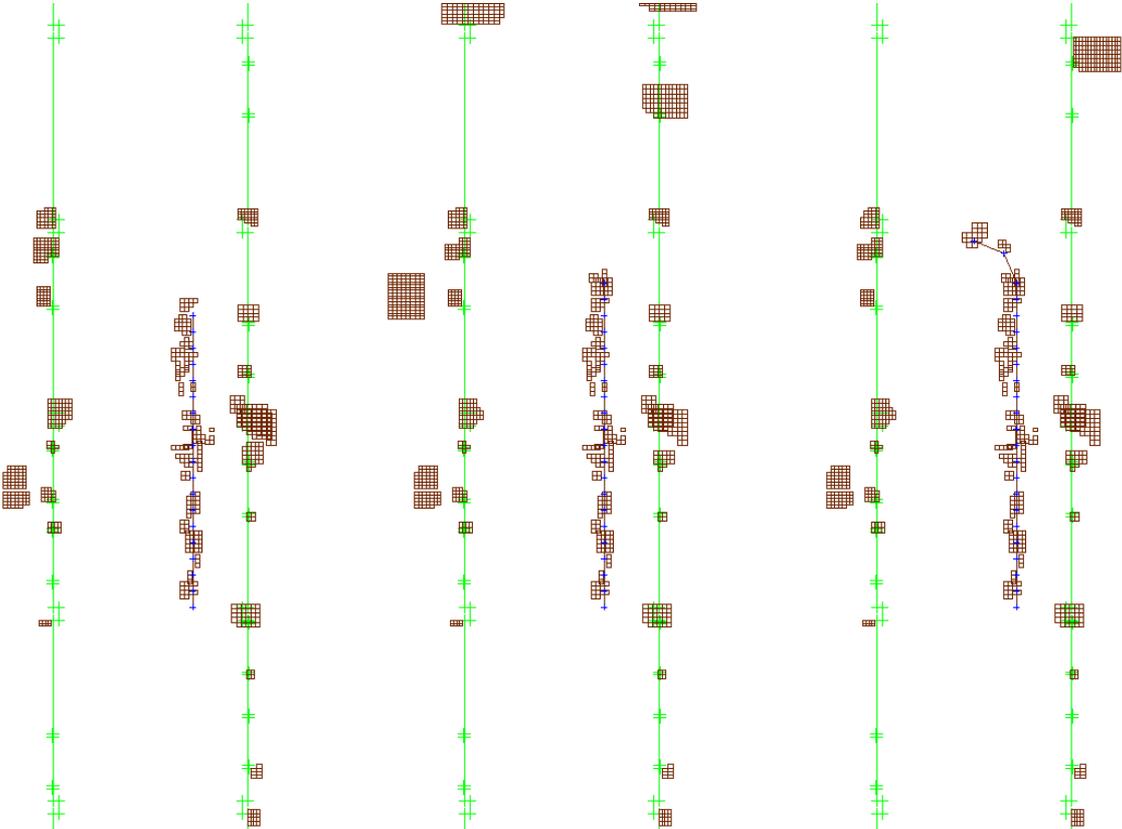




Acquisition 13

Acquisition 15

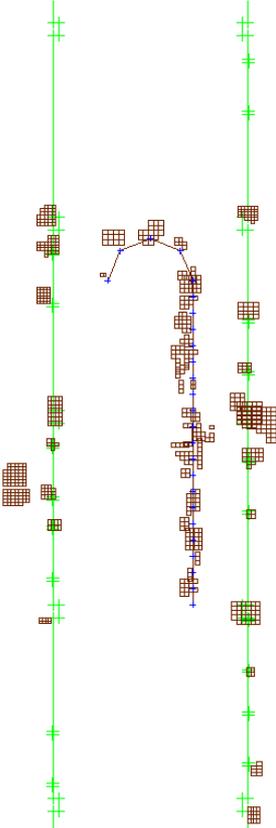
Acquisition 17



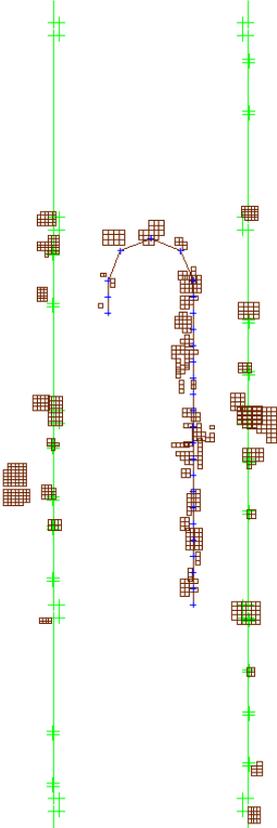
Acquisition 19

Acquisition 21

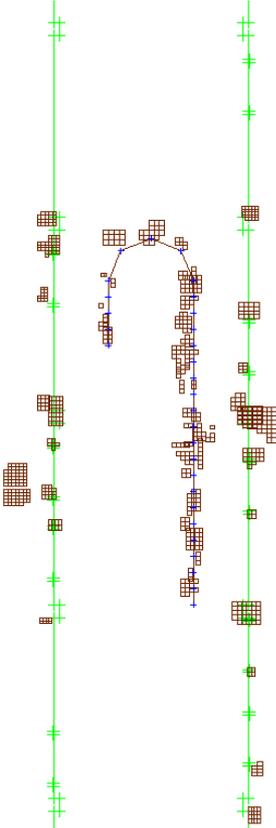
Acquisition 23



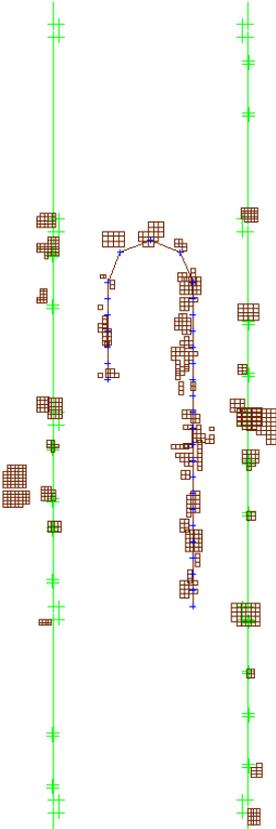
Acquisition 25



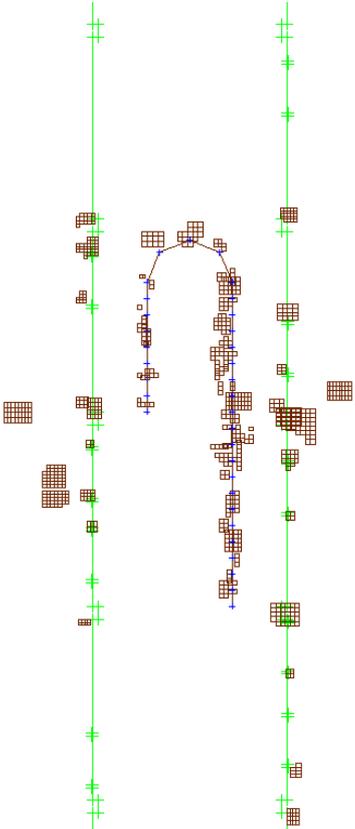
Acquisition 27



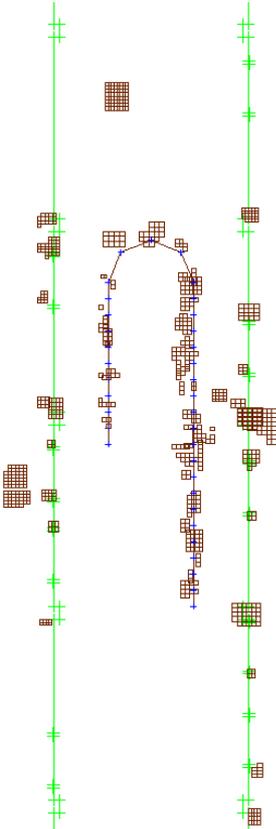
Acquisition 29



Acquisition 31



Acquisition 33



Acquisition 35

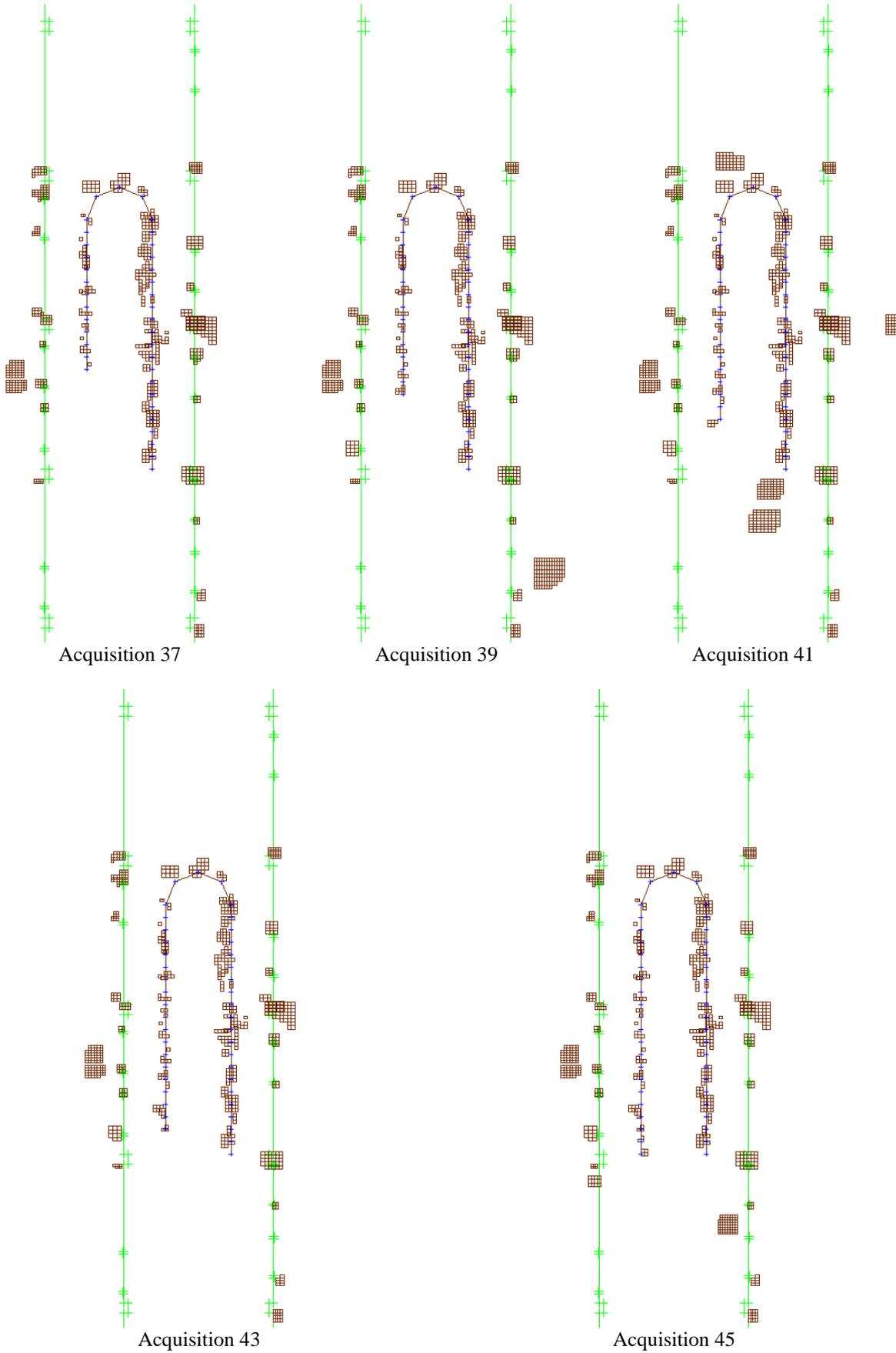


Figure 5-11 : Résultats de localisation et de modélisation simultanée sur 45 acquisitions

La première constatation que nous pouvons faire est que le processus de localisation et modélisation simultanée ne diverge pas. En effet, et contrairement aux résultats obtenus dans le chapitre 3, les 45 acquisitions aboutissent à une construction cohérente de la carte de l'environnement sans connaissance préalable.

D'un point de vue localisation, et tout comme pour la série précédente d'acquisition, l'estimation de la configuration du robot est robuste et la quantification de l'imprécision reste englobante dans la quasi totalité des cas. Les constatations concernant l'évolution du domaine d'imprécision au cours du déplacement sont ici renforcées : les sous pavages obtenus lors du retour du robot sont de plus petites tailles qu'à l'aller. La localisation est plus précise lorsque le robot peut exploiter les connaissances sur l'environnement accumulées sur les acquisitions antérieures. La cohérence de la quantification sur l'imprécision associée à la localisation est mise en évidence sur les acquisitions faites dans les zones de l'environnement où le nombre d'amers est plus faible. La portion de trajectoire où le robot fait demi tour en est une illustration probante : le nombre d'amers faibles et une mauvaise configuration angulaire au niveau de l'observabilité font que l'imprécision sur la localisation augmente fortement.

D'un point de vue modélisation, les mêmes remarques que celles faites précédemment peuvent être reprises. On constate avant tout ici une grande robustesse de l'algorithme de modélisation. Le point important qu'est le maintien de la cohérence de la carte de l'environnement est ici correctement géré. Sur un trajet important, et donc sur une zone d'exploration importante, on constate qu'il n'y a pas de génération en surnombre de primitives cartographiques. En outre, on constate aussi que l'évolution des sous pavages caractérisant l'imprécision des amers de l'environnement est optimale au sens de leur convergence vers la position réelle et de leur diminution au fur et à mesure de l'apport d'informations sensorielles. Ces constats prouvent là encore la validité du module décisionnel permettant de gérer de façon robuste la fusion et l'intégration de données sensorielles.

Enfin l'interaction entre l'erreur sur la localisation et l'erreur commise sur les positions des amers étant prise en compte dans notre démarche de localisation et modélisation, on peut constater que sur le retour du robot, les sous pavages des amers diminuent de façon conséquente, et surtout du côté le plus proche du robot : la précision sur la localisation du robot augmentant au retour permet de minimiser la quantification de l'erreur d'observation. C'est cette minimisation de l'erreur de mesure qui permettra d'apporter de la précision sur l'estimation de la position de l'amer.

Plutôt qu'une alternative, l'approche ensembliste se veut ici être une solution permettant d'intégrer intrinsèquement la notion d'imprécision. Le fait de pouvoir gérer l'imprécision implique la possibilité de prendre en compte des interactions qui ne peuvent l'être avec le formalisme des moindres carrés récursifs. C'est cette gestion rigoureuse de ces interactions qui permet de mener à bien le processus de génération de carte sur des distances importantes.

Nous avons donc réalisé l'objectif que nous nous étions fixé au chapitre 3, qui consistait à développer une autre méthode de localisation/modélisation plus robuste, qui pourrait réaliser cette deuxième trajectoire dans son intégralité.

5.7. Conclusion

Nous avons dans ce chapitre développé une méthode de modélisation de l'environnement basée sur l'utilisation de la théorie ensembliste. La première étape a consisté à modifier l'algorithme de localisation utilisé dans le chapitre précédent pour utiliser des primitives cartographiques différentes. En effet, chaque amer modélisé est dans cette approche un sous pavage.

Nous avons ensuite développé une méthode de décision basée sur l'utilisation des règles de Dempster-Shafer, permettant de gérer une croyance sur différents cas de figures pouvant se présenter dans un processus de génération de carte à partir de chaque observation : la fusion avec un nouvel élément de la carte, l'insertion d'un nouvel amer, ou le rejet de cette observation. Cette méthode est basée sur le pourcentage d'intersection entre le sous pavage de l'observation et le sous pavage représentant les éléments de la carte.

Cette méthode de décision nous a permis d'aborder l'étape suivante qui consiste à intégrer ces nouvelles informations à celles présentes dans la carte. Dans le cas où une fusion est réalisée, la stratégie retenue est la diminution du sous pavage associé à la primitive cartographiée considérée. Cette technique permet un traitement rapide mais nécessite préalablement que tous les éléments soient insérés sous forme de sous pavages réduits au minimum, c'est-à-dire que la taille de chaque pavé soit inférieure à la précision souhaitée.

Lorsqu'une observation doit être insérée comme nouvelle primitive cartographique, nous calculons un sous pavage minimal englobant l'ensemble des pavés de l'observation, issus de la localisation et des erreurs de mesures.

Finalement, un traitement particulier est appliqué aux informations n'ayant pas été observées de façon fréquente, nous considérons en effet que si un élément de l'environnement est par exemple observé une fois, et qu'il ne l'est plus dans le reste des acquisitions, nous pouvons le supprimer de la carte. Ceci nous permet de supprimer la majorité des données aberrantes.

Nous avons testé cette méthode dans deux environnements de référence, pour valider notre approche, mais aussi pour comparer les résultats avec ceux obtenus dans le chapitre 3. Nous avons constaté que la méthode développée fonctionne bien, en effet, le paradigme a été validé sur une trajectoire réalisée dans un couloir de grande dimension sur la totalité des acquisitions. La précision sur la localisation et sur l'estimation de positions des amers est relativement correcte, et aucune dérive n'a été constatée.

Nous avons donc un système permettant simultanément de localiser le robot à partir d'une carte non fiable, et parallèlement, de modéliser de façon incrémentale l'environnement d'évolution du robot, de manière relativement précise. Ces deux étapes étant intimement liées, la qualité de l'une dépend de la précision de l'autre, ainsi l'utilisation de la théorie ensembliste nous a permis de répercuter l'ensemble des imprécisions introduites à chaque étape de notre méthode sur les phases suivantes.

Conclusion générale et perspectives

L'objectif de cette étude était la mise en œuvre d'un paradigme de localisation et modélisation simultanée de l'environnement. Pour ce faire il nous a fallu résoudre un certain nombre de problématiques dont celle de la localisation. Mais le préalable aux étapes de localisation et de modélisation est la construction d'un modèle sensoriel robuste de l'environnement. Il nous a donc fallu choisir un système de perception permettant de gérer des primitives sensorielles les plus pertinentes possibles. Ce choix s'est porté naturellement sur un capteur de vision omnidirectionnelle pour deux raisons : ce capteur a la particularité de fournir des données sur 360 degrés en une seule acquisition et nous pouvons de plus nous appuyer sur l'expérience acquise depuis plusieurs années sur ce type de système par notre laboratoire. Toutefois, les observations obtenues sont de type goniométrique puisque correspondant aux amers de l'environnement. Afin de minimiser la combinatoire liée à la phase de mise en correspondance nous avons décidé de coupler deux acquisitions omnidirectionnelles pour obtenir sur une acquisition la distance des amers verticaux par rapport au robot.

Pour construire notre modèle sensoriel stéréoscopique de l'environnement, il nous a fallu développer des méthodes d'extraction de droites radiales basées sur un filtrage de type gradient. Nous avons affiné l'étape d'extraction avec un post-traitement intégrant un système de « vote ». L'étape suivante a consisté à mettre en correspondance les secteurs de niveaux de gris sur les deux images omnidirectionnelles. Nous avons pour cela développé une méthode originale d'appariement qui se propose de fusionner plusieurs critères de ressemblance portant sur les niveaux de gris et sur des contraintes géométriques. Ces critères sont fusionnés par la théorie des croyances de Dempster-Shafer. Ce traitement fournit une mise en correspondance robuste entre les secteurs de niveaux de gris des deux images. Cet appariement permet d'obtenir par triangulation les coordonnées des amers de l'environnement. La précision et la robustesse du modèle sensoriel ainsi obtenu ont été valorisées dans l'étape de localisation.

Nous avons ramené le problème de localisation à une mise en correspondance de deux ensembles de points : les points observés avec ceux répertoriés dans la carte de l'environnement. Plusieurs méthodes permettent de résoudre ce problème d'appariement relativement classique. Nous avons utilisé dans un premier temps la distance de Hausdorff comme critère de ressemblance. Après avoir conclu à l'inadéquation de ce critère par rapport à notre problème, nous avons cherché à proposer une approche originale basée sur la minimisation de la distance cartésienne cumulée entre les points observés et les amers de l'environnement. Les premiers essais de localisation absolue réalisés dans des environnements structurés ont fourni des résultats très satisfaisants aussi bien en terme de précision qu'en terme de robustesse. La combinatoire de notre algorithme étant relativement forte, nous avons dans un deuxième temps cherché à l'optimiser en utilisant l'information odométrique pour restreindre l'espace de recherche. En outre ce fenêtrage permet de lever des ambiguïtés sur des estimations de configuration non cohérentes.

Les résultats probants obtenus en localisation nous ont permis d'étendre notre méthode au cas de figure où le robot n'a pas de carte a priori de l'environnement. Dans la deuxième partie des travaux, un module de génération de carte a donc été greffé à celui de localisation. Pour ce

faire, il nous a fallu intégrer une phase de mise à jour incrémentale de la carte après chaque localisation. Nous avons montré l'importance de choisir un critère de fusion et d'intégration de données le plus robuste possible : c'est lui qui conditionnera le maintien de la cohérence de la carte. Il nous a alors semblé judicieux de prendre en considération les deux angles d'observation d'un amer, angles fournis par les deux capteurs omnidirectionnels. Une méthode de décision a donc été élaborée et nous a permis d'établir des croyances sur la fusion d'une observation avec une primitive cartographique ou sur l'intégration et l'initialisation d'un nouvel élément. Cette méthode est basée sur la généralisation des règles de combinaison de Dempster-Shafer. Suite à cette phase de prise de décision, il nous fallu traiter le problème de la mise à jour des paramètres des primitives cartographiques. Dans un premier temps, la méthode choisie s'est portée vers une estimation incrémentale au sens des moindres carrés des coordonnées des points de la carte. Les observations n'étant pas disponibles au même instant, nous avons utilisé les moindres carrés récursifs. Nous avons affiné notre représentation cartographique par une prise en compte des liens existants entre chaque amer verticaux. L'information de lien entre deux points nous est fournie par la notion de secteur de niveaux de gris gérée dans la phase de segmentation. Le protocole expérimental permettant de valider notre paradigme de localisation et de modélisation incrémentale de carte a été le même durant toute cette étude. Nous avons considéré un environnement de type hall de taille réduite, et un environnement de taille plus importante, propre à mettre en évidence les phénomènes de dérive. En outre, nous avons intégré dans ce protocole une trajectoire de type aller et retour soulignant la cohérence de la représentation lorsque le robot revient sur une zone déjà modélisée. Deux constats ont été faits sur notre première approche de génération incrémentale de carte. Le premier est que les résultats obtenus en terme de précision et de robustesse sont bons sur un environnement de petite taille, mais plus discutable sur un déplacement important puisque des phénomènes d'accumulation d'erreurs ont pu être observés. La position finale du robot calculée à la fin de la série d'acquisition se trouve être alors très loin de la position réelle. La représentation de la carte sur le retour n'est plus cohérente. Une des causes prépondérantes expliquant cette forte dérive est l'absence totale de prise en compte de la quantification des erreurs entachant d'une part l'estimation de configuration du robot et d'autre part l'estimation de position des différentes primitives cartographique. En outre, l'imprécision n'étant pas prise en charge, l'interaction entre erreur sur la localisation et erreur sur les paramètres des primitives cartographiques ne l'a pas été non plus. Or cette interaction est un point crucial dans une démarche de localisation et de modélisation simultanée de l'environnement : l'erreur de localisation se répercute inévitablement sur celle des amers de l'environnement.

Fort de ce constat, nous avons cherché à intégrer l'information d'imprécision dans toutes les phases de notre paradigme de localisation et modélisation de l'environnement. Plusieurs solutions étaient envisageables et notre choix s'est finalement porté sur le formalisme à erreurs bornées fondé sur l'analyse par intervalle. Cette méthode possède le gros avantage de garantir « l'ensemble solution » cherché. Nous avons donc modifié notre module de génération incrémentale de carte, en commençant par l'étape de localisation. La gestion de l'imprécision par l'inversion ensembliste passe par le choix d'une représentation de l'erreur de mesure. Nous avons donc proposé et comparé plusieurs modèles d'erreurs pour finalement intégrer celui correspondant à des mesures télémétriques : il s'agit du modèle utilisé par Kieffer et qui est fonction des deux paramètres que sont l'angle et la distance. Le sous pavage de localisation est obtenu par inversion ensembliste et sa détermination basée sur l'utilisation de l'algorithme SIVIA. Nous avons bien sûr testé l'algorithme de localisation modifié au sens ensembliste sur les mêmes environnements que ceux utilisés pour la première méthode. Ces tests ont mis en évidence deux points prépondérants. En premier lieu l'estimation de la

configuration du robot est robuste, un taux d'échec nul ayant été constaté par rapport à l'ensemble des acquisitions. D'autre part, la quantification de l'imprécision est cohérente puisque englobant systématiquement la position réelle.

Nous avons étendu l'approche ensembliste à la problématique de génération incrémentale de carte, le robot dans ce cas naviguant sans connaissance a priori sur l'environnement. Le choix de la représentation de chaque amer au sein de la carte s'est porté logiquement et naturellement sur un sous pavage. Dans un premier temps, la procédure de localisation précédente a été modifiée pour tenir compte de la particularité de cette représentation. Dans un deuxième temps, nous avons élaboré un nouveau modèle de décision. Le pourcentage d'intersection entre pavés s'est imposé comme critère robuste permettant de discriminer les deux cas de figure que sont la fusion d'une observation avec une primitive et l'initialisation d'une nouvelle primitive. La détermination du critère décisionnel s'est faite pour éviter l'explosion combinatoire due à une insertion cumulative de primitives erronées. Nous avons ensuite proposé une méthode de mise à jour incrémentale des primitives cartographiques, basée sur les intersections entre pavés observés et pavés de la carte. Dans cette optique le problème d'insertion de nouveaux éléments a été résolu. Nous avons bien sûr testé notre paradigme de localisation et de modélisation simultanée au sens ensembliste sur les mêmes séries d'acquisition que celles utilisées dans la première approche. Les résultats expérimentaux obtenus ont mis en évidence la robustesse de notre processus de localisation et modélisation. Le maintien de la cohérence de la carte est correctement réalisé et les phénomènes d'erreurs cumulatives de la première approche ont été supprimés. La carte finale de l'environnement est très proche de la « vérité terrain », et la quantification des imprécisions est cohérente puisque englobant quasi-systématiquement les positions réelles des amers.

Nous avons, à travers cette étude, apporté une solution à la problématique de localisation et modélisation simultanée dans un environnement totalement inconnu. Nous avons mis en évidence expérimentalement la prépondérance de l'interaction entre erreur de localisation et erreur sur les primitives cartographiques. Finalement, nous avons proposé un paradigme de localisation et modélisation simultanée, d'une part original et d'autre part robuste, ce grâce à l'association d'un système de perception efficace au formalisme à erreur bornée.

Plusieurs points pourront être améliorés dans le futur. La première optimisation concerne l'amélioration de la qualité des données sensorielles fournies par notre système omnidirectionnel. Pour ce faire, une approche d'appariement des deux images traitant directement le signal de niveau de gris pourrait être envisagée : une méthode basée sur la corrélation de deux signaux pourrait par exemple être utilisée. L'amélioration permettrait de mettre en correspondance directement des variations de niveaux de gris caractéristiques associés aux amers verticaux de l'environnement.

Un autre point pourrait porter sur l'utilisation de deux capteurs omnidirectionnels pour obtenir les deux images, plutôt que la solution actuelle qui consiste à en faire translater un unique le long d'un rail. Cette nouvelle architecture engendrerait une zone d'occultation sur les deux images mais minime par rapport aux capteurs existants actuellement. En effet si l'on considère deux capteurs coniques de 5 cm de rayon situés à une distance identique à celle actuellement gérée, à savoir 40 cm, cela nous donnerait une occultation d'environ 14° sur chaque image, ce qui n'impliquerait pas une perte majeure d'informations. La nouvelle implantation pourrait nous apporter les avantages suivants :

- L'obtention des deux images en une seule acquisition, donc un gain de temps majeur,

- Une disparition des parties mécaniques, du moteur pas à pas et de la partie commande de moteur, d'où un encombrement moindre,
- Une meilleure précision sur la distance entre les deux images : même si le capteur que nous utilisons est déplacé le long d'un rail, il est possible d'avoir des fluctuations sur les différentes acquisitions. Les jeux mécaniques et le positionnement du capteur en fin de course en sont les causes majeures.
- Grâce à une fréquence d'acquisition obligatoirement plus importante, et à une synchronisation quasi parfaite entre les deux acquisitions, un suivi d'amer (tracking) pourrait être possible. Une solution pourrait consister suivre un amer sur une image et à vérifier l'hypothèse de suivi sur l'autre image.

Par rapport à l'application du formalisme à erreur bornée, une amélioration pourrait être apportée à l'ensemble des fonctions d'inclusions utilisées lors du calcul des différents pavages qui sont des fonctions d'inclusion naturelles. Leur particularité est de ne pas être minimales dans notre cas à cause des occurrences multiples de certaines variables. Une optimisation possible consisterait à utiliser par exemple la forme centrée standard [Moore 79].

Par rapport au critère de décision qui conditionne l'intégration d'une nouvelle primitive cartographique ou la fusion d'une observation avec une primitive déjà existante, deux points peuvent faire l'objet d'une amélioration. Le premier point porte sur le nombre de critères intégrés dans le processus décisionnel qu'il serait intéressant d'augmenter. Dans notre approche seul le critère du pourcentage d'intersection est pris en compte, mais l'occurrence d'une primitive, sa proximité avec des voisins sont des exemples de paramètres qui pourraient être gérés à ce niveau dans une approche multi-critère. Le deuxième point concerne le calcul du pourcentage d'intersection entre sous-pavages, qui passe actuellement par un redécoupage avec SIVIA. Une solution plus optimale consisterait à calculer directement une approximation plus juste de ce pourcentage.

Enfin, les perspectives de notre approche sont multiples. Deux pourraient par exemple être dégagées à court terme. La première concerne l'intégration de ces travaux dans un projet de développement d'un système d'aide à la conduite. Une partie intéressante d'un tel projet serait de pouvoir détecter l'arrivée d'un danger proche et pouvoir en avertir le conducteur. On appelle ici danger proche, l'implication que peut avoir un ou plusieurs véhicules avoisinant sur une action entreprise par le conducteur, un changement de file par exemple. La modélisation de l'environnement proche du véhicule considérée pourrait s'avérer prépondérante. Un projet régional allant dans ce sens est actuellement géré par notre laboratoire.

Une autre perspective résiderait dans l'intégration d'une architecture de propagation d'incertitudes dans notre paradigme de génération incrémentale de carte. Cette architecture a été récemment développée dans notre laboratoire. L'incertitude répercutée non seulement sur les observations, mais aussi sur les primitives de la carte, pourrait aisément nous renseigner sur la « fiabilité » de cette primitive, donc sur le poids à lui accorder.

Références bibliographiques

- [ADE91] E. Adelson and J. Bergen – *"The plenoptic function and element of early vision"* – In LANDY and MOVSHON, editors, *Computational Models of Visual Processing*, Chap I, MIT Press, 1991.
- [ARM98] J.M. Armingol, L. Moreno, A. de la Escalera et M.A. Salichs – *"Landmark perception planning for mobile robot localization"* – IEEE International Conference on Robotics and Automation ICRA'98, Leuven, Belgique, mai 1998.
- [ARN93] P. Arnould – *"Etude de la localisation d'un robot mobile par fusion de données"* – Thèse de doctorat de l'Institut National Polytechnique de Lorraine, Nancy, septembre 1993.
- [ARR99] K.O. Arras et N. Tomatis – *"Improving robustness and precision in mobile robot localization by using laser range finder and monocular vision"* – Third european workshop on advanced mobile robots Eurobot'99, septembre 1999.
- [ARS98] A. Arsenio et M.I. Ribeiro – *"Computational geometry & optimal filtering for natural landmark based mobile robot localization"* – IEEE International Conference on Intelligent Robotic Systems, Edinburgh, Ecosse, Juillet 1998.
- [ATI93] S. Atiya and G.D. Hager – *"Real-time vision-based robot localization"* – IEEE Trans. on Robotics and Automation, Vol. 9, No. 6, pp.785-800, 1993.
- [BAK98] S. Baker and S. Nayar – *"A theory of catadioptric image formation"* – Proceeding of the 6th International Conference on Computer Vision, Bombay, India, pages 35-42, January 1998.
- [BAR96] M. Barth and C. Barrows – *"A fast panoramic imaging system and intelligent imaging technique for mobile robots"* – Proceedings of International Conference on Intelligent RObots and Systems, Osaka, Japan, pages 626-633, November 1996.
- [BEN96] R. Benosman, T. Maniere and J. Devars – *"Multidirectional stereovision sensor, calibration and scenes reconstruction"* – In International Conference on Pattern Recognition, Vienna, Austria, pp. 161-165, September 1996.
- [BID95] E. Bideaux – *"STAN : Système de transport à apprentissage neuronal. Application de la vision omnidirectionnelle à la localisation d'un robot mobile autonome"* – Thèse de Doctorat de l'Université de Besançon, janvier 1995.
- [BLA00] Blanco, D. Boada, B.L. Moreno, L. Salichs, M.A. – *"Local Mapping from On-line Laser Voronoi Extraction"* – Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2000, Takamatsu, Japon, novembre 2000.
- [BOG95] S. Bogner – *"Introduction to panoramic imaging"* – Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 3099-3106, 1995.
- [BON98] P. Bonnifait et G. Garcia – *"Design and experimental validation of an odometric and goniometric localization system for outdoor robot vehicles"* – IEEE Transactions on Robotics and Automation, vol. 14, n. 4, p. 541-548, Août 1998.
- [BOR91] J. Borenstein, Y. Koren – *"Histogrammic in-motion mapping for mobile robot obstacle avoidance"* – IEEE Trans. on robotics and automation, Vol. 7, N°4, pp. 1688-1693, August 1991.
- [BOR95] J. Borenstein – *"Internal correction of dead-reckoning errors with a dual-drive compliant linkage mobile robot"* – Journal of Robotic systems , Vol. 12, n°4, 1995, p. 257-273.
- [BOR96]. J. Borenstein, L. Feng – *"Gyrodometry : a new method for combining data from gyros and odometry in mobile robots"* – Proc. of the IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota - April 1996.
- [BOR97] J. Borestein, H.R. Everett, L. Feng et D. Wehe – *"Mobile robot positioning - Sensors and techniques"* – Journal of Robotic Systems, vol. 14, n. 4, p. 231-249, 1997.
- [BRA95] E. Brassart – *"Localisation absolue d'un robot mobile autonome par des balises actives et un système de vision monoculaire"* – Thèse de Doctorat de l'Université de Compiègne, Janvier 1995.
- [BUR96] W. Burgard, D. Fox, D. Hennig et T. Schmidt – *"Estimating the Absolute Position of a mobile robot using position probability grids"* – Proc. of the 14th National Conference on Artificial Intelligence AAAI'96.
- [BUR99] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun – *"Sonar-based mapping of large-scale mobile robot environments using EM"* – In Proc. of the International Conference on Machine Learning (ICML' 99), 1999.
- [CAO86] Z.L. Cao, S.J. Oh and E.L. Hall – *"Dynamic Omnidirectionnal Vision for Mobile Robots"* – Journal of Robotic System, 3(1), pages 5-17, 1986.

- [CAU00] C. Cauchois, E. Brassart, L. Delahoche, T. Delhommelle – "*Reconstruction with the Calibrated SYCLOP Sensor*" – Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS'00), October, November 2000, Kagawa University, Takamatsu, Japan, p1493-1498.
- [CAU01b] C. Cauchois – "*Modélisation et Calibration du Capteur Omnidirectionnel SYCLOP : Application à la Localisation Absolue en Milieu Structuré*" – Thèse de doctorat, Université de Picardie Jules Verne, Amiens, 2001.
- [CAU99a] C. Cauchois, E. Brassart, C. Drocourt, P. Vasseur – "*Calibration of the omnidirectional vision sensor : SYCLOP*" – Proc. of the IEEE International Conference on Robotics and Automation (ICRA 99), May 1999, p. 1287 – 1292.
- [CAU99b] C. Cauchois, E. Brassart, C. Pegard, A. Clerentin – "*Technique for Calibrating an Omnidirectional Sensor*" – Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS'99), october 1999, p. 166-171.
- [CHO95] H. Choset and J. Burdick – "*Sensor Based Planning, Part I: The Generalized Voronoi Graph*" – Proceedings of the 1995 IEEE International Conference on Robotics and Automation (ICRA '95), Vol. 2, pp. 1649 - 1655 May 1995.
- [CHO96] H. Choset – "*Sensor Based Motion Planning : The Hierarchical Generalized Voronoi Graph*" – PhD thesis, California Institute of Technology, Pasadena, California, March 1996.
- [CHU01] H. Chung, L. Ojeda et J. Borenstein – "*Sensor fusion for Mobile Robot Dead-reckoning With a Precision-calibrated Fiber Optic Gyroscope*" – Proc. of the 2001 IEEE International Conference on Robotics & Automation, pp.3588-3593, Séoul, Corée, 21-26 mai 2001.
- [CLE01a] A. Clerentin, L. Delahoche, E. Brassart, C. Pegard – "*Omnidirectional sensors cooperation for multi-target tracking*" – IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI 2001), Baden-Baden, August 20-22, 200
- [CLE01b] A. Clémentin – "*Localisation d'un robot mobile par coopération multi-capteurs et suivi multi-cibles*" – Thèse de doctorat, Université de Picardie Jules Verne, Amiens, 2001.
- [COX95] I.J. Cox et M.L. Miller – "*On finding ranked assignments with application to multi target tracking and motion correspondance*" – IEEE trans. on Aerospace and Electronic Systems, vol. 32, n. 1, pp. 486-495.
- [COX96] I.J. Cox and S.L. Hingorani – "*An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and its Evaluation for the Purpose of Visual Tracking*" – IEEE Trans. On PAMI, vol. 18, no. 2, pp. 138-150, Feb. 1996.
- [CRO89] J. Crowley – "*World modelling and position estimation for a mobile robot using ultrasonic ranging*" – Proc. of IEEE Conference on Robotics and Automation, Scottsdale, , p. 674-680, May 1989.
- [CUI01] Y.J. Cui et S.S. Ge – "*Autonomous Vehicle Positioning with GPS in Urban Canyon Environments*" – Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Séoul, Corée, 21-26 mai 2001
- [DEA00] M. Deans, M. Hebert – "*Invariant filtering for simultaneous localization and mapping*" – Proc. IEEE International Conference on Robotics and Automation, pp. 1042-1047, April 2000.
- [DEL97a] L. Delahoche – "*Coopération multi-capteurs appliquée à la localisation des robots mobiles*" – Thèse de doctorat de l'université de Picardie Jules Verne, juillet 1997.
- [DEL97b] L. Delahoche, C. Pégard, B. Marhic, P. Vasseur – "*A navigation system based on an omnidirectional vision sensor*" – Proceedings on IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97), Grenoble, France, Septembre 1997.
- [DEL98] L. Delahoche, C. Pégard, M. Mouaddib, P. Vasseur – "*Incremental map building for mobile robot navigation in an indoor environment*" – Proceedings of International Conf. on Robotics and Automation (ICRA'98), Belgium, May 1998.
- [DEM67] A. P. Dempster – "*Upper and lower probabilities induced by a multi-valued mapping*" – Annals of Mathematical Statistics, vol. 38, 1967.
- [DIS00] G. Dissanayake, HG. Durrant-Whyte, T. Bailey – "*A computationally efficient solution to the Simultaneous Localization and Map Building (SLAM) problem*" – Proc. Of International Conference on Robotics and Automation, ICRA'00, pp. 1009-1014, May 2000.
- [DRU87] M. Drumheller – "*Mobile robot localization using sonar*" – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, n°2, 1987, p. 325-332.
- [DUD00] G. Dudek, D. Jugessur – "*Robust Place Recognition using Local Appearance based Methods*" – IEEE International Conference on Robotics and Automation, 2000, pp 1030-1035.
- [DUD97] G. Dudek, M. Jenkin, E. Milios et D. Wilkes – "*Map validation and robot self-location in a graph-like world*" – Robotic and Autonomous System, vol. 22(2), novembre 1997.
- [DUR95] H. Durrant-Whyte – "*The design of a radar-based Navigation System for large outdoor vehicles*" – Proc. IEEE Int. Conf. on Robotics and Automation, Nagoya, Aichi, Japan, , p. 764-769, May 1995.

- [ELF87] A. Elfes – "*Sonar-based real world mapping and navigation*" – IEEE Journal of robotics and automation, Vol. RA-3, N°3, pp. 249-265, June 1987.
- [ELK94] R.T. Elkins and E.L. Hall – "*Three dimensional line following using omnidirectional vision*" – Proc. SPIE Intelligent Robots and Computer Vision XIII, vol. 2354, pp 130-144, 1994
- [FIL94] P. Fillatreau – "*Localisation et modélisation tridimensionnelle pour un robot mobile autonome tout terrain*" – Thèse de doctorat de l'Université Paul Sabatier de Toulouse, octobre 1994.
- [FOF97] D. Fofi – "*Détection d'obstacles par perception visuelle pour un robot mobile*" – Rapport de DEA, Ecole Nationale Supérieure de l'Electronique et de ses Applications, Université de Cergy-Pontoise, Août 1997.
- [FOX99] D. Fox, W. Burgard, S. Thrun – "*Probabilistic Methods for Mobile Robot Mapping*" – Proc. of the IJCAI-99 Workshop on Adaptive Spatial Representations of Dynamic Environments, 1999.
- [FRA90] G. Frappier – "*Système inertiels de navigation pour robots mobiles*" – Séminaire "Les robots mobiles", EC2, Paris, 1990
- [GAS99] J. Gasos et A. Saffiotti – "*Integrating fuzzy geometric maps and topological maps for robot navigation*" – Proc. of the 3rd International ICSC Symposium on Soft Computing (SOCO'99) Gène, Italie, Juin 1999.
- [GON94] J. Gonzalez, A. Ollero, A. Reiner – "*Map building for a mobile robot equipped with a 2D laser rangefinder*" – Proc. IEEE Int. Conf. on Robotics and Automation, San Diego, p. 1904-1909, May 1994.
- [GRU99a] D. Gruyer – "*Etude du traitement de données imparfaites pour le suivi multi-objets : Application aux situations routières*" – Thèse de doctorat, Université de Technologie de Compiègne, 1999.
- [GRU99b] D. Gruyer, V. Berge-Cherfaoui – "*Matching and decision for Vehicle tracking in road situation*" – IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'99, Kyongju, Corée, 17-21 octobre 1999.
- [GUI00a] J. Guivant, E. Nebot, HF Durrant-Whyte – "*Simultaneous localization and map building using natural features in outdoor environments*" – IAS-6 Intelligent Autonomous Systems 25-28 July 2000, Italy - Intelligent Autonomous Systems 6, vol 1, pp 581-588, July 2000.
- [GUI00b] J. Guivant, E. Nebot, S. Baiker – "*Autonomous navigation and map building using laser range sensors in outdoor applications*" – Journal of Robotic Systems, Vol 17, n° 10, pp 565-283, October 2000.
- [HOL91] J. Hollingum – "*Caterpillar make the earth move : automatically*" – Industrial Robot, Vol. 18, N° 2, pp 15-18, 1991.
- [HON91] J. Hong, X. Tan, B. Pinette, R. Weiss and E. Riseman – "*Image-based homing*" – In Proceedings of IEEE International Conference on Robotics and Automation, pp. 620-625, 1991.
- [HUT93] D.P. Huttenlocher, G.A. Klanderma, W.J. Rucklidge – "*Comparing Images Using the Hausdorff Distance*" – IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(9):850-863, 1993.
- [ISH90] H. Ishiguro, M. Yamamoto and S. Tsuji – "*Analysis of omnidirectional views at different location*" – In Proceedings of IEEE International Conference on Intelligent Robotics and Systems, pp 659-664, 1990.
- [ISH92] H. Ishiguro and S. Tsuji – "*Omni-Directional Stereo*" – IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2):257-262, 1992.
- [ISH98] H. Ishiguro – "*Development of Low-Cost Compact Omnidirectional Vision Sensors and Their Applications*" – Int Conf on Information Systems, Analysis and Synthesis, ISAS'98, pp. 433-439.
- [JAU94] L. Jaulin – "*Solution globale et garantie de problèmes ensemblistes; Application à l'estimation non linéaire et à la commande robuste*" – Thèse de doctorat, Université de Paris XI Orsay, 1994.
- [JAU97] L. Jaulin, and E. Walter – "*Global numerical approach to nonlinear discrete-time control*" – IEEE Trans. on Autom. Control, 42, 872-875 (1997).
- [JOU01] I.S. Joung et H.S. Cho – "*Self localization for mobile robots by matching of two consecutive environmental range data*" – Proceedings of the 2001 IEEE International Conference on Robotics & Automation, pp. 1603-1608, Séoul, Corée, 21-26 mai 2001
- [JOU99] S. Jouannin – "*Association et fusion de données : Application au suivi et à la localisation d'obstacles par radar à bord d'un véhicule routier intelligent*" – Thèse de doctorat de l'université Blaise Pascal de Clermont-Ferrand, janvier 1999.
- [KAN97] S. Kang and R. Szeliski – "*Scene data recovery using omnidirectional multibaseline stereo*" – International Journal of Computer Vision, 25(2):167-183, 1997.
- [KAW98] T. Kawanishi and K. Yamazawa and H. Iwasa and H. Takemura and N. Yokoya – "*Generation of high-resolution stereo panoramic images by omnidirectional sensor using hexagonal pyramidal mirrors*" – In 14th International Conference on Pattern Recognition, pp. 485-489, Brisbane, Australia, August 1998.

- [KEM93] K. Kemmotsu, T. Kanade – "*Uncertainty in object pose determination with three light-stripe range measurements*" – Proc. IEEE International Conference on Robotics and Automation, Atlanta (USA), pp. 128-134, May 1993.
- [KIE00] M. Kieffer, L. Jaulin, E. Walter and D. Meizel – "*Localisation et suivi robustes d'un robot mobile grâce à l'analyse par intervalles*" – Traitement du signal, numéro spécial sur la robotique: "fusion de données pour véhicules intelligents", volume 17, n° 3, 207-219.
- [KIE98] M. Kieffer, L. Jaulin and E. Walter – "*Guaranteed recursive nonlinear state estimation using interval Analysis*" – Proc. 37th IEEE Conference on Decision and Control, pp 3966-3971, Tampa, Florida, 16-18 décembre 1998.
- [KIE99] M. Kieffer – "*Estimation ensembliste par analyse par intervalles : Application à la localisation d'un véhicule*" – Thèse de doctorat, Université de Paris Sud Orsay, 1999.
- [KOR94a] D. Kortenkamp – "*Perception for mobile robot navigation: a survey of the state of the art*" – Proc. NASA Dual-use Space Technology Transfer Conf., 1994.
- [KOR94b] D. Kortenkamp et T. Weymouth – "*Topological mapping for mobile robot using a combinaison of sonar and vision sensing*" – AAI'94, 1994.
- [KOT98] S. Kotani, K. Kaneko, T. Shinoda and H. Mori – "*Mobile Robot Navigation Based on Vision and DGPS Information*" – Proc. Of International Conf. on Robotics and Automation (ICRA'98), Belgium, pp. 2524-2530, May 1998.
- [KRO95] E. Krotov, M. Hebert – "*Mapping and positioning for a prototype lunar rover*" – Proc. IEEE Int. Conf. on Robotics and Automation, Nagoya, Aichi, Japan, p. 2913-2919, May 1995.
- [KUI91] B. Kuipers, Y.T. Byun – "*A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations*" – Robotics and Autonomous Systems, 8 1991
- [LAB88] M. Labarrere, J.P. Krief, B. Gimonet – "*Le filtrage et ses applications*" – Cepadues Edition, 1988.
- [LAL98] A. Lallement, M. Dufaut et R. Husson – "*Sensor cooperation and world perception of the autonomous mobile robot Romane*" – Third IFAC Symposium on Intelligent Autonomous Vehicles, Madrid, Espagne, Mars 1998.
- [LEO92] J. Leonard, H.F. Durrant-Whyte, J. Cox – "*Dynamic map building for an autonomous mobile robot*" – The International Journal of Robotics Research, Vol. 11, n° 4, p 286-298, August 1992.
- [LEV98] O. Leveque – "*Méthodes ensemblistes pour la localisation de véhicules*" – Thèse de doctorat, Université de Technologie de Compiègne, 1998.
- [LIT98] J.J. Little, J. Lu, and D.R. Murray – "*Selecting Stable Image Features for Robot Localization Using Stereo*" – Proc. on IEEE/RSJ International Conference on Intelligent Robotic Systems. Victoria, Canada, October 1998.
- [LOW01] S. Se, D. Lowe, J. Little – "*Vision-based Mobile Robot Localization And Mapping using Scale-Invariant Features*" – Proc. IEEE International Conference on Robotics and Automation, Seoul, Korea, pp.2051-2058, May 2001.
- [MAH98] R. Mahkovic, T. Slivnik – "*Generalized Local Voronoi Diagram of Visible Region*" – Proc. Of International Conf. on Robotics and Automation (ICRA'98), Belgium, pp. 3437-3443, May 1998.
- [MAK91] H. Makela, K. Koskinen – "*Navigation of outdoor mobile robots using dead reckoning and visually detected landmarks*" – 5th Int. Conf. on Advanced Robotics ICAR'91, Pisa, Italy, June 1991, pp 1051-1056.
- [MAL01] A. Mallet – "*Localisation d'un robot mobile autonome en environnements naturels*" – Thèse de l'Institut National Polytechnique de Toulouse, LAAS, juillet 2001.
- [MAR99] B. Marhic – "*Localisation Absolue de Robots Mobiles par le Capteur de Vision Omnidirectionnelle SYCLOP*" – Thèse de doctorat, Université de Picardie Jules Verne, Amiens, Décembre 1999.
- [MAT00] T. Matui, H. Asoh, S. Thompson – "*Mobile Robot Localization Using Circular Correlations of Panoramic Images*" – Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2000, Takamatsu, Japon, novembre 2000.
- [MAT95a] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe et T. Litwin – "*Mars microrover navigation : performance evaluation and enhancement*" – Autonomous Robot Journal, special issue on autonomous vehicle for planetary exploration, vol. 2, n. 4, 1995.
- [MAT95b] B.O. Matthews, D. Perdue and E. L. Hall – "*Omnidirectional vision applications for line following*" – In Proc. SPIE Intelligent Robots and Computer Vision XIV, vol. 2588, pp 438-449, 1995.
- [MAY97] M. May – "*Introduction à l'optique - cours et problèmes résolus*" – Ed. Dunod, 1997.
- [MOO79] R.E. Moore – "*Methods and Applications of Interval Analysis*" – Philadelphia, SIAM Publ., 1979.
- [MOR00] K. Moriya and K. Ohba – "*Image Projection Criteria with the Epipolar Line for Precise Correlation with Omni Images*" – In Proceedings of the 4th Asian Conference on Computer Vision, Taipei, Taiwan, January 2000.

- [MOR88] H.P. Moravec – "*Sensor fusion in certainty grids for mobile robots*" – AI Magazine, Summer 1988.
- [MOR89] T. Morita and Y. Inamoto and T. Uchiyama and Y. Yasukawa and S. Kawakami – "*Measurement in three dimensions by motion stereo and spherical mapping*" – In Proc. of Computer Vision and Pattern Recognition, pp 422-428, 1989.
- [MOU91] P. Moutarlier – "*Modélisation autonome de l'environnement par un robot mobile*" – Thèse de doctorat de l'université Paul Sabatier de Toulouse, octobre 1991.
- [MUR95] D.W. Murray – "*Recovering Range Using Virtual Multicamera Stereo*" – In Computer Vision and Image Understanding, 61(2):285-291, March 1995
- [NAG98] K. Nagatani, H. Choset and S. Thrun – "*Towards Exact Localization without Explicit Localization with the Generalized Voronoi Graph*" – Proc. Of International Conf. on Robotics and Automation (ICRA'98), Belgium, pp. 3437-3443, May 1998.
- [NAL96] V. Nalwa – "*Bell labs solid-state video invention, CCD, comes full circle*" – <http://www.lucent.com/press/0196/960103.bla.html>, January 1996.
- [NAY97] S.K Nayar – "*Omnidirectionnal Video Camera*" – In Proceeding of DARPA Image Understanding Workshop, New Orleans, USA, May 1997
- [NEI99] J. Neira, J. Tardos, J. Horn et G. Schmidt – "*Fusing range and intensity images for mobile robot localization*" – IEEE Transactions on Robotics and Automation, vol. 15, n. 1, Février 1999.
- [NEN98] S. Nene and S. Naya – "*Stereo with mirrors*" – In Proceedings International Conference on Computer Vision, Bombay, India, pages 1087-1094, January 1998.
- [OHY94] A. Ohya, E. Shoji et S. Yuta – "*3-D range sensor using fiber grating for recognition of autonomous mobile robots passage space*" – International Conference on Intelligent Robots and Systems, 1994.
- [OLS98] C.F. Olson et L.H. Matthies – "*Maximum likelihood rover localization by matching range maps*" – Proc. of the IEEE Int. Conf. On Robotics and Automation ICRA'98, p. 272-277, May 1998.
- [PAG98] D. Pagac, E. Nebot et H. Durrant-Whyte – "*An evidential approach to map-building for autonomous vehicles*" – IEEE Transactions on Robotic and Automation, vol. 14, n. 4, p. 623-629, août 1998.
- [PAP97] D. Papadopoulos Orfanos – "*Numérisation géométrique automatique à l'aide d'un capteur 3D de précision à profondeur de champ réduite*" – Thèse de doctorat de l'ENST, 1997.
- [PAR92] L. Parisot – "*Modélisation d'environnement et localisation d'un robot mobile à l'aide de capteurs télémétriques*" – Thèse de Doctorat de l'Université de Paris VI, novembre 1992.
- [PEG96] C. Pegard, M. Mouaddib – "*A mobile robot using a panoramic view*" – Proc. IEEE Int. Conf. On Robotics and Automation, Minneapolis, Minnesota, April 1996, pp. 89-94.
- [PEG99] C Pegard, E Mouaddib, B Mahric, L Delahoche, A Clerentin, E Brassart – "*Three localization methods for mobile robots using panoramic sensors*" – Proc. of WESIC'99, Newport, UK, pp. 51-59.
- [PER96] J-P. Perez – "*Optique- fondement et application*" – Collection Enseignement de la Physique, Ed. Masson, 1996.
- [PER99] J.A. Pérez, J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós – "*Continuous Mobile Robot Localization: Vision vs. Laser*" – Proc. Of International Conf. on Robotics and Automation (ICRA'99), Michigan, pp. 2917-2924, May 1999.
- [RIV99] P. Rives, J.L. Sequeira, P. Lourtie – "*Localization and map building for a mobile robot*" – International Symposium on Experimental Robotics, ISER'99, pp 225-234
- [ROM97] M. Rombaut, V. Berge-Cherfaoui – "*Decision making in data fusion using Dempster-Schafer's theory*" – 3th IFAC Symposium on Intelligent Components and Instrumentation for Control Applications, Annecy, France, 9-11 juin 1997.
- [ROM98] M. Rombaut – "*Decision in Multi-obstacle Matching Process using Theory of Belief*" – AVCS'98, Amiens, France, 1-3 juillet 1998.
- [RON86] J.J. Roning, Z.L. Cao, and E.L. Hall – "*Color target recognition using omnidirectional vision*" – SPIE Vol 728 Optics, Illumination, and Image Sensing for Machine Vision, 1986.
- [ROY01] C. Royere, D. Gruyer et V. Cherfaoui – "*Data association with believe theory*" – 3rd international conference on information fusion FUSION 2000, Paris, France, Juillet 2000.
- [RUP00] T. Rupp, P. Levi – "*Optimized Landmark Arrangement for Absolute Localization - A Practical Approach*" – Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2000, Takamatsu, Japon, pp. 448-453, novembre 2000.
- [SAB94] A. Sabatini, O. DI Benedetto – "*Towards a robust methodology for mobile robot localisation using sonar*" – Proc. IEEE Int. Conf. on Robotics and Automation, San Diego, May 1994, p. 3142-3147.

- [SAI86] A. R. de SAINT-VINCENT – "*Perception et Modélisation de l'Environnement d'un Robot Mobile : une Approche par Stéréovision*" – Thèse de l'Université P. Sabatier, Toulouse, France, Octobre 1986.
- [SAR89] K. Sarachik – "*Characterising an indoor environment with a mobile robot and uncalibrated stereo*" – In Proc. IEEE Conference on Robotics and Automation, pages 984-989, 1989.
- [SCH94] B. Schiele, J.L. Crowley – "*A comparison of position estimation techniques using occupancy grids*" – Proc. IEEE Int. Conf. on Robotics and Automation, San Diego, California, pp 1628-1634, May 1994.
- [SHA76] G. Shafer – "*A mathematical theory of evidence*" – Princeton : university press, 1976
- [SHA97] S. Shah and J. K. Aggarwal – "*Mobile robot navigation and scene modeling using stereo fish-eye lens system*" – Machine Vision and Applications, 10(4):159-173, 1997.
- [SIM96] K. Simsarian, T.J. Olson et N. Nandhakumar – "*View invariant regions and mobile robot self-localization*" – IEEE Transactions on Robotics and Automation, Octobre 1996.
- [SMA01] L. Smadja, R. Benosman and J. Devars – "*Determining epipolar constraint on cylindrical images and using it for 3D reconstruction*" – Proc. of the IEEE Workshop on "Omnidirectional Vision Applied to Robotic Orientation and Nondestructive Testing (NDT)" at 10th International Conference on Advanced Robotics (ICAR'01), Budapest, Hungary, pages 93-98, August 2001.
- [SME94] P. Smets et R. Kennes – "*The transferable belief model*" – Artificial Intelligence, vol. 66 n°2 pages 191-234, 1994.
- [SUK98] S. Sukkarieh, E.M. Nebot and H.F. Durrant-Whyte – "*Achieving Integrity in an INS/GPS Navigation Loop for Autonomous Land Vehicle Applications*" – Proc. Of International Conf. on Robotics and Automation (ICRA'98), Belgium, pp. 3437-3443, May 1998.
- [SVO98a] T. Svoboda, T. Padjla, and V. Hlavac – "*Epipolar Geometry of Panoramic Cameras*" – In 5th European Conference on Computer Vision, Freiburg, Germany, pages 218-231, July 1998.
- [SVO98b] T. Svoboda, T. Padjla, and V. Hlavac – "*Motion Estimation Using Central Panoramic Cameras*" – In Proceedings of IEEE Conference on Intelligent Vehicles, Stuttgart, Germany, pages 335-340, October 1998.
- [TAK95] J. Takeno, U. Rembold – "*Stereovision systems for autonomous mobile robots*" – Proc. Int. Conf. Intelligent Autonomous Systems, pp. 26-41, 1995.
- [THR00] S. Thrun, W. Burgard, D. Fox – "*A real time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping*" – IEEE International Conference on Robotics and Automation, 2000.
- [THR01] R. Thrapp, C. Westbrook et D. Subramanian – "*Robust localization algorithms for an autonomous campus tour guide*" – Proceedings of the 2001 IEEE International Conference on Robotics & Automation, pp. 2065-2071, Séoul, Corée, 21-26 mai 2001.
- [THR98] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers – "*Integrating topological and metric maps for mobile robot navigation: A statistical approach*" – In Proceedings of the 15th National Conference of the American Association for Artificial Intelligence (AAAI' 98), July 1998.
- [THR99] S. Thrun – "*Learning metric topological maps for indoor mobile robot navigation*" – Artificial Intelligence 99(1), pp 21-71.
- [ULR00] I. Ulrich, I. Nourbaskhsh – "*Appearance based Place Recognition for Topological Localization*" – IEEE International Conference on Robotics and Automation, 2000, pp 1023-1029.
- [VAG93] J. Vaganay – "*Conception d'un système multisensoriel de localisation dynamique 3D pour robot mobile*" – Thèse de doctorat, LIRMM, Montpellier, juillet 1993.
- [VAN01] D. Van Zwynsvoorde – "*Construction incrémentale de modèles topologiques pour la Navigation d'un robot mobile*" – thèse de l'Université Paul Sabatier, Décembre 2000.
- [VIC00] A. Victorino, P. Rives et J.J. Borrelly – "*Localisation d'un robot mobile et cartographie par télémétrie laser*" – Journées des Jeunes Chercheurs en Robotique, Bourges, France, Février 2000.
- [WAN90] M. Wang – "*Localisation Estimation and Analysis for Mobile Robots*" – In Autonomous Robot Vehicles, Cox I.J. et Wilfong, G.T. Editors, Springer Verlag, 1990
- [WEI94] G. Weiss, C. Wetzler et E. Von Puttkamer – "*Keeping track of position and orientation of moving indoor systems by correlation of range finder scans*" – IEEE int. conf. on intelligent robots and systems IROS'94, septembre 94, Munich, Allemagne, pp. 595-601
- [WIN99] N. Winters et J.S. Victor – "*Mobile robot navigation using omnidirectionnal vision*" – Proc of the third Irish Machine Vision and Image Processing Conf. (IMVIP'99), Dublin, Irlande, septembre 1999.
- [YAG90] Y. Yagi, S. Kawato – "*Panorama Scene Analysis with Conic Projection*" – IEEE International Workshop on Intelligent Robots and Systems, 1990, p 181-187.

- [YAG94] Y. Yagi, S. Kawato, T. Saburo – "*Real-Time Omnidirectional Image Sensor (COPIS) for Vision-Guided Navigation*" – IEEE Transactions on Robotics and Automation, Vol. 10, n°1, February 1994, p. 11-22.
- [YAG95] Y. Yagi, Y. Nishizawa, M. Yachida – "*Map-based navigation for a mobile robot with omnidirectional image sensor COPIS*" – IEEE Transactions on Robotics and Automation, Vol. 11, n°5, October 1995, p. 634-648.
- [YAM93] K. Yamazawa, Y. Yagi et M. Yachida – "*Omnidirectional imaging with hyperboloidal projection*" – Proc. Int. Conf. on Intelligent Robots and Systems, IROS, 1993.
- [ZHE90] J.Y. Zheng and S. Tsuji – "*Panoramic Representation for Route Recognition by a Mobile Robot*" – In Proc. of 10th International Conference on Pattern Recognition, pages 161-167, June 1990.

Bibliographie de l'auteur

Workshop

- C. Drocourt, L. Delahoche, E. Brassart, C. Cauchois. "Simultaneous localization and map building paradigm based on omnidirectional stereoscopic vision" - *Proc. of the IEEE Workshop on "Omnidirectional Vision Applied to Robotic Orientation and Nondestructive Testing (NDT)" at 10th International Conference on Advanced Robotics (ICAR'01)*, Budapest, Hungary, pages 73-79, August 22, 2001.
- C. Cauchois, E. Brassart, L. Delahoche, C. Drocourt. "Absolute localization with the calibrated SYCLOP sensor" - *Proc. of the IEEE Workshop on "Omnidirectional Vision Applied to Robotic Orientation and Nondestructive Testing (NDT)" at 10th International Conference on Advanced Robotics (ICAR'01)*, Budapest, Hungary, pages 65-72, August 22, 2001.
- E. Brassart, L. Delahoche, C. Cauchois, C. Drocourt, C. Pégard, E. M. Mouaddib. "Experimental results got with the omnidirectional vision sensor : SYCLOP" - *Proc. of the 1st IEEE Workshop on Omnidirectional Vision (OMNIVIS'00) at Computer Vision and Pattern Recognition (CVPR-2000)*, Hilton Head Island, South Carolina, USA, pages 145-152, June 12, 2000.

Congrès internationaux avec comité de lecture

- C. Drocourt, L. Delahoche, C. Pégard, C. Cauchois. "Localization method based on omnidirectional stereoscopic vision and dead-reckoning" - *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS'99)*, Korea, pages 960-965, October 1999.
- A. Clémentin, C. Pégard, C. Drocourt. "Environment Exploration Using an Active Vision Sensor" - *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS'99)*, Korea, pages 1525-1530, October 1999.
- C. Drocourt, L. Delahoche, C. Pégard, A. Clémentin. – "Mobile Robot Localization Based on an Omnidirectional Stereoscopic Vision Perception System" - *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, Michigan, USA, pages 1329 - 1334, May 1999.
- C. Cauchois, E. Brassart, C. Drocourt, P. Vasseur. – "Calibration of the omnidirectional vision sensor : SYCLOP" - *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, Michigan, USA, pages 1287 - 1292, May 1999.

Colloques nationaux et Groupe De Recherche

- C. Drocourt, L. Delahoche. "Localisation et construction de carte simultanées basées sur la vision omnidirectionnelle stéréoscopique" – *Journée organisée par l'OT5.1 du GDR/ISIS sur le thème "Vision omnidirectionnelle"*, le 7 Juillet 2000 à Amiens.
- C. Drocourt, L. Delahoche, A. Clémentin. "Perception de l'environnement et localisation par vision omnidirectionnelle stéréoscopique" - *10^{èmes} Journées des Jeunes Chercheurs en Robotique (JJCR'10)*, Amiens, pages 59-64, Novembre 1998.

LOCALISATION ET MODELISATION DE L'ENVIRONNEMENT D'UN ROBOT MOBILE PAR COOPERATION DE DEUX CAPTEURS OMNIDIRECTIONNELS

Résumé

Les travaux présentés dans cette thèse portent sur un paradigme de localisation et de modélisation simultanée de l'environnement (SLAM). Dans ce cadre, le système de perception utilisé est un système de vision omnidirectionnelle stéréoscopique basé sur la translation rigide du capteur SYCLOP développé notre laboratoire. La première problématique abordée est celle de la construction d'un modèle sensoriel robuste et cohérent de l'environnement à partir de deux images omnidirectionnelles. Une méthode originale d'association des secteurs issus de chaque image panoramique est proposée et utilise la « théorie des croyances » de Dempster-Shafer. Cette approche associative multi-critères permet par triangulation d'obtenir un modèle sensoriel où les amers verticaux sont représentés par des primitives de types points. La deuxième problématique traitée est celle la mise en correspondance du modèle sensoriel avec un modèle cartographié de l'environnement. Cette étape est prépondérante puisqu'elle permet au robot de se localiser par rapport à une carte pouvant être a priori connue ou construite de manière incrémentale. Les critères de robustesse et de précision nous ont conduits à valider et à utiliser un algorithme basé sur le calcul de la distance cartésienne. La troisième problématique abordée dans ces travaux, et la plus importante, est celle permettant de au robot de construire incrémentalement la carte de son environnement. Cette phase est indissociable de celle qui consiste à localiser le robot. On parle alors de paradigme de localisation et de modélisation simultanée de l'environnement. La phase de génération incrémentale de carte nécessite de s'intéresser aux problèmes que sont (1) le choix d'une représentation, (2) la distinction du cas où une observation est fusionnée avec une primitive cartographique de celui où l'intégration d'une nouvelle primitive est nécessaire, (3) la prise en compte de l'interaction entre erreur de localisation et erreur sur l'estimation des paramètres des primitives cartographiques. Nous avons proposé une première approche basée sur une estimation des paramètres des primitives cartographiques au sens des moindres carrés. Le critère décisionnel est géré avec la théorie des croyances de Dempster-Shafer. Nous montrons les limites de cette première approche qui engendre des phénomènes de dérive cumulative. Une des raisons expliquant cette dérive est l'absence de prise en compte de la troisième contrainte. Son intégration nous a conduit à proposer une deuxième approche du paradigme SLAM basée sur le formalisme de l'inversion ensembliste (analyse par intervalle). Après avoir reformulé le problème de la localisation au sens ensembliste, nous présentons un module de génération incrémentale de carte basé sur une représentation sous forme de sous-pavages. Les résultats obtenus sur des trajets importants montrent la minimisation des dérives ainsi qu'une précision importante.

Mots-clés : Robot mobile, Vision omnidirectionnelle, Localisation, Modélisation de l'environnement, Analyse par intervalles, Erreurs bornées.

Abstract

The work presented in this thesis concerns a Simultaneous Localization And Map Building (SLAM) paradigm. The perception system used is a stereoscopic omnidirectional vision system based on a rigid translation of the SYCLOP sensor developed in our laboratory. The first problematic we have treated concerns the environment sensorial model construction from two omnidirectional SYCLOP images. An original matching method between the sectors of the two images is presented. This method uses the Dempster-Shafer theory. This associative multi criteria approach enables to get by triangulation a sensorial model in which the vertical landmarks are represented by point primitives. The second problematic concerns the matching between the sensorial model and the cartographical model of the environment. This stage is preponderant since it enables the robot to localize itself from a model which can be either a priori known or incrementally built. The robustness and precision criteria forced us to use an algorithm based on the Cartesian distance computation. The third problematic, which is the most important, concerns the incremental construction of the robot's environment. This stage is linked to the localization: it is the notion Simultaneous Localization and Map Building (SLAM) paradigm. The incremental map construction obliges us to treat several problems: (1) the choice of a representation, (2) the distinction between the case where an observation is fused with a cartographical primitive and the case where a new primitive is integrated in the map, (3) the taking into account of the interaction between the localization and the error on the cartographical primitives. We have proposed two approaches which permit to optimally treat these constraints. The first method is a least square method. The decisional criterion is managed with the help of the Dempster-Shafer theory. We show the limits of this approach which generates a cumulative drift. One of the reason which explain this drift is the no taking into account of the third constraint. In order to integrate the interaction between the localization error and the modeling error, we have proposed a second approach of the SLAM paradigm based on interval analysis. The results obtained on important trajectories show the drift minimization and an important precision.

Keywords: Mobile robot, Omnidirectional vision, Localization, Map building, Interval analysis, Bounded errors.