



HAL
open science

Localisation d'un robot mobile: application à l'aide à la mobilité des personnes handicapées moteurs

Alain Courcelle

► **To cite this version:**

Alain Courcelle. Localisation d'un robot mobile: application à l'aide à la mobilité des personnes handicapées moteurs. Automatique / Robotique. Université Paul Verlaine - Metz, 2000. Français. NNT : 2000METZ004S . tel-01775408

HAL Id: tel-01775408

<https://hal.univ-lorraine.fr/tel-01775408>

Submitted on 24 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

UNIVERSITÉ DE METZ

THÈSE

présentée pour l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE METZ

Spécialité Automatique

par

Alain COURCELLE

LOCALISATION D'UN ROBOT MOBILE : APPLICATION À L'AIDE À LA MOBILITÉ DES PERSONNES HANDICAPÉES MOTEUR

Soutenue publiquement le 14 janvier 2000
devant la commission d'examen :

Président : **Tanneguy REDARCE**

Rapporteurs : **Michèle ROMBAUT**
Roland PRAJOUX

Examineurs : **Odile HORN**
Philippe MARTINET
Alain PRUSKI

BIBLIOTHEQUE UNIVERSITAIRE DE METZ



022 304926 6

6 162845 S/172

UNIVERSITÉ DE METZ

THÈSE

présentée pour l'obtention du

DOCTORAT DE L'UNIVERSITÉ

DE METZ

Spécialité Automatique

par

Alain COURCELLE

BIBLIOTHEQUE UNIVERSITAIRE - METZ	
N° inv.	2000 0745
Cote	S/173 00/4
Loc	Magasin

LOCALISATION D'UN ROBOT MOBILE :
APPLICATION À L'AIDE À LA MOBILITÉ DES
PERSONNES HANDICAPÉES MOTEUR

Soutenu publiquement le 14 janvier 2000
 devant la commission d'examen :

- Président : **Tanneguy REDARCE**
- Rapporteurs : **Michèle ROMBAUT**
Roland PRAJOUX
- Examineurs : **Odile HORN**
Philippe MARTINET
Alain PRUSKI

“ Etre conscient que l'on est ignorant est un grand pas vers le savoir ”

(Benjamin Disraeli, 1804-1881)

*à Karine
à l'enfant qu'elle porte
à Maëva
à mes Parents
à Eliane*

AVANT PROPOS

Ce mémoire de thèse présente la part la plus importante des travaux que j'ai effectués au Laboratoire d'Automatique des Systèmes Coopératifs (L.A.S.C) de l'Université de METZ.

Je remercie Monsieur Alain PRUSKI, Professeur à l'Université de METZ, mon directeur de thèse, de m'avoir accueilli dans son laboratoire.

J'adresse mes sincères remerciements à Madame Odile HORN, Maître de Conférences à l'Université de METZ, ma responsable de recherche, pour l'intérêt constant porté à mon travail, pour m'avoir fait bénéficier de ses conseils et de sa rigueur scientifique.

Je remercie Monsieur Tanneguy REDARCE, Professeur à l'Institut National des Sciences Appliquées de LYON d'avoir accepté la présidence de mon jury.

Je tiens à exprimer ma reconnaissance à Madame Michèle ROMBAUT, Professeur à l'IUT de TROYES, ainsi qu'à Monsieur Roland PRAJOUX, Directeur de Recherche CNRS au LAAS-CNRS de Toulouse, qui m'ont fait l'honneur d'examiner mon mémoire avec rigueur et attention.

Mes remerciements s'adressent également à Monsieur Philippe MARTINET, Habilité à diriger des Recherches, Maître de Conférences à l'Université de CLERMONT-FERRAND, qui a accepté de s'intéresser à ce travail et de participer au jury.

Je voudrais aussi remercier l'ensemble du personnel du laboratoire ainsi que les collègues de TP. J'exprime tout particulièrement ma reconnaissance à Olivier HABERT et Alain PATERNOGA dont les conseils et encouragements m'ont été très précieux. Je les remercie également pour leur disponibilité, leur gentillesse et leur amitié.

Enfin et surtout, merci, pour leur amour, leur soutien sans faille et leur patience, à Karine, à mes Parents, à Eliane ainsi que le reste de la famille et plus particulièrement Antoine et "La Marraine".

SOMMAIRE

INTRODUCTION	8
I Outils et méthodes de localisation d'un robot mobile.....	13
I.1 INTRODUCTION	13
I.2 DIFFERENTS TYPES DE CAPTEURS	14
I.2.1 <i>Les systèmes proprioceptifs</i>	14
I.2.1.1 L'odométrie	14
I.2.1.2 L'effet Doppler	16
I.2.1.3 Les mesures inertielles	16
I.2.1.4 Les magnétomètres	17
I.2.2 <i>Les capteurs extéroceptifs</i>	17
I.2.2.1 Les ultrasons	17
I.2.2.2 Les télémètres lasers	19
I.2.2.3 Les capteurs à infrarouges.....	20
I.2.2.4 Le GPS.....	20
I.2.2.5 Les caméras.....	21
I.3 MODELISATION DES MESURES	21
I.3.1 <i>Modèles géométriques</i>	21
I.3.1.1 Par télémétrie	22
I.3.1.2 Par vision	23
I.3.2 <i>Grilles d'occupation</i>	23
I.4 LA LOCALISATION	26
I.4.1 <i>Méthodes utilisant des balises</i>	26
I.4.1.1 Principe de la localisation	27
I.4.1.1.1 Par la mesure de distances.....	27
I.4.1.1.2 Par la mesure d'angles ou triangulation	28
I.4.1.2 Nature des balises	29
I.4.1.2.1 Les balises actives	29
I.4.1.2.2 Les repères artificiels	30
I.4.1.2.3 Les repères naturels.....	32
I.4.2 <i>La mise en correspondance de modèles</i>	34
I.4.2.1 Les cartes géométriques.....	35
I.4.2.2 Les grilles.....	37
I.5 CONCLUSION	38
II Méthode de localisation du VAHM	41
II.1 INTRODUCTION	41
II.2 LE CAPTEUR A ULTRASONS	41
II.3 LES GRILLES D'OCCUPATION.....	45
II.3.1 <i>Modélisation des mesures ultrasonores</i>	45
II.3.2 <i>Création de la Grille Locale</i>	50
II.3.3 <i>Définition de la Grille Globale</i>	53
II.4 MISE EN CORRESPONDANCE DES GRILLES	55
II.4.1 <i>Principe</i>	55
II.4.2 <i>Critère de mise en correspondance</i>	56
II.4.3 <i>Zone de recherche</i>	58
II.4.4 <i>Algorithme</i>	59
II.5 RESULTATS PRELIMINAIRES	61
II.5.1 <i>Simulation</i>	61
II.5.2 <i>Tests réels</i>	63
II.6 CONCLUSION	65

III AFFINEMENT DES MODELES.....	67
III.1 INTRODUCTION.....	67
III.2 CHOIX DES VALEURS DES CELLULES DE LA GRILLE LOCALE.....	68
III.2.1 <i>Cône d'un capteur à ultrasons</i>	68
III.2.2 <i>Surface du robot</i>	73
III.3 RECHERCHE DE COHERENCE POUR LA GRILLE LOCALE.....	76
III.3.1 <i>Superposition des cônes ultrasonores</i>	76
III.3.1.1 Incohérences dans la grille locale.....	76
III.3.1.2 Analogie avec les méthodes de mise à jour d'une grille d'occupation.....	77
III.3.1.3 Principes de superposition pour notre méthode.....	78
III.3.2 <i>Discretisation des arcs</i>	83
III.3.2.1 Influence de l'orientation du capteur.....	83
III.3.2.2 Notions de géométrie discrète.....	83
III.3.2.3 Application aux arcs des mesures ultrasonores.....	85
III.3.3 <i>Problèmes liés à la rotation des grilles</i>	85
III.3.3.1 Non-invariance de la rotation discrète.....	85
III.3.3.2 Application d'une rotation analogique.....	86
III.3.4 <i>Essais</i>	89
III.4 AFFINEMENT DE LA GRILLE GLOBALE ET ESSAIS.....	91
III.5 CONCLUSION.....	96
IV RECHERCHE D'UNE OPTIMISATION POUR LA MISE EN ŒUVRE DE LA METHODE.....	98
IV.1 INTRODUCTION.....	98
IV.2 REDUCTION DE GRILLES.....	99
IV.2.1 <i>Principe</i>	99
IV.2.2 <i>Mise en œuvre</i>	101
IV.2.2.1 Définition des niveaux.....	101
IV.2.2.2 Première évaluation.....	102
IV.2.2.3 Correction et résultats.....	105
IV.3 FILTRAGE DES MESURES.....	108
IV.3.1 <i>Seuillage</i>	108
IV.3.2 <i>Moyenne</i>	111
IV.4 PERSPECTIVES D'OPTIMISATION POUR LES GRILLES LOCALE ET GLOBALE.....	112
IV.4.1 <i>Dimension de la grille locale</i>	113
IV.4.2 <i>Adaptation de la grille globale</i>	115
IV.4.2.1 <i>Partie visible</i>	115
IV.4.2.2 <i>Rotation</i>	116
IV.5 CONCLUSION.....	118
CONCLUSION.....	120
ANNEXES.....	122
ANNEXE 1 : FORMULES DE CHANGEMENT DE REPÈRES.....	123
ANNEXE 2 : ESSAIS RELATIFS À LA GRILLE LOCALE.....	124
ANNEXE 3 : ESSAIS RELATIFS À LA GRILLE GLOBALE.....	126
ANNEXE 4 : ALGORITHME PERMETTANT D'OBTENIR LA GRILLE LOCALE UTILE.....	127
ANNEXE 5 : EXTRACTION DE LA PARTIE VISIBLE DE LA GRILLE GLOBALE.....	129
REFERENCES BIBLIOGRAPHIQUES.....	132

TABLE DES FIGURES

Figure ① : Le prototype du V.A.H.M et ses capteurs à ultrasons.....	10
Figure I-1 : Principe de l'odométrie.....	15
Figure I-2 : Principe du capteur à effet Doppler.....	16
Figure I-3 : Exemple de ceinture de 18 capteurs à ultrasons.....	18
Figure I-4 : Exemple d'utilisation d'un capteur tri-aural.....	19
Figure I-5 : Création d'une RCD (Regions of Constant Depth).....	22
Figure I-6 : Construction d'une carte par fusion de données de plusieurs capteurs (d'après [ELF89]).....	24
Figure I-7 : Exemple de seuils de décision.....	25
Figure I-8 : Localisation par mesure de distances de trois balises.....	27
Figure I-9 : Localisation par mesure de distances incertaines de trois balises.....	28
Figure I-10 : Localisation par triangulation.....	28
Figure I-11 : Estimation de la position d'un robot mobile utilisant un repère artificiel (d'après [KIM94]).....	31
Figure I-12 : Le capteur omnidirectionnel et l'image conique (d'après [MAR99]).....	33
Figure II-1 : Deux obstacles renvoyant la même mesure.....	41
Figure II-2 : Répartition des énergies.....	42
Figure II-3 : Phénomène de spéularité.....	43
Figure II-4 : Exemple de réflexions multiples.....	43
Figure II-5 : Exemple de diaphonie.....	44
Figure II-6 : Fonction de probabilité d'un capteur idéal.....	45
Figure II-7 : Grille pour un capteur idéal.....	45
Figure II-8 : Grille pour un capteur idéal prenant en compte l'espace libre.....	46
Figure II-9 : Exemple d'obstacle mal perçu.....	46
Figure II-10 : Grille prenant en compte l'ouverture angulaire du capteur.....	47
Figure II-11 : Profils de probabilité appliqués au capteur à ultrasons (d'après [ELF87a]).....	48
Figure II-12 : Vue des profils de probabilité en 3D.....	48
Figure II-13 : Exemple de modélisation d'un capteur à ultrasons par une gaussienne ($\sigma_r = 1\text{cm}$, $\sigma_\theta = 12^\circ$ et $d = 1\text{m}$) pour $r \in [0,1]$	49
Figure II-14 : Modélisation du capteur à ultrasons.....	50
Figure II-15 : Exemple de deux configurations possibles du robot pour la même grille.....	51
Figure II-16 : Projection de l'espace occupé par le robot sur la grille.....	51
Figure II-17 : Cellules représentant le robot.....	52
Figure II-18 : Cellules correspondant à une mesure d'un capteur à ultrasons.....	52
Figure II-19 : Exemple de grille locale (résolution de 10x10 cm par cellule).....	53
Figure II-20 : Exemple d'environnement et codage d'un objet.....	54
Figure II-21 : Grille globale (résolution de 10x10 cm par cellule).....	54
Figure II-22 : Mise en correspondance des grilles.....	56
Figure II-23 : Signe du produit des cellules.....	57
Figure II-24 : Mise en correspondance des cellules.....	58
Figure II-25 : Exemple de zones de recherche déduites à partir de l'information "se situe à l'entrée de la cuisine".....	59
Figure II-26 : Algorithme de localisation.....	60
Figure II-27 : Simulation d'une mesure des capteurs à ultrasons.....	61
Figure II-28 : Configuration du cône d'émission d'un capteur à ultrasons.....	62
Figure II-29 : Exemple de localisation réalisée en simulation.....	63
Figure II-30 : Exemple de localisation réalisée en conditions réelles.....	64
Figure II-31 : Autre exemple de localisation du VAHM.....	65
Figure III-1 : Configuration induisant de grandes différences entre les mesures.....	68
Figure III-2 : Poids de chaque mesure dans la grille locale.....	69
Figure III-3 : Correction de la modélisation du capteur à ultrasons.....	70
Figure III-4 : Exemple de quelques configurations.....	71
Figure III-5 : Mesures simulées et mesures réelles (1 : réflexions multiples, 2 : diaphonie) de la configuration n°12.....	72
Figure III-6 : Résultat de la localisation simulée pour la configuration n°19.....	74

Figure III-7 : Résultat de la localisation en corrigeant la surface du robot.....	75
Figure III-8 : Grille locale présentant des incohérences.....	76
Figure III-9 : Configurations de recouvrement de cônes possibles.....	77
Figure III-10 : Analogie entre mesures successives d'un capteur et mesures de deux capteurs.....	78
Figure III-11 : Recouvrement de cônes.....	80
Figure III-12 : Exemple où la zone commune des arcs ne correspond pas à la localisation d'un obstacle.....	81
Figure III-13 : Modélisation des recouvrements de cônes.....	82
Figure III-14 : Nombre de cellules dans l'arc en fonction de l'orientation du capteur (à droite : 0 degré, arc de 13 cellules / à gauche : 40 degrés, arc de 15 cellules).....	83
Figure III-15 : Voisins d'une cellule.....	84
Figure III-16 : Chemins 4-connexe (a) et 8-connexe (b).....	84
Figure III-17 : Correction des arcs de la figure III-14 en utilisant la 8-connexité.....	85
Figure III-18 : Non-invariance de la rotation d'une grille.....	86
Figure III-19 : Comparaison de la rotation discrète à la rotation analogique.....	88
Figure III-20 : Les trois environnements utilisés pour évaluer l'importance de l'environnement dans la réussite de la localisation.....	89
Figure III-21 : Influence de l'environnement sur le résultat de la localisation.....	90
Figure III-22 : Exemple de localisation faussée par la grille globale.....	91
Figure III-23 : Analyse de la superposition pour un cône.....	92
Figure III-24 : Obtention des contours des objets.....	93
Figure III-25 : Exemple de localisation corrigée grâce à la correction de la grille globale.....	94
Figure III-26 : Essais réalisés dans un appartement simulé.....	95
Figure IV-1 : Principe de la recherche hiérarchisée.....	100
Figure IV-2 : Algorithme de la localisation hiérarchisée.....	102
Figure IV-3 : Principe de la compression de grilles.....	103
Figure IV-4 : Exemple de compression d'une grille globale.....	104
Figure IV-5 : Exemple sur l'utilisation des réductions.....	105
Figure IV-6 : Exemple de réduction induisant des incohérences dans la grille réduite.....	106
Figure IV-7 : Création des grilles locales réduites.....	106
Figure IV-8 : Comparaison des temps de calculs sur le PC 486 embarqué.....	107
Figure IV-9 : Comparaison des temps de calculs pour réaliser la localisation.....	107
Figure IV-10 : Configuration du robot pour la distance maximale mesurable.....	108
Figure IV-11 : Filtrage des mesures par seuillage.....	109
Figure IV-12 : Réflexions multiples faussant la localisation.....	110
Figure IV-13 : Filtrage des mesures par moyenne.....	111
Figure IV-14 : Surface utile d'une grille locale.....	113
Figure IV-15 : Surface utile en plaçant le robot à 315°.....	114
Figure IV-16 : Réduction de la grille globale.....	115
Figure IV-17 : Configurations où les mesures traversent les murs.....	116
Figure IV-18 : Estimation et comparaison des temps de calculs pour la méthode proposée.....	118

INTRODUCTION

INTRODUCTION

Depuis quelques années on voit apparaître des applications scientifiques utilisant les dernières avancées technologiques développées dans le domaine de la robotique, comme le robot explorateur envoyé sur mars, ou encore dans le domaine médical avec la téléopération où l'on voit un chirurgien opérer un patient à distance. Il existe une collaboration entre la robotique et le milieu médical qui ouvre la voie à des domaines de recherche comme celui de l'aide à la mobilité des personnes handicapées moteur. Ce domaine consiste à utiliser les techniques de la robotique mobile afin d'assister les personnes handicapées dans leurs déplacements.

En règle générale, les personnes à fort handicap moteur ont recours à l'utilisation de fauteuil électrique commandé par un capteur proportionnel de type joystick. Cependant certaines d'entre elles disposent de capacités motrices résiduelles trop faibles pour pouvoir utiliser un tel capteur et ne peuvent alors actionner que des capteurs tout ou rien. Une aide à la mobilité de ces personnes est alors nécessaire ; c'est ce que le projet V.A.H.M (Véhicule Autonome pour Handicapés Moteur) se propose de développer [BOU93]. Pour ce faire, le fauteuil électrique est équipé de modules d'aides à la conduite permettant notamment l'exécution automatique de certains déplacements ; l'objectif est de réaliser un système qui adapte au mieux la répartition des tâches entre l'homme et la machine suivant le degré de handicap de l'utilisateur et la complexité de l'environnement.

Dans ce contexte, les modes de fonctionnement possibles sont :

- le mode manuel qui reproduit le fonctionnement d'un fauteuil électrique classique,
- le mode manuel assisté qui ajoute au précédent des primitives telles que le suivi de mur ou l'évitement d'obstacles afin d'alléger la charge de conduite du fauteuil,
- le mode automatique où le fauteuil effectue des déplacements autonomes pour atteindre l'objectif désigné par la personne. Dans ce cas, les problèmes rencontrés sont très similaires à ceux des robots mobiles autonomes.

Dans le cadre de la robotique mobile, pour réaliser un déplacement autonome, il est indispensable de pouvoir se repérer dans l'environnement, aussi bien au cours du mouvement qu'avant le début de la navigation. En effet, il est inutile de savoir « *où aller* » et « *comment y aller* » si on ne sait pas « *où on est* ». De ce fait, la localisation a suscité beaucoup d'intérêt dans les travaux de recherche de ces dernières années. Un système connu du grand public, le repérage par satellites GPS, a notamment fait son apparition, il est maintenant utilisé par des particuliers. Il est alors étonnant de voir qu'il est possible de repérer une voiture n'importe où sur le globe alors que

des travaux sont au stade de la recherche pour déterminer la position d'un robot dans une pièce. En effet, à cette échelle les problèmes sont tout autres, les capteurs utilisés, l'environnement et la précision sont différents.

Comme pour toute application de robotique mobile, le fauteuil doit être capable de déterminer automatiquement sa position et son orientation dans son environnement. Cette information est nécessaire aussi bien avant d'exécuter un déplacement pour planifier la trajectoire à faire suivre au fauteuil (localisation statique), qu'en cours de mouvement pour s'assurer de la bonne exécution de la navigation (localisation dynamique).

En ce qui concerne la localisation dynamique, la méthode employée permet de corriger la position odométrique du robot en cours de mouvements [HOR94]. Pour ce faire, l'environnement est décrit par un modèle global constitué d'un ensemble de segments représentant les limites de l'espace libre. Les mesures des capteurs à ultrasons définissent un modèle ponctuel local autour du fauteuil dans lequel chaque mesure est représentée. Partant de la position odométrique, la recherche du petit réajustement du robot (translation + rotation) donnant la meilleure correspondance entre le modèle local et le modèle global est effectuée. Pour cela, la distance entre les points du modèle local et les droites de l'environnement est minimisée en utilisant le critère des moindres carrés. Cette approche n'est applicable que dans le cas où la rotation et le déplacement de réajustement sont de suffisamment faible amplitude pour que les points soient associés aux droites qui conviennent.

Notre contribution dans le projet V.A.H.M consiste à établir le mode de localisation statique du fauteuil. La localisation doit intervenir soit au démarrage, soit après un déplacement si une erreur odométrique trop grande n'a pas permis d'exploiter la localisation dynamique tout au long de la trajectoire. Les seules informations disponibles à l'arrêt du robot sont les mesures des capteurs à ultrasons et une information vague sur la position du robot. Cette information provient soit du module de navigation quand l'erreur odométrique sur la position devient trop grande, soit de l'utilisateur au démarrage. Dans ce dernier cas, l'utilisateur exprime cette indication par une formulation simple du type "se situe à l'entrée de la cuisine" ou "se situe près du téléviseur".

Pour implanter les différents modules, nous disposons d'une base ROBUTER reliée à un micro-ordinateur PC486 muni d'un écran LCD autorisant un dialogue entre l'homme et la machine sans trop altérer le champ visuel de l'utilisateur (Figure ①). La base mobile munie de deux roues motrices à l'arrière et de deux roues folles à l'avant, possède des caractéristiques de mobilité similaires à celles d'un fauteuil électrique standard ; elle est équipée d'une ceinture de 14 capteurs à

ultrasons de type POLAROID placés à environ 40 cm du sol et répartis de manière à favoriser la perception dans le sens de déplacement (Figure ①).

Les caractéristiques de la base et des capteurs sont les suivantes :

Distance maximale mesurable d'un capteur à ultrasons : 12 m.

Distance minimale mesurable d'un capteur à ultrasons : 12 cm.

Ouverture du lobe des capteurs à ultrasons : 24° .

Longueur de la base : 110 cm.

Largeur de la base : 70 cm.

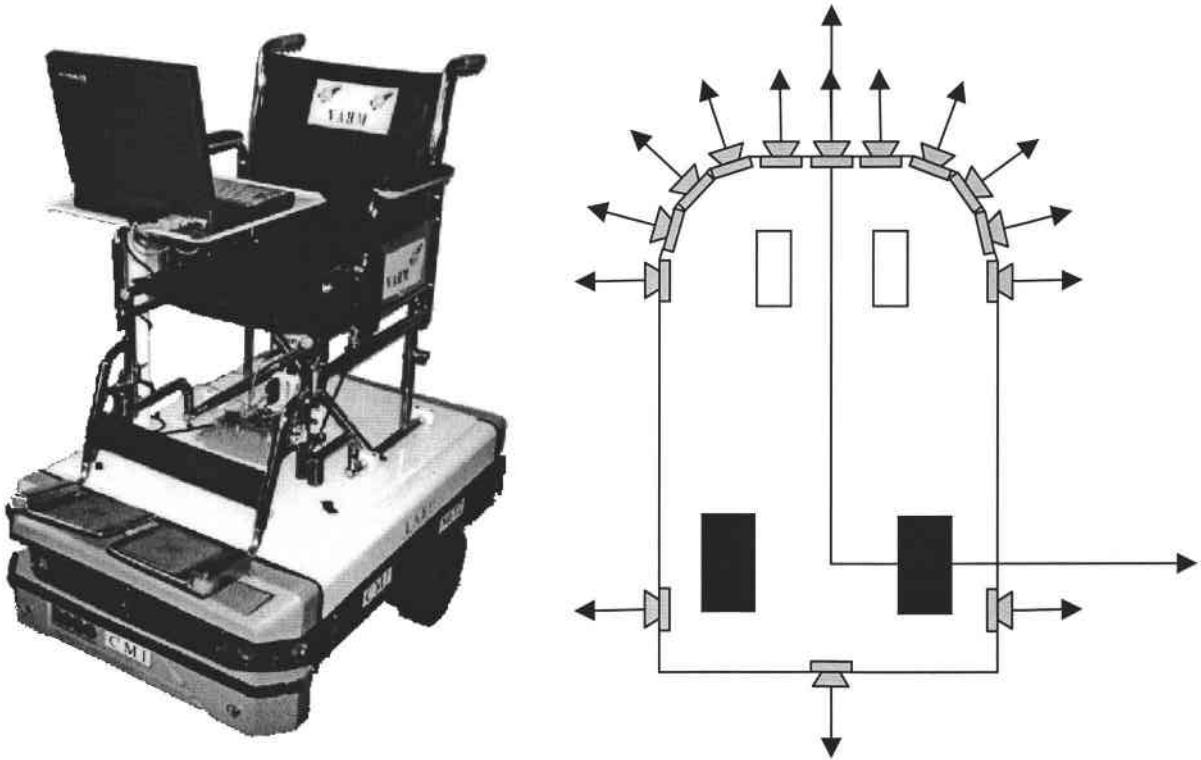


Figure ① : Le prototype du V.A.H.M et ses capteurs à ultrasons.

Ce projet étant destiné à une application médicale, il existe plusieurs contraintes spécifiques. La première est sans aucun doute le coût du système proposé. Cela implique des limites sur la performance des systèmes de commande et de perception du robot. La seconde contrainte vient de l'environnement qui ne doit pas être modifié. Enfin, une personne étant sur le fauteuil électrique, le champ d'actions est limité, nous ne pouvons pas réaliser de déplacements préliminaires à la trajectoire à effectuer.

L'originalité principale de notre travail provient de l'utilisation d'une seule ceinture de capteurs à ultrasons fournissant une quantité d'informations limitée. Nous décrivons dans ce mémoire comment à partir de ces capteurs nous déterminons la position et l'orientation du robot mobile.

Ce mémoire se compose de quatre chapitres.

Dans le premier chapitre, nous présentons les outils et les méthodes de localisation existant dans la littérature. Nous présentons les capteurs couramment utilisés, comment représenter les mesures et les différentes méthodes de localisation.

Dans le second chapitre, nous détaillons les capteurs que nous utilisons pour notre étude. Nous donnons les principes physiques qui régissent leur fonctionnement et nous exposons les différentes sources d'erreurs inhérentes à leur utilisation. Nous proposons ensuite la méthode de localisation que nous avons développée.

Cette méthode utilise deux modèles : le premier décrit l'environnement d'évolution du robot qui est connu ; le second représente l'espace perçu par le robot à partir des mesures des capteurs. Au cours du troisième chapitre, nous précisons ces modèles afin de les rendre plus cohérents et d'avoir une localisation fiable.

Enfin, le dernier chapitre s'intéresse à la recherche d'une optimisation sur les temps de calculs et sur le filtrage des mesures pour la mise en œuvre de la méthode. Nous présentons les travaux réalisés sur ce projet puis nous évoquons différents axes d'étude susceptibles d'aboutir à une amélioration de ce qui a été développé.

Dans la conclusion, nous résumons l'essentiel des travaux exposés dans ce mémoire et nous faisons ressortir les apports aussi bien dans le contexte du projet VAHM que dans le contexte général de la robotique mobile.

OUTILS ET METHODES
DE LOCALISATION D'UN
ROBOT MOBILE

I OUTILS ET METHODES DE LOCALISATION D'UN ROBOT MOBILE

I.1 INTRODUCTION

Le grand nombre de travaux sur le positionnement d'un robot mobile peut s'expliquer par le fait qu'il n'existe pas à ce jour de solution simple et élégante au problème. La plupart des travaux peuvent être classés dans une des deux catégories suivantes :

- le positionnement *relatif* qui détermine la position du véhicule par rapport à un positionnement antérieur,
- le positionnement *absolu* qui détermine la position du véhicule dans son environnement de navigation.

Bien souvent, les deux méthodes de positionnement sont combinées pour pallier les manques et les erreurs de chacune d'elles.

Il existe un lien direct entre les capteurs employés, la manière de représenter les mesures et le mode de localisation d'un robot mobile. C'est pourquoi nous décrivons d'une façon générale les capteurs couramment utilisés en robotique mobile. Nous exposons leurs avantages et inconvénients ainsi que leurs domaines d'application. Ces capteurs sont présentés dans deux parties, suivant qu'ils sont de type proprioceptif, exploités pour la localisation relative en estimant la distance parcourue par le robot, ou extéroceptif, employés pour la localisation absolue à partir de la perception de l'environnement extérieur au robot. Ensuite nous passons en revue les approches qui traitent de la modélisation des mesures en distinguant les méthodes de modélisation géométrique des méthodes de modélisation par grilles. Finalement, nous proposons plusieurs méthodes de localisation existant dans la littérature, en dissociant les techniques utilisant les balises de celles mettant en correspondance plusieurs modèles.

I.2 DIFFERENTS TYPES DE CAPTEURS

I.2.1 Les systèmes proprioceptifs

Ces systèmes sont largement utilisés en navigation pour connaître la position relative du véhicule. En effet, les capteurs sont capables de fournir des informations sur le déplacement, la vitesse et le cap (direction de déplacement) du véhicule. Cependant, ils ne peuvent pas procurer de renseignements lors de l'arrêt du véhicule.

Nous traitons dans cette partie des capteurs reposant sur l'odométrie, l'effet Doppler, les mesures inertielles et les magnétomètres.

I.2.1.1 L'odométrie

Les systèmes odométriques fournissent la position du véhicule pendant son mouvement, par intégration des rotations élémentaires de ses roues. Pour les applications de robotique mobile, les mesures de rotation des roues sont effectuées dans la plupart des cas par des codeurs optiques incrémentaux. Il existe néanmoins d'autres codeurs (magnétiques, inductifs, capacitifs...).

Le principe de l'odométrie est le suivant :

Le déplacement angulaire d'un robot à deux roues motrices est :

$$\delta\theta = \frac{R_d \cdot \delta\theta_d - R_g \cdot \delta\theta_g}{L}$$

où R_d et R_g désignent les rayons des deux roues, L représente la longueur de l'essieu entre les roues motrices, $\delta\theta_d$ et $\delta\theta_g$ indiquent les rotations élémentaires des roues droite et gauche.

Le déplacement élémentaire du robot est :

$$\delta d = \frac{R_d \cdot \delta\theta_d + R_g \cdot \delta\theta_g}{2}$$

En supposant que le déplacement élémentaire δd est circulaire ou rectiligne, et que le déplacement angulaire $\delta\theta$ reste assez petit (Figure I-1), on peut calculer la position (x_t, y_t, θ_t) actuelle du robot en fonction de la position ($x_{t-1}, y_{t-1}, \theta_{t-1}$) précédente par :

$$\begin{cases} x_t = x_{t-1} + \delta d \cdot \cos\left(\theta_{t-1} + \frac{\delta\theta}{2}\right) \\ y_t = y_{t-1} + \delta d \cdot \sin\left(\theta_{t-1} + \frac{\delta\theta}{2}\right) \\ \theta_t = \theta_{t-1} + \delta\theta \end{cases}$$

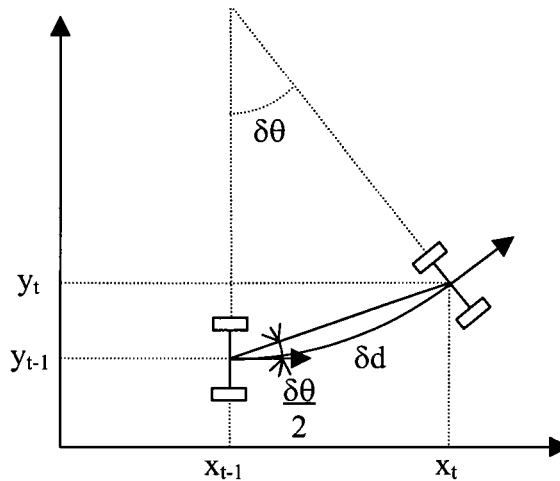


Figure I-1 : Principe de l'odométrie.

L'avantage des capteurs odométriques est qu'ils fonctionnent indépendamment de l'environnement et que la méthode est simple, facile à mettre en œuvre et d'un faible coût ; les inconvénients de ces capteurs sont qu'ils sont sujets aux problèmes provenant du glissement, de l'usure ou du fléchissement des roues et que l'erreur sur la position croît au fur et à mesure du déplacement du fait de l'intégration des mesures nécessaire pour obtenir la position du robot. On peut pallier cet inconvénient en utilisant une référence indépendante périodiquement [COX91] ; ce qui a été appliqué par exemple dans la méthode de localisation dynamique du projet V.A.H.M [HOR94], ou encore pour la correction de la position 3D d'un véhicule tous terrains [UCH98]. En modélisant les erreurs (imprécision sur les rayons des roues, non-planéité du sol) il est possible d'évaluer l'incertitude sur la position du robot au cours du mouvement [MOU89], ou à la fin du mouvement en mémorisant le trajet effectué [CHO97a]. Ces modélisations ne peuvent pas prendre en compte les erreurs dues au glissement des roues, il faut donc réussir à les détecter. Ceci peut être réalisé en utilisant un robot d'une architecture particulière, comme le robot mobile OmniMate de l'université de Michigan [BOR97]. Ce robot a la particularité d'être composé de deux parties reliées par un axe. Quand un glissement apparaît sur l'une des deux parties, la différence d'orientation mesurée au niveau de l'axe permet de l'évaluer et de corriger la position du robot.

1.2.1.2 L'effet Doppler

Les systèmes de navigation Doppler sont employés couramment pour des applications aéronautiques et maritimes pour mesurer la vitesse du véhicule par rapport à la terre. Le principe de l'opération est basé sur la différence de fréquence observée quand l'énergie rayonnée se réfléchit sur une surface qui se déplace par rapport à l'émetteur (Figure I-2).

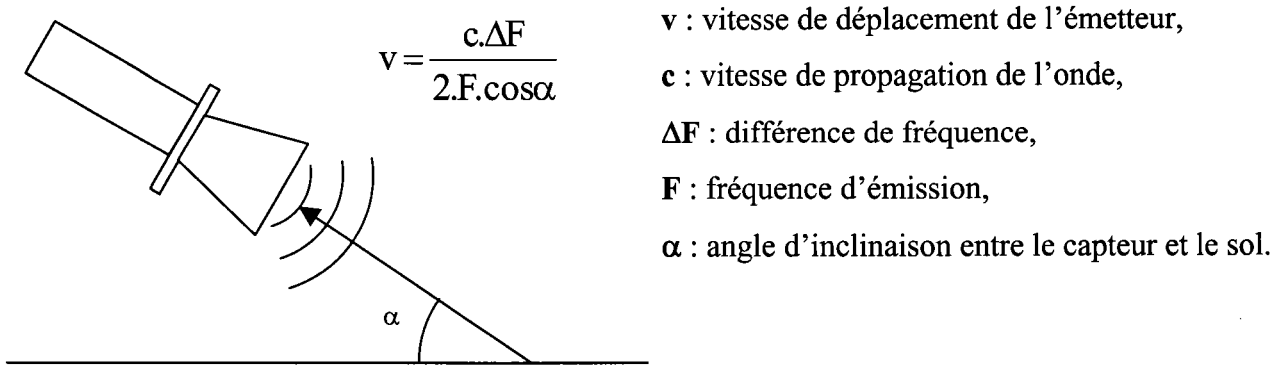


Figure I-2 : Principe du capteur à effet Doppler.

Dans certaines applications, les techniques Doppler sont employées pour réduire les effets des erreurs odométriques. Dans l'industrie agricole par exemple où le glissement des pneus peut sérieusement compromettre le besoin de répandre des graines ou des engrais à un taux proportionnel à l'avance du véhicule [HAR87]. Bien que ces capteurs soient insensibles au patinage et au glissement des roues, ils sont plus difficiles à mettre en œuvre que les capteurs odométriques. De plus, dans certains cas une mauvaise interprétation peut être mise en évidence si un véhicule est placé sur un sol en mouvement. Par exemple, si un filet d'eau coule entre les roues du véhicule, le capteur Doppler va interpréter la vitesse de l'eau comme étant celle du véhicule [BYR92].

1.2.1.3 Les mesures inertielles

Les capteurs inertiels sont de deux types. On distingue les gyroscopes qui permettent la mesure de l'angle de rotation de la trajectoire du véhicule et les accéléromètres qui mesurent l'accélération du véhicule. Ces mesures sont intégrées pour fournir la position du véhicule. Ces capteurs sont particulièrement répandus en positionnement de robots mobiles car ils aident à compenser pour la plupart, l'imprécision sur l'orientation fournie par l'odomètre [FER91],

[BOR96], [FUK96]. En effet, dans une méthode de positionnement basée sur l'odométrie, une petite erreur d'orientation va causer une erreur en position croissante au fur et à mesure du déplacement. Pour éviter cela, les capteurs inertiels sont employés pour corriger les erreurs d'orientation de l'odomètre. Cependant toutes les erreurs ne seront pas corrigées car les données inertielles dérivent elles aussi avec le temps.

1.2.1.4 Les magnétomètres

Les capteurs magnétiques sont des instruments qui mesurent le champ magnétique terrestre. Ce dernier étant perturbé par les masses métalliques, l'utilisation du compas magnétique n'est possible que pour des véhicules évoluant dans la nature, moyennant une procédure d'initialisation pour compenser l'influence magnétique de l'engin porteur. Comme les capteurs inertiels, les magnétomètres sont utilisés pour indiquer l'orientation du véhicule [ARN93], [KAR97].

1.2.2 Les capteurs extéroceptifs

Les capteurs extéroceptifs sont employés en robotique mobile dès qu'il est nécessaire de collecter des informations sur l'environnement d'évolution du robot. Ils sont notamment utilisés dans les domaines d'application tels que l'évitement d'obstacle, la localisation, la navigation et la modélisation d'environnements. Les principaux procédés couramment utilisés en robotique mobile sont : les ultrasons, les lasers, les infrarouges, le GPS et les caméras.

1.2.2.1 Les ultrasons

Les capteurs à ultrasons sont aujourd'hui les capteurs les plus communément employés dans les systèmes de robotique mobile autonome d'intérieur en raison de leurs bas prix, de leur faible consommation, de leur simplicité et de leur compacité. Depuis plusieurs années, ils ont été exploités dans des domaines tels que la navigation, la modélisation d'environnement, l'évitement d'obstacle ou l'estimation de la position. Puis des travaux ont montré la possibilité de les employer en milieu

extérieur [ELF87a], [LAN92], [PIN93]. Il existe différents types de capteurs à ultrasons ; les plus utilisés en robotique mobile sont de type Polaroid [POL91].

Le capteur à ultrasons mesure le temps de vol d'une onde ultrasonore entre son émission et sa réception après réflexion. A partir de ce temps et connaissant la vitesse de propagation de l'onde, la distance séparant le capteur de l'obstacle qu'il perçoit est directement disponible sans analyses complémentaires. Les acquisitions des mesures sont rapides, ils peuvent donc être utilisés pour les processus qui demandent un traitement en temps réel comme l'évitement d'obstacles [BOR91], [BEL94], [MEN95]. Il existe néanmoins des erreurs potentielles sur les systèmes de ces capteurs comme la variation de la vitesse de propagation de l'onde qui peut être influencée par les changements de température, par les courants d'air et dans une moindre mesure par l'humidité. Une autre caractéristique de ces capteurs est leur cône d'émission assez large (de 20 à 30°) qui ne permet pas de connaître précisément la position de l'obstacle dans le cône d'émission du capteur.

Le prix des capteurs à ultrasons n'étant pas excessif, il est possible de les disposer en ceinture autour du robot afin de percevoir une grande partie de l'environnement du robot (Figure I-3). Par ce dispositif, des régions à trous correspondant à l'espace entre les cônes de deux capteurs voisins apparaissent. S'il est utile pour l'application d'avoir une couverture totale de l'environnement, le robot doit alors se déplacer.

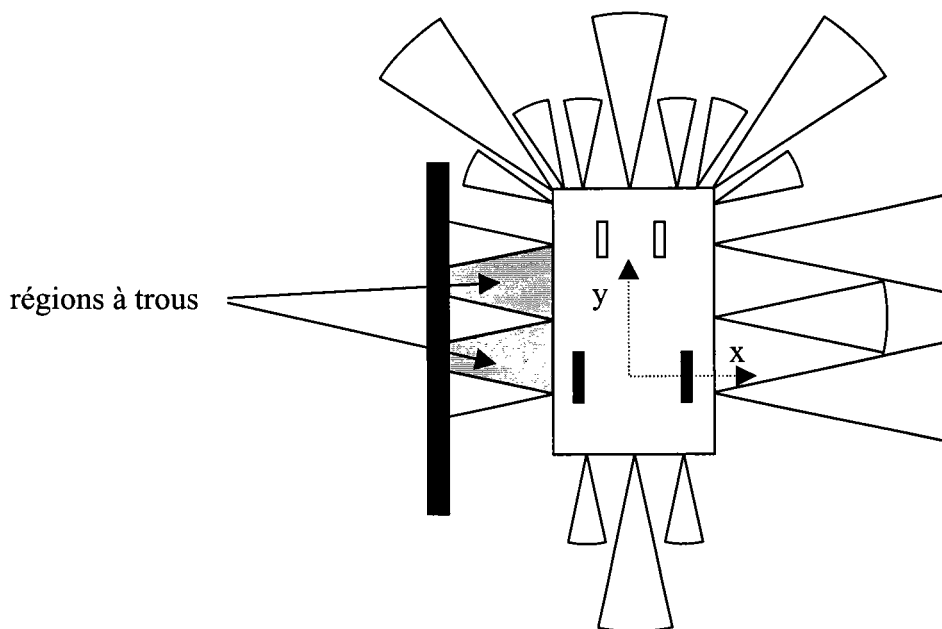


Figure I-3 : Exemple de ceinture de 18 capteurs à ultrasons.

Une autre méthode pour obtenir un balayage complet de l'environnement est d'utiliser un capteur tournant. Cela nécessite l'utilisation d'un système électromécanique qui commande les rotations du capteur [DRU87].

Par ailleurs, des systèmes composés de plusieurs capteurs ont été conçus pour affiner les mesures, par exemple un émetteur-récepteur et deux récepteurs qui constituent un capteur tri-aural (Figure I-4) [OHY94], [SAB94]. La différence entre les temps de réception de l'onde des trois récepteurs détermine l'orientation et la forme de la cible (coin, plan, cylindre...). En effet, si les trois points de mesure sont alignés, l'obstacle détecté est certainement un plan ; par contre si les trois points forment un arc, il s'agit sans doute d'un coin (arc courbé vers le capteur) ou d'une arête (arc fuyant le capteur).

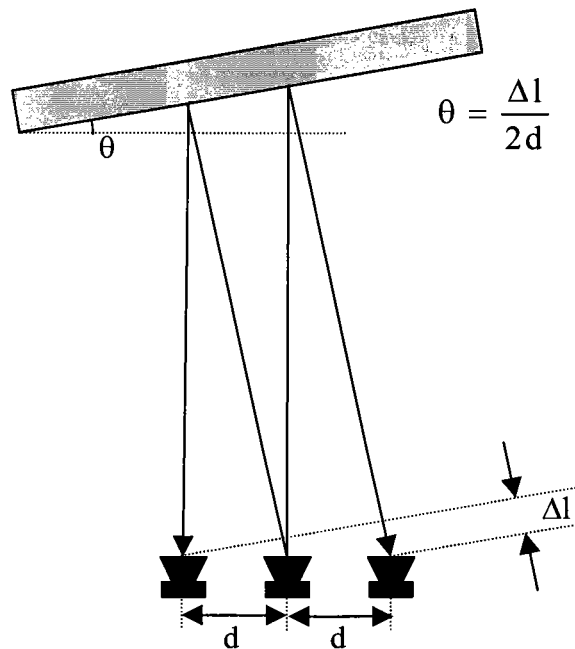


Figure I-4 : Exemple d'utilisation d'un capteur tri-aural.

I.2.2.2 Les télémètres lasers

Il existe deux manières de mesurer une distance grâce à un système laser. Soit il émet en mode continu et la distance est déterminée à partir du déphasage entre l'onde émise et l'onde réfléchiée par l'obstacle, soit il émet en mode pulsé et la distance est proportionnelle au « *temps de vol* » de l'onde entre son émission et sa réception.

Le faisceau d'ondes émis est très concentré, ce qui permet d'avoir un cône d'émission très étroit et donc une bonne précision de mesure. Pour obtenir un balayage de l'environnement en deux,

voir en trois dimensions, différents systèmes de miroirs tournants peuvent être associés au laser [JOR95], [KWO97], [WEC98].

Les lasers existants diffèrent par leurs caractéristiques techniques et leurs domaines d'applications. En effet, la distance maximale mesurable des lasers peut aller d'une dizaine de mètres jusqu'à quelques centaines de mètres.

La précision angulaire des systèmes laser est meilleure que celle des capteurs à ultrasons, ils sont cependant moins utilisés car plus coûteux, plus fragiles et plus délicats à mettre en œuvre en raison du matériel optique utilisé (lentilles, miroirs...).

1.2.2.3 Les capteurs à infrarouges

Les capteurs à infrarouges fonctionnent suivant le même principe que le laser et les capteurs à ultrasons. Ils se composent d'un couple émetteur-récepteur. L'émission s'effectue par une diode électroluminescente infrarouge. Le récepteur est une photodiode ou un phototransistor, sensible au flux lumineux rétrodiffusé par la cible.

Les capteurs à infrarouges sont d'un faible encombrement mais sensibles aux perturbations lumineuses ainsi qu'à la nature des surfaces réfléchissantes. Leurs portées n'étant que de quelques mètres, ils sont utilisés essentiellement pour détecter les obstacles proches [GOU94].

1.2.2.4 Le GPS

Un nouveau système de localisation apparaît de plus en plus ces dernières années : le système de positionnement global GPS [MIY97], [LI98]. Le GPS (Global Positioning System) est un système basé dans l'espace utilisant les ondes radio. Mis à part son utilisation pour la localisation qui sera détaillée au chapitre I.4.1.2.1, ce système est aussi utilisé pour :

- l'aide à la navigation d'avions et de bateaux lors de l'approche d'aéroports ou de ports,
- la surveillance de camionnettes de livraison et de véhicules d'urgence, en vue de choisir au mieux leurs itinéraires.

Ce sont des systèmes coûteux, une précision centimétrique se chiffre en centaines de milliers de francs et les ondes employées sont fortement atténuées en traversant les murs, ce qui rend impossible l'utilisation du GPS en milieu intérieur [PET97].

1.2.2.5 Les caméras

La caméra permet d'obtenir une image de l'environnement local du robot. C'est sans doute le capteur fournissant la plus grande variété d'informations à partir d'une seule acquisition. En effet, en appliquant divers algorithmes de traitement d'images, il est possible d'extraire des éléments caractéristiques de l'environnement comme des segments de droite, des plans, mais aussi des balises, des couleurs, ou encore des distances en utilisant par exemple la stéréovision.

Les systèmes de vision peuvent être utilisés pour la navigation par fusion avec des capteurs à ultrasons [TRA97], ou avec un système GPS [LI98]. La modélisation en trois dimensions de l'environnement est aussi possible en s'aidant de mesures de distances de capteurs à ultrasons [AKB96], ou d'un télémètre laser [WEC98]. La localisation de robots mobiles est un domaine de recherche dans lequel ce capteur est très employé. Certains travaux réalisent une localisation par fusion des données issues de l'image avec celles fournies par un odomètre [DEL97], [OHY97] ou avec les données d'un laser [NEI96]. A partir de repères naturels choisis dans une image, le mouvement du robot peut être évalué en recherchant ces mêmes repères dans l'image suivante [FER97], [HAG97]. Par ailleurs lorsque le robot est à l'arrêt, la recherche dans l'image de repères fixes dont les positions sont connues dans l'environnement permet d'accomplir sa localisation statique [MAD98], [MAR98]. D'autres travaux ont aussi évalué l'utilité des capteurs vidéo pour corriger l'orientation d'un robot destiné à l'exploration lunaire [COZ97].

Les limites rencontrées avec ces capteurs sont leur sensibilité aux conditions d'éclairage et le matériel informatique puissant requis pour les algorithmes de traitements.

I.3 MODELISATION DES MESURES

1.3.1 Modèles géométriques

Les modèles géométriques sont des entités telles que des points, des segments de droites, des rectangles, des polygones...obtenues à partir des mesures et constituant une représentation des différents éléments de l'environnement. La représentation la plus utilisée pour la localisation de robots mobiles est celle par segments de droite, que l'on obtient en effectuant un pré-traitement des points de mesure qui consiste à les filtrer et à les regrouper.

Nous avons différencié les modèles géométriques obtenus à partir des mesures de capteurs télémétriques qui donnent la limite des obstacles, des modèles établis à partir d'une image d'un capteur de vision qui donne une autre perception de l'environnement.

1.3.1.1 Par télémétrie

Il existe différents filtrages et regroupements de points de mesure. Par exemple la méthode proposée par [CRO89a] consiste à prendre les mesures d'un capteur à ultrasons tournant une par une, et à créer un segment dès que trois mesures sont suffisamment proches et alignées. Pour qu'un nouveau point puisse appartenir à un segment existant, il faut que la distance du point au segment soit plus petite que l'incertitude sur la mesure. D'autres travaux ont repris cette approche en adaptant la détection des segments en fonction des capteurs utilisés [MAC94].

Au lieu de regrouper les points de mesure suivant leur alignement pour avoir une représentation par segments, il est possible de les réunir en fonction de la distance entre le capteur (tournant) et les points, ce qui constitue une région conique appelée RCD (Regions of Constant Depth) [BEO95]. Ainsi, quand plusieurs mesures consécutives renvoient une distance semblable, elles sont regroupées en un arc défini par la distance δ entre le capteur et l'obstacle, l'orientation β de l'axe central du RCD et l'ouverture angulaire η (Figure I-5).

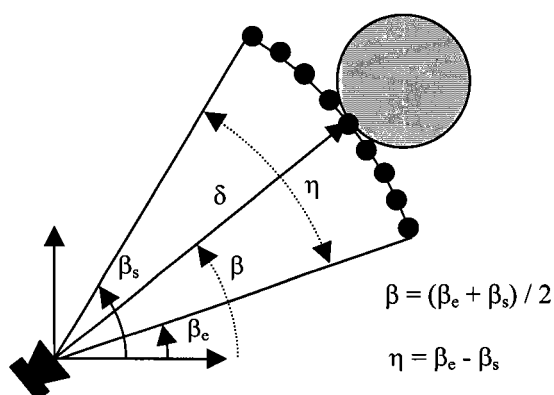


Figure I-5 : Création d'une RCD (Regions of Constant Depth).

I.3.1.2 Par vision

Pour obtenir une représentation géométrique de l'environnement à partir d'un capteur de vision il faut réaliser un pré-traitement de l'image. Ce traitement consiste à extraire les contours des objets présents dans l'image grâce à un filtre de « *Sobel* » par exemple utilisé dans [DEL97] et [ARR99] ou encore un filtre de « *Deriche* » exploité par [MAR99]. Dans la plupart des travaux de localisation de robots mobiles, des algorithmes utilisent uniquement les contours verticaux pour établir une liste de segments. Ainsi, dans les travaux de recherche sur l'estimation de l'orientation du robot par vision monoculaire [KRE99], les segments sont obtenus en recherchant les contours verticaux ayant un nombre de pixels consécutifs supérieur à un seuil fixé. En travaillant avec les contours extraits de deux images (par un capteur de vision stéréoscopique), il est possible d'obtenir une carte 3D composée de segments [WEC98].

I.3.2 Grilles d'occupation

Ces méthodes consistent à représenter les mesures sous forme d'un ensemble de cellules. Chaque cellule possède plusieurs attributs exprimant chacun une propriété de la zone correspondante, notamment son occupation. Pour des scènes d'intérieur, où le sol est plat, la plupart des chercheurs décrivent l'état de chaque cellule par un attribut unique qui est la probabilité d'occupation. On parle alors de *grille d'occupation*. Une cellule est dite *libre* si sa probabilité d'occupation est inférieure à un seuil choisi et *occupée* dans le cas inverse.

MORAVEC, ELFES et MATTHIES ont été parmi les premiers à utiliser le principe des grilles d'incertitude (ou grille d'occupation) [MOR85], [ELF87b]. L'objectif de leurs travaux est de construire de manière autonome la carte de l'environnement d'un robot mobile. Pour cela le robot évolue dans un environnement inconnu non structuré et s'y déplace en évitant les obstacles, sans disposer d'aucun modèle a priori du lieu où il se trouve. Il doit construire une carte du lieu à partir des seules informations données par une ceinture de 24 capteurs à ultrasons de type Polaroid montée sur le robot. Chaque mesure de distance fournie par cette ceinture est interprétée comme définissant un volume d'espace probablement vide et un volume probablement occupé dans le cône d'émission du capteur. Cette information d'occupation est une densité de probabilité que l'on projette verticalement sur une carte à deux dimensions, où l'on voit apparaître des aires élémentaires vides, occupées et inexplorées. Le profil de probabilité utilisé permet de générer deux

grilles d'occupations, l'une représentant les régions certainement vides, l'autre représentant les régions certainement occupées.

Au fur et à mesure que le robot se déplace, les nouvelles mesures sont intégrées dans ces cartes, affinant ainsi la connaissance des zones déjà explorées, et découvrant les autres. La mise à jour des cartes s'effectue en utilisant des formules d'addition de probabilités [ELF87a]. Ces formules permettent de calculer les nouvelles probabilités d'occupation et de non-occupation à partir des probabilités mémorisées dans la cellule et des nouvelles probabilités fournies par la nouvelle mesure. Deux zones vides superposées se renforcent mutuellement, deux zones occupées également. Une zone occupée et une zone vide superposées donnent une zone soit occupée soit vide suivant la plus forte des probabilités. Les zones vides mettent en évidence les limites des régions occupées. A chaque nouvelle mesure l'incertitude sur les positions des obstacles diminue et la précision de la carte augmente.

Cette méthode peut être utilisée à partir des mesures issues de plusieurs types de capteurs [ELF89]. Chaque capteur construit sa carte locale qui va ensuite être intégrée à une carte locale du robot servant à établir la carte globale de l'environnement (Figure I-6).

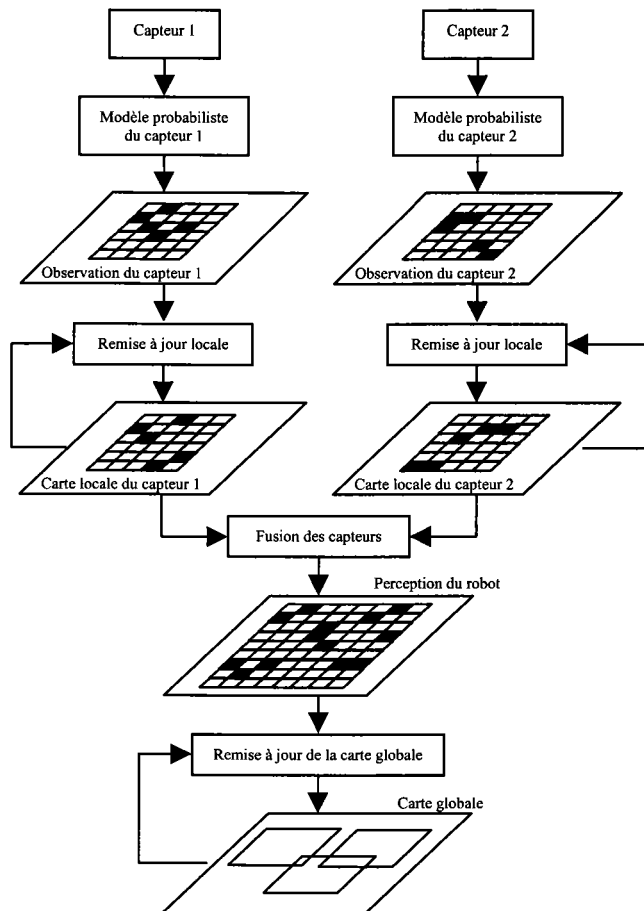


Figure I-6 : Construction d'une carte par fusion de données de plusieurs capteurs (d'après [ELF89]).

Un inconvénient de ce principe de mise à jour de la probabilité d'occupation d'une cellule est l'élimination des mesures contradictoires. Dans d'autres travaux [BUL96], un test permet de gérer les mesures contradictoires en les traitant différemment. Quand une cellule libre à l'instant t est perçue comme étant occupée à l'instant $t+1$, la probabilité d'occupation de la cellule est incrémentée. Par contre quand une cellule occupée à l'instant t est perçue comme étant libre à l'instant $t+1$, un test est lancé pour vérifier si la nouvelle mesure a subi une spécularité. Si le résultat du test est positif, la probabilité d'occupation de la cellule n'est pas modifiée, dans le cas contraire elle est décrémentée.

Ce principe favorise la représentation des obstacles, ce qui est appréciable pour la sécurité du robot, mais cette méthode différencie l'ordre d'arrivée des mesures. En effet, une cellule d'abord libre puis perçue comme étant occupée n'est pas traitée de la même manière qu'une cellule occupée qui par la suite est perçue comme étant libre. Ce qui peut être gênant puisque pour le même environnement, en reprenant les mêmes configurations du robot en sens inverse, le résultat de la carte sera différent.

Certains travaux utilisent ces grilles d'occupation en remplaçant le seuil de décision unique par deux seuils (Figure I-7) ; cela permet de donner une valeur neutre (ni libre, ni occupée) aux cellules ayant une valeur proche du seuil unique [SOI97].

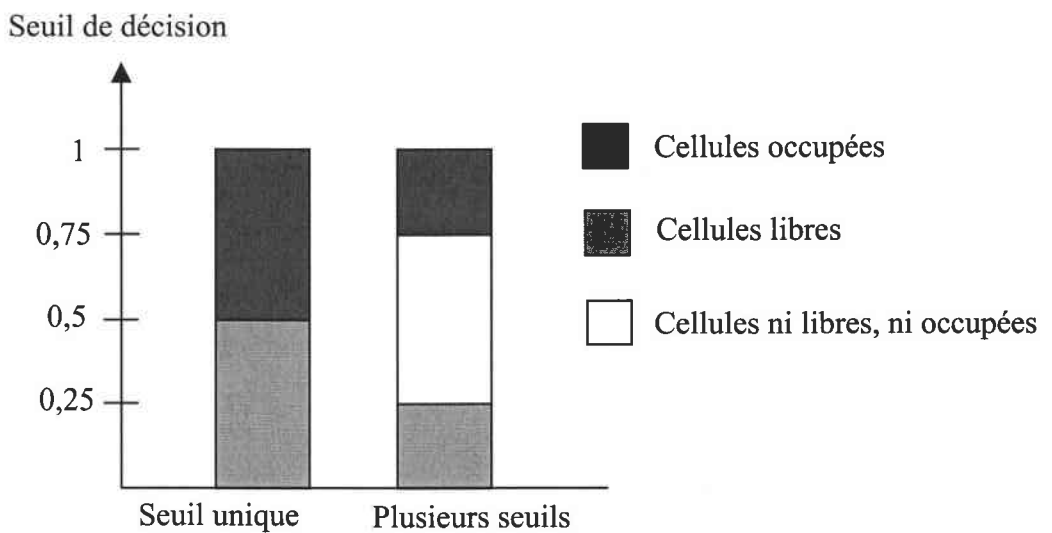


Figure I-7 : Exemple de seuils de décision.

D'autres auteurs [PAG98] utilisent une modélisation différente. Chaque cellule de la carte contient deux probabilités : la probabilité P_L que la cellule soit libre et la probabilité P_O que la cellule soit occupée. La mesure d'un capteur à ultrasons peut être représentée par un cône composé

d'un secteur représentant l'espace libre et d'un arc indiquant la présence d'un obstacle. Chaque probabilité d'une cellule est définie par une valeur inversement proportionnelle au nombre de cases recouvertes par le cône. Par exemple s'il y a n cellules sur l'arc et m cellules dans le secteur, les

$$\text{probabilités de chaque cellule de l'arc sont : } \begin{cases} P_L = 0 \\ P_O = \frac{1}{n} \end{cases} \text{ et les probabilités de chaque cellule du} \\ \text{secteur sont : } \begin{cases} P_L = \frac{1}{m} \\ P_O = 0 \end{cases} .$$

Pour pouvoir ensuite utiliser des règles de probabilités pour la mise à jour de la carte, une cellule doit avoir la somme de ses probabilités égale à 1 ce qui n'est pas le cas puisque pour une cellule $P_L + P_O \neq 1$. On a alors recours à l'interprétation des données ultrasonores par la théorie de DEMPSTER-SHAFFER [SHA76]. Cette théorie ajoute deux probabilités : la probabilité P_\emptyset que la cellule ne soit ni vide ni occupée et la probabilité P_{LO} que la cellule soit vide ou occupée. Par ailleurs, pour chaque cellule de la grille il est impossible qu'une cellule ne soit ni vide ni occupée et donc $P_\emptyset = 0$, de plus $\Sigma P = P_\emptyset + P_L + P_O + P_{LO} = 1$. Dans ces conditions il est seulement nécessaire de déterminer les probabilités P_L et P_O puis P_{LO} est calculé en utilisant la relation $P_{LO} = 1 - P_L - P_O$. Pour mettre ensuite à jour les probabilités d'une cellule lors d'une nouvelle mesure, les nouvelles valeurs sont calculées en fonction des trois valeurs mémorisées (cellule) et des trois nouvelles données (mesure) par les règles de combinaison de DEMPSTER.

I.4 LA LOCALISATION

I.4.1 Méthodes utilisant des balises

La localisation à l'aide de balises peut se faire soit par la mesure des distances séparant le véhicule des différentes balises grâce à la télémétrie, soit par la mesure de l'angle sous lequel le robot perçoit chaque balise au moyen de la goniométrie.

Chacune de ces deux méthodes peut utiliser trois types de balises dont les positions sont connues ; les balises actives qui émettent un signal, les balises passives ou repères artificiels qui réfléchissent le signal envoyé par un instrument embarqué et les repères naturels ou amers qui sont des caractéristiques de l'environnement.

I.4.1.1 Principe de la localisation

I.4.1.1.1 Par la mesure de distances

Si l'on connaît les distances qui séparent le robot de plusieurs balises, on peut en déduire sa position par l'intersection de cercles (en 2D) ou de sphères (en 3D). Par exemple, en deux dimensions, l'intersection des trois cercles centrés sur trois balises donne la position du véhicule (Figure I-8).

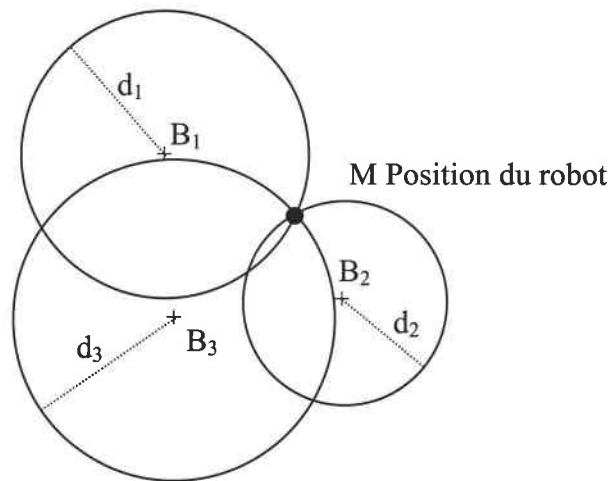


Figure I-8 : Localisation par mesure de distances de trois balises.

Soit $M(x_M, y_M)$ la position du robot.

Soit d_i la distance mesurée entre le robot et la balise $B_i(x_i, y_i)$.

La position de M est calculée en cherchant la solution (x, y) du système composé des équations des cercles C_i :

$$C_i : (x-x_i)^2 + (y-y_i)^2 = d_i^2 \quad \text{pour } i = 1..3$$

Si l'on prend en compte les erreurs de mesures, il faut alors considérer des anneaux à la place des cercles. Et on obtient ainsi une zone à la place d'un point (Figure I-9).

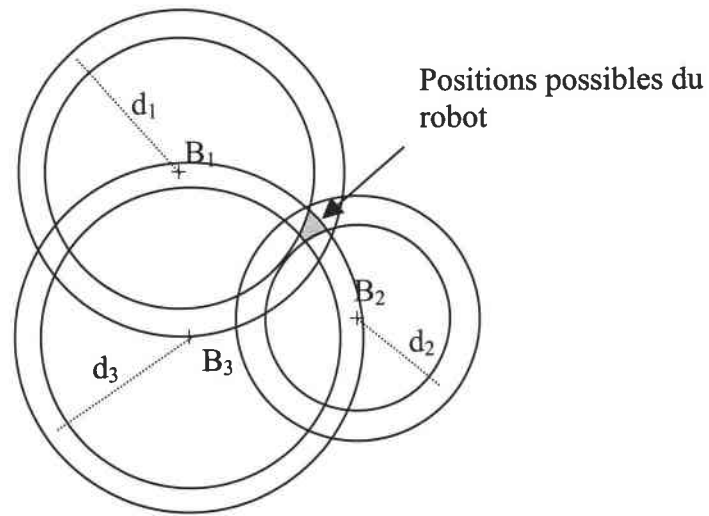


Figure I-9 : Localisation par mesure de distances incertaines de trois balises.

I.4.1.1.2 Par la mesure d'angles ou triangulation

Au lieu d'employer les distances, cette méthode utilise les angles séparant l'axe du véhicule des directions sous lesquelles il perçoit les balises. Par triangulation on en déduit la position du robot (Figure I-10).

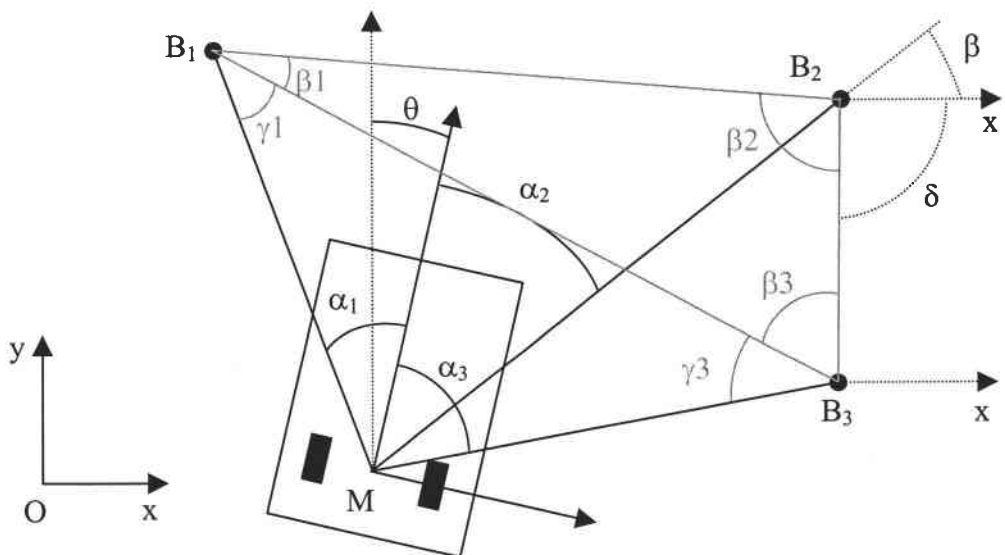


Figure I-10 : Localisation par triangulation.

Soit $M(x_M, y_M)$ la position du robot et θ son orientation.

Soit α_i l'angle sous lequel le robot perçoit la balise B_i .

Sachant les positions de chaque balise dans l'environnement, les distances $B_i B_j$, les angles β_i et l'angle δ sont connus. A partir des α_i , les angles $B_3 M B_2$ et $B_1 M B_2$ sont déterminés. Les propriétés de trigonométrie dans un triangle nous donnent les relations :

$$\frac{MB_2}{\sin(\beta_3 + \gamma_3)} = \frac{B_2 B_3}{\sin(B_3 M B_2)} \text{ et}$$

$$\frac{MB_2}{\sin(\beta_1 + \gamma_1)} = \frac{B_2 B_1}{\sin(B_1 M B_2)}$$

En isolant MB_2 on obtient une relation où les seules inconnues sont γ_1 et γ_3 . En utilisant la relation $B_3 M B_2 + B_1 M B_2 + \gamma_1 + \gamma_3 = \pi$, on exprime γ_1 en fonction de γ_3 . Puis on substitue γ_1 dans la relation obtenue précédemment, ce qui permet de déterminer γ_3 .

Connaissant MB_2 et γ_3 on détermine l'angle β par la relation $\beta = B_3 M B_2 + \gamma_3 + \beta_3 - \delta$ et on obtient les coordonnées x_M et y_M du robot et son orientation θ par :

$$\begin{cases} x_M = x_{B_2} - MB_2 * \cos(\beta) \\ y_M = y_{B_2} - MB_2 * \sin(\beta) \\ \theta = -\frac{\pi}{2} + \beta + \alpha_2 \end{cases}$$

1.4.1.2 Nature des balises

1.4.1.2.1 Les balises actives

Les balises actives ont été utilisées depuis plusieurs siècles pour la navigation maritime. En effet, les étoiles peuvent être considérées comme des balises actives, ainsi que les lumières des phares. Dans ce contexte, les émetteurs actuels, utilisant généralement des fréquences radio ou lumineuses, doivent être à des emplacements connus de l'environnement.

Un des systèmes les plus connus de ces dernières années est le Système de Positionnement Global. Chaque satellite du GPS émet un signal contenant des informations telles que son identification, l'heure d'émission du signal et les paramètres définissant son orbite donc sa position. En connaissant sa distance par rapport à trois satellites, ainsi que la localisation de chaque satellite au moment de l'émission du signal, le récepteur peut calculer sa position en trois dimensions. Un

exemple d'utilisation de ce système est donné dans [MIY97] qui décrit une application destinée à une tondeuse à gazon autonome pour les terrains de sport ; la tondeuse doit décrire des lignes parallèles sur l'ensemble du terrain. L'estimation de la position est fournie par un système proprioceptif composé d'un gyroscope optique et d'un odomètre. La dérive progressive du gyroscope et les problèmes liés à l'odométrie dégradent le parallélisme de la trajectoire, et l'espace entre deux lignes n'est pas uniforme. En utilisant le GPS, et en fusionnant toutes les données, l'erreur maximale sur l'espace entre deux lignes est réduite à 20cm au lieu de 40cm sans le GPS.

Dans d'autres travaux, les données du GPS ont été fusionnées avec les données d'une caméra [LI98]. Les données d'un GPS bon marché fournissent la position approximative du mobile (erreur de l'ordre de 30 à 100 mètres), puis l'algorithme corrige la position du robot en comparant l'image actuelle de la caméra avec celles mémorisées lors d'une phase d'exploration.

D'autres travaux utilisent des diodes électroluminescentes à infrarouges comme balises actives [BRA95]. Chaque balise est constituée de trois diodes, elles sont installées au plafond pour permettre les déplacements temporaires de matériel dans la zone d'évolution du robot mobile. Chaque balise module le flux lumineux de ses trois diodes par un code différent. Ceci permet au robot d'identifier la balise en utilisant un récepteur à infrarouges. Pour se localiser, le robot observe la balise à l'aide d'une caméra de type CCD matricielle. En déterminant la position des trois diodes dans l'image de la caméra et connaissant la position de chacune d'elles dans l'environnement, le robot peut alors calculer sa position. Des balises à infrarouges peuvent aussi être utilisées pour corriger les erreurs odométriques [OUS97], [PAN98].

Notons qu'une source sonore peut aussi être une balise active [HUA97]. Le capteur dans ce cas est constitué de trois micros. En mesurant les différences de temps de réception entre les trois micros, la direction du son et donc l'orientation du robot peuvent être déduites.

I.4.1.2.2 Les repères artificiels

Dans cette méthode, des repères artificiels sont placés à des endroits connus de l'environnement. Comme les balises actives, il est nécessaire de « voir » au moins trois balises pour permettre l'estimation de la position. Cependant, en utilisant un capteur de vision, une seule image peut représenter plusieurs repères, il est alors possible d'en extraire la position et l'orientation du robot.

Par ce principe, un panneau dont la position est connue et sur lequel sont dessinées des bandes verticales blanches et noires peut être utilisé comme repère [KIM94]. Une caméra CCD vise le panneau et l'algorithme proposé détermine la position du robot en analysant les déformations des bandes vues dans l'image (Figure I-11).

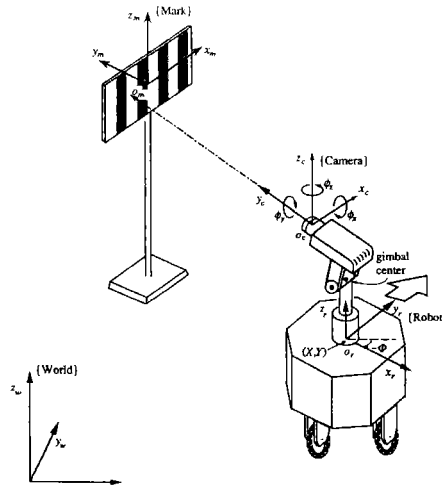


Figure I-11 : Estimation de la position d'un robot mobile utilisant un repère artificiel (d'après [KIM94]).

Une autre application, destinée à l'atterrissage automatique d'hélicoptère, utilise comme balise un cercle blanc peint sur le sol [SAT97]. L'image du cercle donnée par la caméra embarquée sur l'hélicoptère est une ellipse. Comme dans le cas précédent, la déformation de la balise dans l'image est utilisée. La méthode s'appuie sur la résolution du système d'équations définissant l'ellipse pour déterminer la position relative de l'hélicoptère par rapport au cercle.

En environnement intérieur, les repères peuvent être des réflecteurs placés sur le mur. La localisation du robot est alors réalisée par une triangulation pour laquelle les angles séparant le robot des balises sont mesurés. Ces angles peuvent être mesurés à partir d'un capteur à ultrasons tournant et de réflecteurs acoustiques [KO96], ou d'un goniomètre laser et de réflecteurs de lumières [HAN96]. Dans ce dernier exemple, l'environnement long d'environ 14 mètres nécessite 34 balises pour couvrir tout l'espace.

En règle générale trois balises identiques sont nécessaires pour pouvoir localiser le robot. Dans le cas de deux balises identiques, l'intersection des deux cercles fournit deux solutions possibles. Si les balises ne sont pas identiques et qu'il est possible de les distinguer lors des

mesures, deux balises suffisent pour lever l'ambiguïté sur une éventuelle deuxième position. Ce principe est utilisé pour estimer la position et l'orientation d'un robot mobile en utilisant deux balises passives et un capteur à ultrasons tournant [BEO95]. Les balises sont deux cylindres de diamètres différents. Le capteur à ultrasons scrute l'environnement et en déduit les régions de même profondeur (RCD). Pour trouver les deux RCD correspondant aux deux balises, l'algorithme élimine toutes celles qui ne satisfont pas aux trois critères géométriques suivant :

- test angulaire : par expérience, les auteurs ont montré que les balises cylindriques utilisées renvoient des RCD avec une ouverture angulaire comprise entre 18 et 26 degrés. Les RCD n'ayant pas leur ouverture angulaire comprise dans cet intervalle sont supprimées.
- test de symétrie : les RCD renvoyées par des balises cylindriques sont symétriques par rapport à l'axe reliant le capteur au centre de la balise. Ce test vérifie que le nombre de mesures de part et d'autre de l'axe d'une RCD est sensiblement le même.
- test de distance : la distance séparant les deux balises est connue. La distance entre deux RCD est alors calculée et comparée à la distance réelle. Deux RCD trop proches ou trop éloignées sont éliminées.

Les deux RCD restants fournissent les deux distances des balises au robot et ce dernier peut alors se localiser.

Les balises peuvent aussi servir à corriger la position du véhicule lors de la navigation. Ce principe a été utilisé pour une application destinée à un véhicule autonome sur autoroute [KIM97]. Dans cette application, une carte est créée au cours d'une phase d'exploration dans laquelle une personne conduit le véhicule. Ensuite, dans la phase de navigation, l'estimation de la position est fournie par un gyroscope, mais elle dérive avec le temps. Un radar détectant les balises (cubes ou objets naturels métalliques) permet alors de corriger la position du véhicule.

I.4.1.2.3 Les repères naturels

Ici, les repères sont naturels, dans le sens où ils ne nécessitent pas un équipement spécial de l'environnement. On parle dans ce cas *d'amers*. La fiabilité de ces méthodes n'est pas aussi grande

que celle des repères artificiels puisqu'il faut réussir à qualifier les repères rencontrés pour les différencier les uns des autres.

De récents travaux se situent dans ce contexte [MAR99]. La méthode proposée consiste à localiser un robot mobile connaissant la position des amers verticaux de l'environnement tels que les montants des portes. Ces amers sont détectés grâce à un capteur de vision omnidirectionnelle constitué d'un miroir conique qui reflète l'environnement sur une caméra. Ce capteur a l'avantage de fournir une vue panoramique (360°) de l'environnement (Figure I-12). Les droites représentant les différents éléments verticaux de l'environnement (montants de portes, coins, arêtes...) convergent vers le centre de l'image. Les auteurs utilisent le *birapport*, principe basé sur le rapport de distances, pour identifier dans l'image quatre droites correspondant à quatre éléments verticaux connus dans l'environnement. L'orientation du robot dans l'image étant représentée par un segment dirigé vers le haut, l'angle sous lequel le robot perçoit un élément vertical de l'environnement est directement disponible. La position du robot et son orientation sont alors calculées par triangulation.

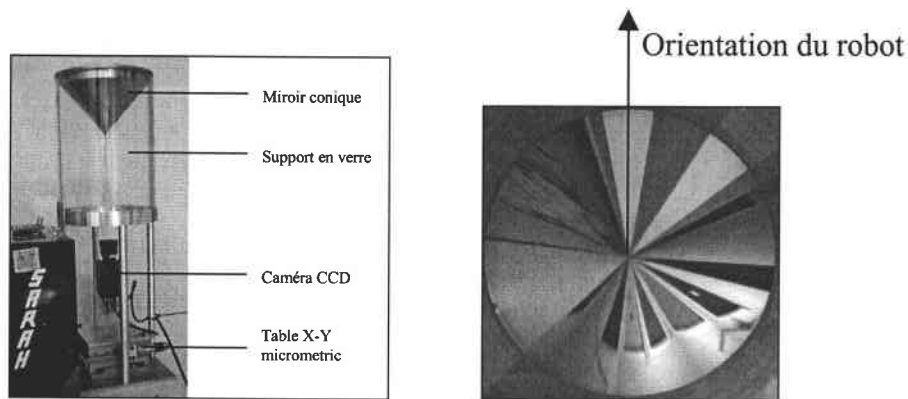


Figure I-12 : Le capteur omnidirectionnel et l'image conique (d'après [MAR99]).

D'autres repères naturels ont été utilisés dans une application de positionnement dans un couloir qui permet à un robot de transporter les plateaux repas dans un milieu hospitalier [HAS97]. Les repères utilisés sont des objets de position connue dans le couloir et facilement détectables par un système de vision stéréo (cadres, extincteurs, montant des portes...). Ces repères sont répertoriés dans une carte de l'environnement. En partant d'une première position du véhicule initialisée par l'utilisateur, le robot détermine les coordonnées des repères détectés dans l'image. Puis, au cours du mouvement le principe de poursuite de cible corrige la position du robot en comparant les coordonnées des repères dans l'image à celles procurées par la carte. Il arrive dans certains cas que les objets de l'environnement soient en mouvement et que certains repères ne soient plus visibles. L'unité de poursuite de cible risque donc de s'interrompre. Pour pallier ce défaut, plusieurs repères

sont utilisés. Quand un repère n'est plus visible, et connaissant la relation spatiale qu'il a avec un deuxième repère, l'unité de poursuite continue en prenant comme cible le deuxième repère.

Lorsque plusieurs repères sont disponibles pour la localisation, une sélection est nécessaire. Par exemple, dans une application utilisant la vision pour déterminer la position du robot par un calcul de triangulation, de nombreux repères (lignes verticales, cadres, posters...) sont disponibles dans l'environnement [MAD98]. Le choix du triplet de repères permettant d'avoir la localisation la plus précise est donc déterminant. Pour ce faire, la position estimée du robot, fournie par un système odométrique, permet de sélectionner tous les amers pouvant être visibles. Puis la liste de tous les triplets possibles est déduite. Cette liste est ensuite réduite en ne considérant que les triplets susceptibles de se trouver simultanément dans le champ de vision de la caméra. Un critère basé sur la déformation des repères dans l'image permet de ne garder que le triplet optimisant la localisation. Ce triplet est alors utilisé pour effectuer le calcul de triangulation.

1.4.2 La mise en correspondance de modèles

La méthode de localisation consiste dans ce cas à trouver la meilleure correspondance entre deux modèles. Le premier, représentant l'environnement partiellement ou dans sa totalité, est appelé carte globale. Le second, correspondant à la partie de l'environnement perçue par le robot depuis sa position, est appelé carte locale.

Comme nous l'avons vu dans la partie précédente, il existe principalement deux méthodes pour représenter les mesures : la méthode géométrique et la méthode par grilles. Ces deux méthodes induisent donc deux techniques de localisation distinctes. Une troisième technique liant les deux méthodes est aussi possible.

La carte locale est formée à partir des informations acquises par les capteurs embarqués sur le robot, et dépend de la position du robot. En revanche la carte globale est connue soit par intégration de plusieurs cartes locales lors de l'exploration d'un environnement inconnu, soit par une connaissance à priori de l'environnement. Dans le cas où seules certaines parties de l'environnement total sont intéressantes pour la tâche à effectuer, il est utile d'avoir recours à une carte topologique. Ce type de carte représente l'environnement par un graphe dont les nœuds représentent des lieux particuliers de l'environnement et les arcs désignent le chemin entre deux de ces lieux. C'est le cas par exemple d'un environnement constitué d'un couloir donnant accès à différentes salles de travail. Seules les salles ont un intérêt pour la localisation (chaque salle

représente un nœud du graphe), le couloir ne servant qu'à relier les salles (chaque chemin entre deux salles représente un arc du graphe) [COU94].

1.4.2.1 Les cartes géométriques

Pour établir la carte locale géométrique, il est souvent nécessaire en raison de l'imprécision sur les mesures des capteurs de modéliser les incertitudes. Dans le cas de la localisation dynamique, de nombreux travaux utilisent alors le filtre de Kalman généralisé qui permet d'estimer successivement après chaque acquisition de mesures la position du robot. Lorsque la carte globale de l'environnement est construite au fur et à mesure de l'exploration du robot, ce filtre permet aussi d'estimer la position de chaque élément géométrique introduit dans la carte.

Dans les travaux de LEONARD, DURRANT-WHYTE et COX [LEO92], le filtre de Kalman est appliqué de la manière suivante :

Une prédiction des mesures des capteurs est réalisée à partir d'une position approximative du robot fournie par l'odomètre, de la connaissance de l'environnement et de la modélisation des capteurs. Les données réelles des capteurs sont ensuite enregistrées et la différence entre les mesures obtenues et les mesures prédites constitue l'innovation. Une porte de validation est appliquée sur cette dernière afin d'éliminer les mesures réelles trop différentes des mesures prédites. Puis partant de cette innovation et de la position approximative du robot qui constitue la prédiction, le filtre de Kalman permet de calculer la position estimée du robot.

Ce filtre a été utilisé pour construire une carte de l'environnement à partir de capteurs à ultrasons [LAW95], d'un télémètre laser [CAS97], ou d'un système de vision [OHY97]. Cependant, la vision étant un système demandant beaucoup de temps de calcul, cette technique est utilisée par rétroaction pour éviter l'arrêt du robot ; le temps que le robot fasse ses calculs, il continue à avancer, puis, quand le résultat est disponible, la position est corrigée en tenant compte du déplacement supplémentaire effectué.

Cette méthode a aussi été appliquée pour corriger la position du robot [HON97b] en introduisant dans le filtre les informations odométriques et les balises géométriques 3D détectées par un capteur à ultrasons spécifique [HON97a].

Une autre approche permet de réaliser une mise en correspondance de cartes sachant qu'il existe une incertitude sur ces dernières en utilisant la méthode des moindres carrés [COX91]. Le plan de l'environnement fourni est constitué d'un ensemble de segments et à partir de l'odométrie,

la position du robot est approximativement connue dans ce plan. Un capteur laser monté sur une tourelle, permet au robot d'acquérir un panorama complet de l'environnement. La mise en correspondance des mesures avec le plan se fait par association du point de mesure au segment du modèle le plus proche. La somme des distances au carré séparant les points de mesure des segments associés est calculée et la position du robot est obtenue en recherchant celle qui minimise cette somme. Cette approche est aussi employée à partir des mesures issues d'une ceinture de capteurs à ultrasons pour la localisation en cours de mouvement développée dans le cadre de notre projet [HOR94]. Dans ces travaux, le mode d'association d'une mesure au segment le plus proche est réalisé en définissant une ellipse d'incertitude pour chaque segment, et en cherchant pour un point donné son appartenance à une ellipse.

Pour réduire le temps de recherche du segment le plus proche du point de mesure, la méthode des moindres carrés peut être employée avec une carte à deux niveaux [GON95]. Le premier niveau est constitué d'un ensemble de segments, le second d'une grille. Les cellules de cette grille indiquent s'il y a au moins un segment dans la cellule ou s'il n'y a aucun segment dans cette cellule. Ceci permet de simplifier la recherche de segments proches d'une mesure puisque celle-ci se fait uniquement sur les segments passant dans la case pleine la plus proche.

Dans une autre méthode utilisant des cartes composées de segments de droites à la place de points, le processus de mise en correspondance entre le modèle local et le modèle global consiste à rechercher la similarité entre les segments de droite suivant trois critères : orientation, position et longueur. Si plusieurs segments passent les trois tests, le plus long d'entre eux est choisi. La position du robot est corrigée dans une direction perpendiculaire au segment identifié et l'orientation est corrigée en fonction de la différence d'orientation entre les segments détectés et modélisés [CRO89a].

Les segments issus de la carte globale peuvent aussi être fournis par une représentation topologique. La base topologique de l'environnement est constituée de cartes locales construites en différents endroits de la pièce [EIN97]. Chaque carte correspond à une liste de segments formés à partir des mesures d'un laser panoramique scrutant l'environnement sur 360°. La localisation du robot se fait tout d'abord par la recherche de la représentation la plus proche de la mesure actuelle par comparaison des segments. Puis l'algorithme calcule la transformation $(\Delta_x, \Delta_y, \Delta_\phi)^T$ faisant correspondre au mieux (au sens des moindres carrés) la représentation trouvée à la nouvelle acquisition. La nouvelle position du robot est donc déterminée à partir de la position de la représentation la plus proche sur laquelle est appliquée la transformation $(\Delta_x, \Delta_y, \Delta_\phi)^T$.

1.4.2.2 Les grilles

Une première méthode de mise en correspondance de deux grilles est décrite dans [ELF87a] ; les cartes globale et locale sont constituées de cellules dont les valeurs indiquent la présence ou l'absence d'un obstacle. Pour ce faire l'auteur affecte des valeurs positives aux cellules représentant les obstacles et des valeurs négatives aux cellules correspondant aux espaces libres. En superposant les deux grilles, le produit de deux cellules superposées indique s'il y a correspondance ou non puisque le produit de deux cellules identiques est positif et qu'il est négatif quand elles sont opposées. Ce produit est calculé sur toute la grille et la somme de tous ces produits est évaluée. Ainsi, en calculant cette somme pour différentes positions et orientations de la carte locale par rapport à la carte globale, l'emplacement qui fournit la somme maximale et donc qui donne la meilleure correspondance entre les deux grilles permet de localiser le robot.

Une autre méthode à base de grilles utilise une représentation topologique pour décrire l'environnement [COU94] ; chaque nœud de la carte topologique correspond à une pièce de travail, le couloir reliant deux pièces étant représenté par un arc. Une première phase d'exploration permet au robot de construire la carte topologique en représentant chaque pièce par trois grilles construites à partir des mesures issues d'un type de capteurs : capteurs à ultrasons, caméra et capteurs à infrarouges. Les grilles sont créées par la méthode HMM (Histogramic In-Motion Mapping) introduite par BORENSTEIN et KOREN [BOR91a], puis sont décrites de façon unique en calculant des paramètres basés sur les valeurs des cellules de la grille. Pour déterminer la pièce de travail au cours de la navigation, le robot construit de nouvelles grilles par cette même méthode. En comparant les paramètres extraits des grilles actuelles à ceux extraits des grilles construites lors de l'exploration, la région de travail (couloir ou différentes salles) dans laquelle le robot se trouve est déterminée. Cette localisation ne sera donc que grossière puisqu'elle n'indique que la région de travail et ne précise pas le positionnement exact du robot dans cette région.

Une autre méthode de mise en correspondance de deux cartes a été développée pour corriger les mesures odométriques [YAM96]. La particularité de la méthode est qu'elle ne dispose pas de carte globale mais de deux cartes locales créées à deux instants différents. Ces cartes sont établies en utilisant les profils probabilistes des capteurs définis par ELFES. Les mesures sont fournies par 16 capteurs à ultrasons et un télémètre laser qui scrutent l'environnement sur 360°. La première carte locale est créée et l'odomètre est initialisé. Le robot effectue sa mission puis revient au point de départ. Là, une deuxième carte locale est créée. Les deux cartes sont ensuite superposées pour calculer un facteur de correspondance M . Ce facteur augmente à chaque fois que deux cellules identiques sont superposées. Ce calcul est effectué pour toutes les translations et rotations possibles

d'une grille par rapport à l'autre ; puis la transformation qui maximise M est choisie pour corriger l'erreur odométrique. Les inconvénients de cette méthode sont le temps d'acquisition qui nécessite plus de deux minutes par cartes et la nécessité de revenir au point de départ pour effectuer la comparaison des deux grilles.

1.5 CONCLUSION

Nous avons vu au cours de ce chapitre qu'il existe une grande variété de méthodes de localisation qui diffèrent par les capteurs utilisés, les modèles d'environnement et le mode d'association entre les mesures et le modèle. Les choix sont établis en fonction des applications auxquelles elles sont dédiées.

Le cadre de notre étude se situe dans l'aide à l'autonomie des personnes handicapées moteur. Notre objectif est d'établir la méthode pour la localisation statique d'un fauteuil électrique autonome, qui est développé pour libérer la personne des contraintes de conduite de ses déplacements. Le robot évolue dans un appartement, l'environnement est connu (plan d'architecte) et de dimensions relativement faibles (quelques mètres, voire une dizaine de mètres). L'application devra s'adapter facilement à différents appartements pour différentes personnes sans nécessiter d'équipements trop lourds. Les méthodes de localisation utilisant les balises n'ont donc pas été envisagées.

Une personne se trouvant sur le fauteuil et ne voulant pas subir de mouvements non demandés, le robot doit alors pouvoir se situer sans se déplacer et dans une position quelconque ; dans ce contexte les méthodes de mise en correspondance de modèles géométriques sont peu appropriées. En effet, les modèles géométriques, représentant l'environnement en une liste de primitives géométriques du type point, segment, plan..., permettent d'avoir une représentation des éléments de l'environnement mais demandent une quantité importante de mesures. L'emploi de tels modèles peut être bien adapté pour une localisation dynamique dans laquelle plusieurs mesures successives sont interprétées. Pour utiliser cette méthode dans le cas d'une localisation statique, seul un capteur tournant permet d'avoir une couverture complète de l'environnement mais ce type de capteur ne saurait trouver sa place sur le fauteuil.

En résumé, la méthode de localisation doit pouvoir s'employer quel que soit l'emplacement du robot, sans déplacements, à partir de mesures de capteurs à ultrasons fixes et doit fournir une position et une orientation suffisamment précises pour qu'une trajectoire puisse débiter. Cette précision est fixée par le module de localisation dynamique, établi par ailleurs, pour qu'il puisse

être initialisé puis utilisé lors du mouvement. Les valeurs sont : une erreur maximale en position de 10 cm et un écart en orientation de moins de 7 degrés.

La mise en place d'une méthode par mise en correspondance de grilles nous paraît donc appropriée puisque cette méthode permet de représenter les incertitudes associées aux mesures des capteurs, notamment les capteurs à ultrasons, qu'elle peut être appliquée à partir d'un nombre de mesures limité et qu'elle présente l'avantage de fournir une représentation bien adaptée aux traitements informatiques.

METHODE DE
LOCALISATION DU
VAHM

II METHODE DE LOCALISATION DU VAHM

II.1 INTRODUCTION

Au cours de ce chapitre, nous présentons les grilles d'occupation développées pour notre méthode de localisation. La première grille, appelée grille locale, permet de décrire l'environnement proche du robot grâce aux données des capteurs à ultrasons. La seconde grille, appelée grille globale, représente l'environnement d'évolution du robot. La méthode de localisation est ensuite décrite et des résultats préliminaires concluent le chapitre.

II.2 LE CAPTEUR A ULTRASONS

Comme nous l'avons décrit au chapitre I.2.2.1, le capteur à ultrasons présente l'avantage de donner directement une information de distance mais son inconvénient est que cette mesure est assez imprécise. En effet, l'angle d'ouverture (de 20 à 30°) introduit un facteur d'incertitude concernant la direction dans laquelle se situe l'obstacle perçu. Pour une mesure d donnée par le capteur, en considérant un espace à deux dimensions et en négligeant les incertitudes sur la mesure, la région correspondant aux positions possibles de l'obstacle détecté prend la forme d'un arc centré sur le capteur, avec un rayon égal à d . Les dimensions de l'objet sont aussi masquées par cette propriété du capteur à ultrasons : il peut s'agir d'un mur ou seulement d'un objet de petite taille. La figure II-1 illustre la situation où deux obstacles possibles produisent la même mesure.

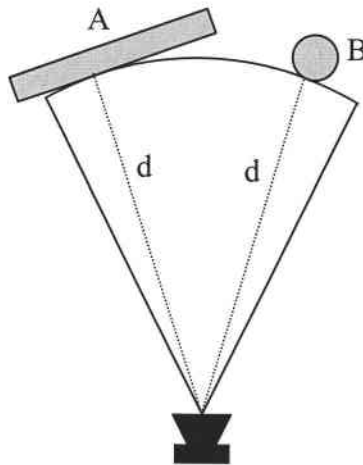


Figure II-1 : Deux obstacles renvoyant la même mesure.

Une incertitude sur la distance mesurée, appelée également incertitude radiale, est aussi présente. Elle provient des phénomènes atmosphériques (température, courants d'air...) pouvant modifier la vitesse de l'onde.

Pour les capteurs utilisés comme émetteur et récepteur tels que ceux que nous utilisons, une distance minimale détectable est définie. Cette distance est déduite du temps nécessaire pour que la membrane du capteur puisse se stabiliser après l'émission de l'onde (12 cm sur le prototype).

Quand une onde acoustique heurte un objet, un écho détecté représente seulement une petite partie du signal original. L'énergie restante est réfléchiée dans des directions dispersées et peut être absorbée par la cible ou passée à travers elle, ceci dépendant de la surface de l'objet et de l'angle d'incidence du rayon (Figure II-2). Par ailleurs, plus la distance parcourue par l'onde est grande plus cette dernière est atténuée. Ainsi, une distance maximale correspondant au minimum d'énergie détectable est définie comme limite de la portée du capteur (12 m sur le prototype).

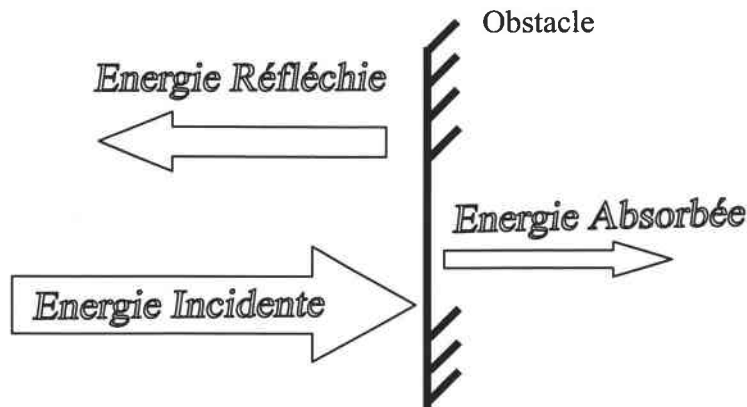


Figure II-2 : Répartition des énergies.

Dans le domaine de perception du capteur, des phénomènes tels que la spécularité, les réflexions multiples ou la diaphonie peuvent intervenir, nous allons en décrire les causes et les effets.

↳ La spécularité.

La spécularité, encore appelée effet miroir peut provoquer la non-détection des obstacles. En effet, la fiabilité des mesures ultrasonores est très dépendante de la texture de la surface réfléchissante de l'objet. Les surfaces rugueuses retournent l'onde ultrasonore produite par le capteur quel que soit l'angle d'incidence, contrairement aux surfaces lisses qui ne renvoient que les ondes ayant un angle d'incidence proche d'un angle droit (Figure II-3). Dans le cas d'une étude en

simulation, la modélisation du capteur à ultrasons proposée par [HAR98] permet de prévoir la distance qui devrait être mesurée dans un cas réel connaissant la texture de la surface, l'angle d'incidence de l'onde et la distance séparant le capteur de l'obstacle.

Si une spécularité intervient, la mesure issue du capteur concerné est inexploitable, ce qui peut être gênant puisque nous n'avons que 14 mesures de capteurs à ultrasons pour décrire l'environnement autour du robot. Si l'onde ne se réfléchit pas sur un autre obstacle pour revenir sur le capteur, la mesure sera nulle ; par contre, si la spécularité est suivie d'une ou plusieurs réflexions sur les obstacles, nous avons alors des réflexions multiples.

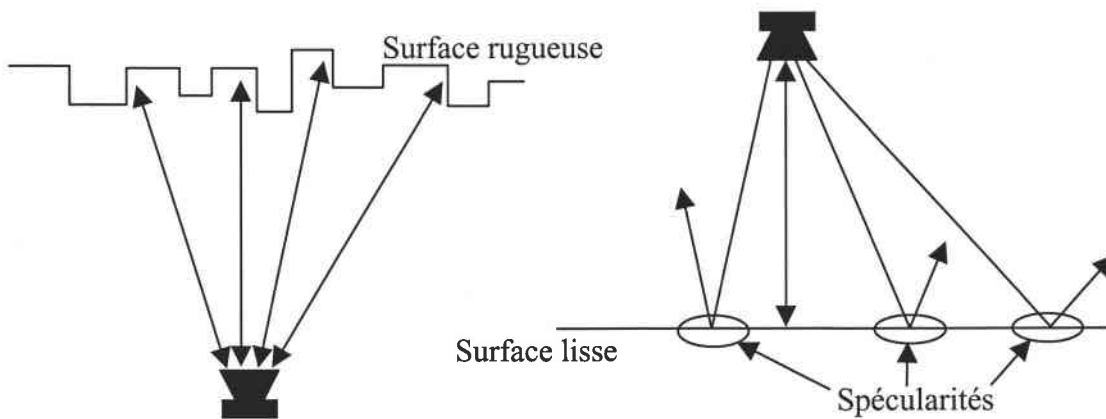


Figure II-3 : Phénomène de spécularité.

↳ Les réflexions multiples.

Le phénomène de spécularité peut se produire plusieurs fois, il s'ensuit des réflexions multiples. Le signal ultrasonore peut donc se réfléchir sur plusieurs surfaces avant de retourner au capteur. Par ce fait, l'obstacle paraît plus loin que son emplacement réel (Figure II-4). Les mesures issues de réflexions multiples ne correspondent pas à un obstacle de l'environnement et risquent de fausser la localisation en associant la mesure à un autre obstacle situé plus loin.

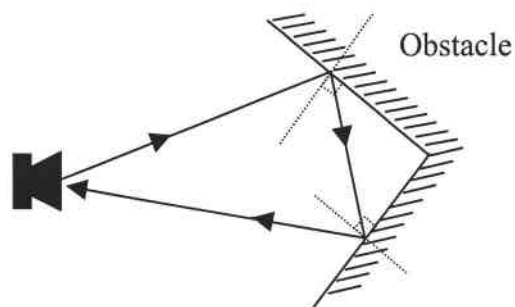


Figure II-4 : Exemple de réflexions multiples.

↳ La diaphonie.

La diaphonie intervient quand un capteur perçoit l'onde émise par un autre. Supposons qu'un capteur x émette une onde ultrasonore, le cône émis est renvoyé vers le robot. Si le capteur y se trouve en mode de réception, il interprétera l'onde émise par le capteur x et renvoyé par l'obstacle comme étant la sienne, et percevra un objet plus proche qu'il ne devrait. Ce phénomène est appelé diaphonie par chemin critique direct (Figure II-5a). Le capteur y peut aussi recevoir une onde issue de réflexions multiples, il s'agit alors de diaphonie par chemin critique indirect (Figure II-5b). Là encore ce type de mesures représente des obstacles fictifs et peut fausser la localisation ; le regroupement des capteurs sous forme de nœuds vise à limiter la diaphonie en évitant d'activer simultanément des capteurs voisins.

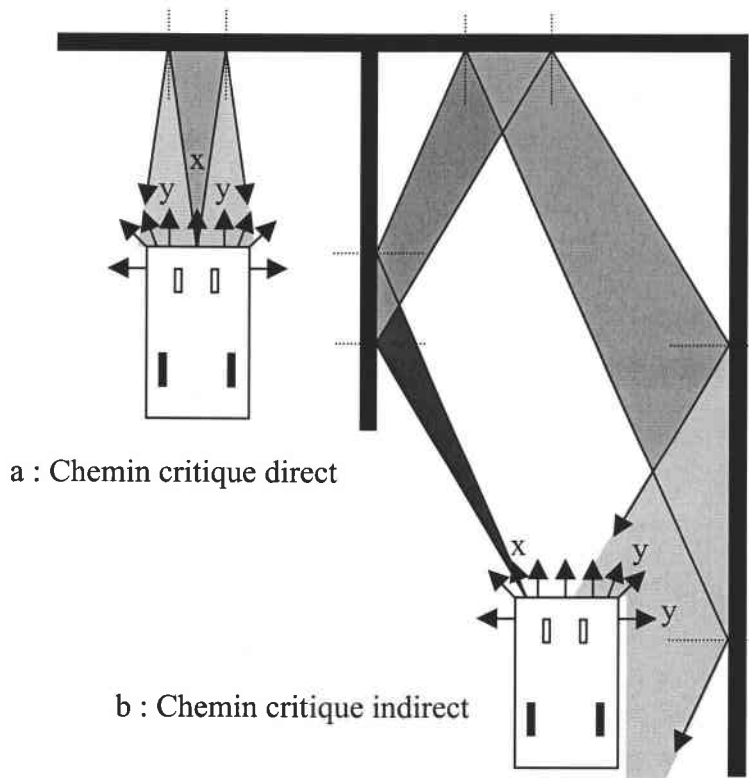


Figure II-5 : Exemple de diaphonie.

La spécularité, les réflexions multiples et la diaphonie sont des phénomènes parasites plus ou moins imprévisibles contrairement aux incertitudes angulaire et radiale qui sont toujours présentes dans la mesure. En conséquence, le modèle que nous proposons représente les incertitudes angulaire et radiale mais ne tient pas compte des autres phénomènes. Il est toutefois important de connaître leur existence afin de prendre en considération la possibilité de mesures aberrantes dont il faudra limiter l'influence lors de la localisation.

II.3 LES GRILLES D'OCCUPATION

II.3.1 Modélisation des mesures ultrasonores

Un moyen de modéliser le capteur à ultrasons consiste à déterminer la fonction densité de probabilité $p(r | d)$ de la mesure de profondeur r connaissant la distance d entre le capteur et l'obstacle.

En supposant le capteur idéal, la mesure est interprétée comme étant la distance au plus proche obstacle dans la direction de l'axe acoustique (Figure II-6).

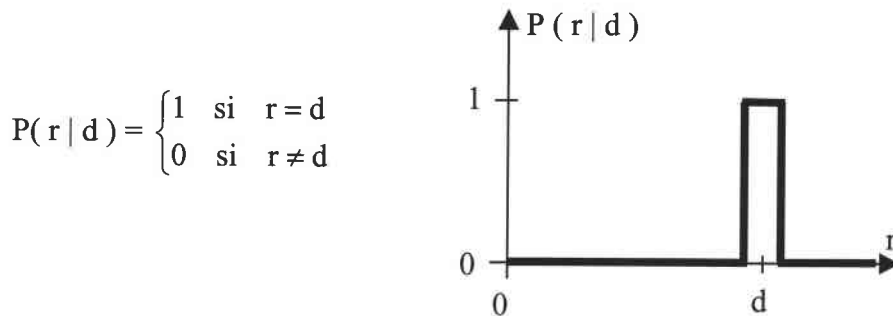


Figure II-6 : Fonction de probabilité d'un capteur idéal.

En enregistrant ce modèle dans une grille, seule la cellule située sur l'axe acoustique du capteur à la distance mesurée est modifiée (Figure II-7). De cette façon, l'incertitude sur la distance mesurée peut être contenue dans la cellule.

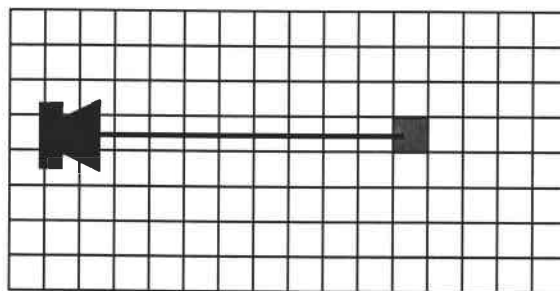


Figure II-7 : Grille pour un capteur idéal.

Ce modèle a été utilisé pour effectuer des tâches d'évitement d'obstacles pour un robot mobile [BOR91]. Les cellules de la grille représentant l'environnement sont initialisées à la valeur 0. Puis, au cours du déplacement du robot, chaque obstacle détecté incrémente la valeur de la cellule correspondante. Ainsi, au fur et à mesure du déplacement, les obstacles sont repérés par des cellules dont les valeurs augmentent, ce qui permet de les éviter.

La modification de ce modèle, prenant en compte l'espace libre entre le capteur et l'obstacle, a permis de réaliser la localisation d'un robot mobile [COU94]. Comme précédemment, la valeur initiale de chaque cellule de la grille est nulle. Puis, lors d'une acquisition, les valeurs des cellules correspondant à l'espace libre sont décrémentées tandis que la valeur de la cellule représentant l'obstacle est augmentée de la valeur 3, les valeurs des cellules restantes n'étant pas modifiées (Figure II-8).

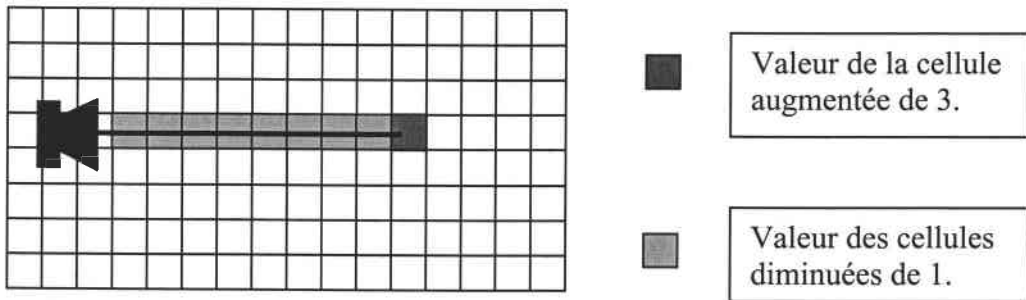


Figure II-8 : Grille pour un capteur idéal prenant en compte l'espace libre.

Le problème de ces modèles est qu'ils ignorent l'incertitude angulaire des capteurs à ultrasons. Un capteur de type Polaroid a un cône d'émission de 24° , les réflexions retournées au capteur peuvent provenir d'objets situés dans la périphérie du cône. Ramener la mesure simplement sur l'axe de tir, considérant le capteur à ultrasons comme un capteur ponctuel (comme le laser), peut donc être source d'erreurs car beaucoup d'obstacles seront mal ou pas perçus (Figure II-9).

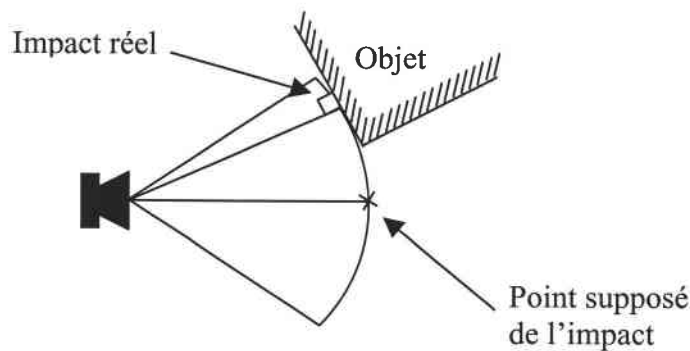


Figure II-9 : Exemple d'obstacle mal perçu.

L'incertitude sur les mesures intervient donc sur la profondeur r , mais aussi sur l'angle θ par rapport à l'axe de tir. Une première approche utilisée pour la détection d'obstacles lors de la navigation de robots mobiles consiste à modifier le premier modèle (Figure II-7) en un arc de cercle

(Figure II-10) [MEN95] ; la valeur attribuée à chaque cellule étant inversement proportionnelle au nombre de cellules recouvertes par l'arc entier.

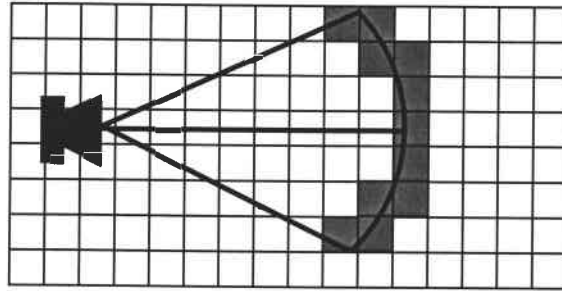


Figure II-10 : Grille prenant en compte l'ouverture angulaire du capteur.

Une autre méthode pour représenter la mesure sur le secteur entier consiste à estimer que si un objet est détecté par le capteur à ultrasons, il est plus probable que cet objet soit proche de l'axe acoustique que de la périphérie du cône. Selon ce principe, les données ultrasonores sont interprétées par des profils de probabilité [ELF87a]. Le cône ultrasonore est décomposé en deux volumes de l'espace : l'un représentant l'espace libre, l'autre représentant l'espace occupé. Les points situés dans la région certainement vide ont une probabilité d'être libres P_L donnée par :

$$P_L = L_r(r) \cdot L_a(\theta)$$

avec :

$$L_r(r) = \begin{cases} 1 - \left(\frac{r - R_{\min}}{R - \varepsilon - R_{\min}} \right)^2 & \text{si } r \in [R_{\min}, R - \varepsilon] \\ 0 & \text{sinon} \end{cases}$$

$$L_a(\theta) = 1 - \left(\frac{2\theta}{\Omega} \right)^2 \quad \text{si } \theta \in \left[-\frac{\Omega}{2}, \frac{\Omega}{2} \right].$$

r et θ désignent les coordonnées polaires d'un point dans un référentiel placé sur le capteur, R correspond à la distance mesurée, R_{\min} représente la distance minimale détectable par le capteur, ε désigne l'incertitude sur la mesure et Ω décrit l'ouverture angulaire du capteur.

De même, les points situés dans la région certainement occupée ont une probabilité d'être occupés P_O définie par :

$$\mathbf{P}_O = \mathbf{O}_r(\mathbf{r}) \cdot \mathbf{O}_a(\theta)$$

avec :

$$\mathbf{O}_r(\mathbf{r}) = \begin{cases} 1 - \left(\frac{r-R}{\varepsilon} \right)^2 & \text{si } r \in [R - \varepsilon, R + \varepsilon] \\ 0 & \text{sinon} \end{cases}$$

$$\mathbf{O}_a(\theta) = 1 - \left(\frac{2\theta}{\Omega} \right)^2 \quad \text{si } \theta \in \left[-\frac{\Omega}{2}, \frac{\Omega}{2} \right].$$

La figure II-11 illustre ces distributions de probabilité pour un capteur à ultrasons fournissant la mesure R , et la figure II-12 représente ces probabilités en trois dimensions.

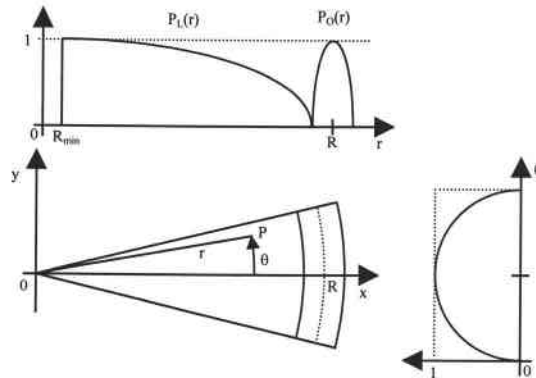


Figure II-11 : Profils de probabilité appliqués au capteur à ultrasons (d'après [ELF87a]).

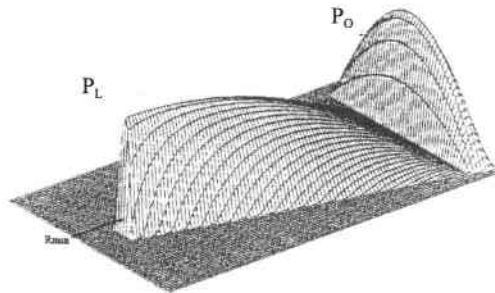


Figure II-12 : Vue des profils de probabilité en 3D.

Un modèle gaussien [BUL96] en deux dimensions (Figure II-13) représentant uniquement l'espace occupé rejoint ce modèle:

$$\mathbf{P}(\mathbf{r} | \mathbf{d}, \theta) = \frac{1}{2\pi\sigma_r\sigma_\theta} \exp \left[-\frac{1}{2} \left(\frac{(r-d)^2}{\sigma_r^2} + \frac{\theta^2}{\sigma_\theta^2} \right) \right]$$

où σ_r et σ_θ représentent les écarts types des erreurs radiales et angulaires du capteur. En l'absence de phénomènes parasites, les capteurs à ultrasons ont une faible incertitude sur la mesure, environ 1% de la mesure ce qui est de l'ordre de quelques centimètres. L'écart type radial étant liée à cette incertitude, ceci explique la forme pointue de la représentation. En revanche, l'ouverture angulaire du capteur liée à l'écart type angulaire est grande (ouverture du cône de l'ordre de 30°), ceci justifie la largeur de la courbe.

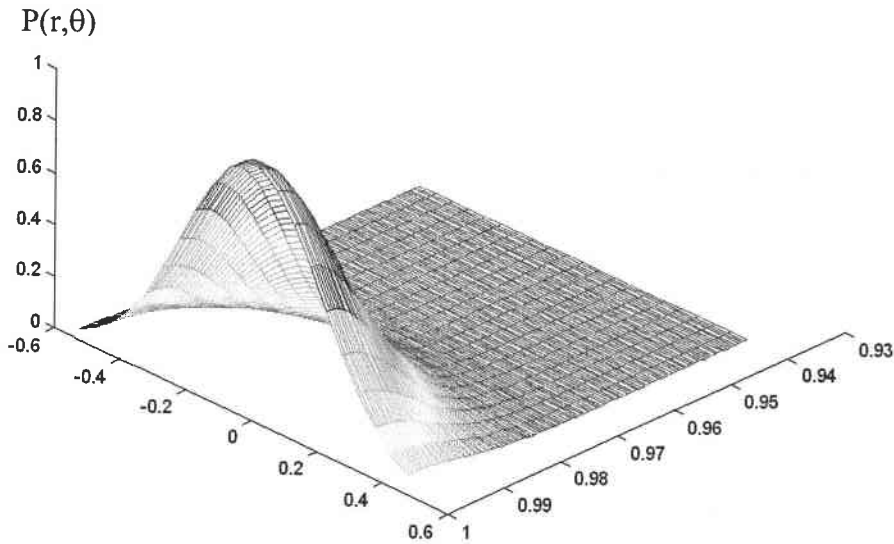


Figure II-13 : Exemple de modélisation d'un capteur à ultrasons par une gaussienne ($\sigma_r = 1\text{cm}$, $\sigma_\theta = 12^\circ$ et $d = 1\text{m}$) pour $r \in [0,1]$.

Ces derniers modèles favorisent l'axe acoustique par rapport à la périphérie du cône ultrasonore en affectant des valeurs plus grandes aux cellules situées au centre.

En considérant que sans informations complémentaires sur la mesure ultrasonore tous les points de l'arc sont équivalents pour une position possible de l'obstacle, il nous paraît logique d'attribuer la même valeur à chacune des cellules représentant le même état (libre ou occupé). Notre méthode utilise cette idée et attribue aux cellules une des trois valeurs définies par :

- $C_L = 0$: valeur d'une cellule sur laquelle aucune information n'est disponible,
- $C_L = 1$: valeur d'une cellule représentant un obstacle,
- $C_L = -1$: valeur d'une cellule représentant l'espace libre avant l'obstacle détecté.

Le modèle du capteur à ultrasons ainsi que la grille correspondante sont représentés par la figure II-14.

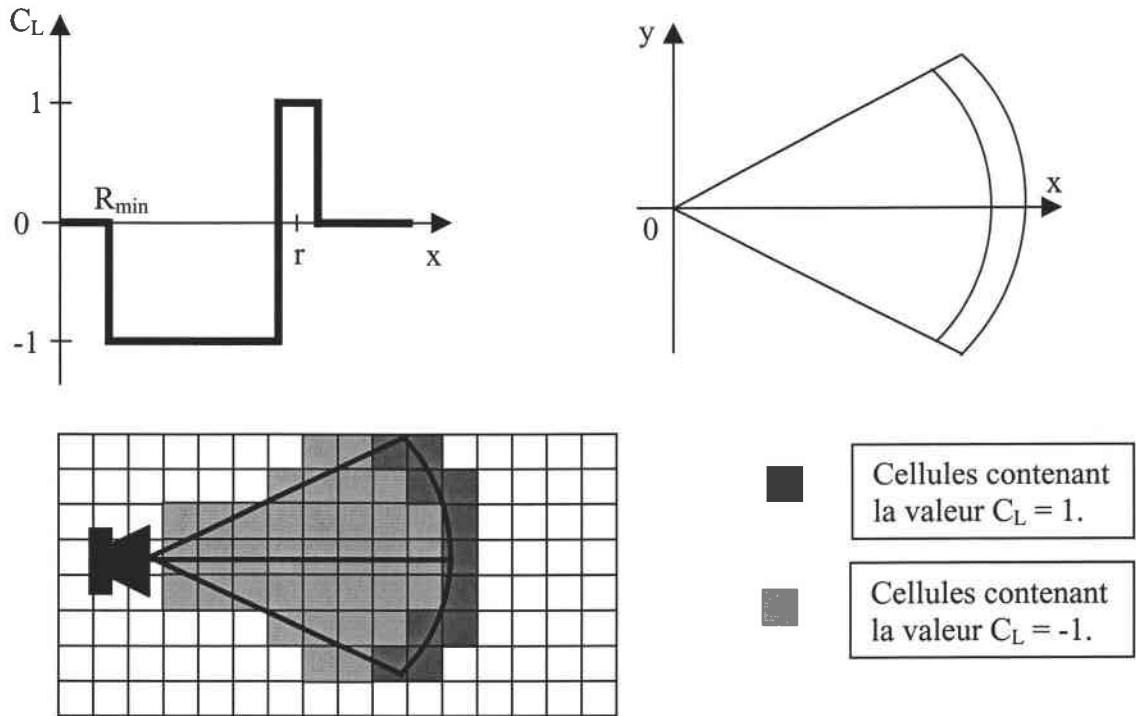


Figure II-14 : Modélisation du capteur à ultrasons.

Ce modèle a l'avantage d'éviter le traitement des mesures à l'aide de lois de probabilités lourdes à gérer. La zone occupée est une limite donnée par les capteurs à ultrasons, c'est pourquoi l'arc n'est réalisé que sur une épaisseur d'une ligne ; du point de vue du robot, la zone à l'intérieur des obstacles, bien qu'occupée est considérée comme inconnue. Une seule carte est nécessaire pour représenter l'espace occupé et l'espace libre ; la différence se fait sur le signe de la valeur de la cellule.

II.3.2 Création de la Grille Locale

La grille locale est centrée sur le robot et sa taille est ajustée sur la plus grande mesure ultrasonore détectée. Ces choix proviennent du fait que nous ne connaissons pas l'orientation du robot. En dimensionnant la taille de la grille sur la plus grande mesure cela permet d'avoir toutes les orientations possibles du robot sans devoir redéfinir la taille de la grille. La figure II-15 montre deux configurations différentes pour la même grille locale.

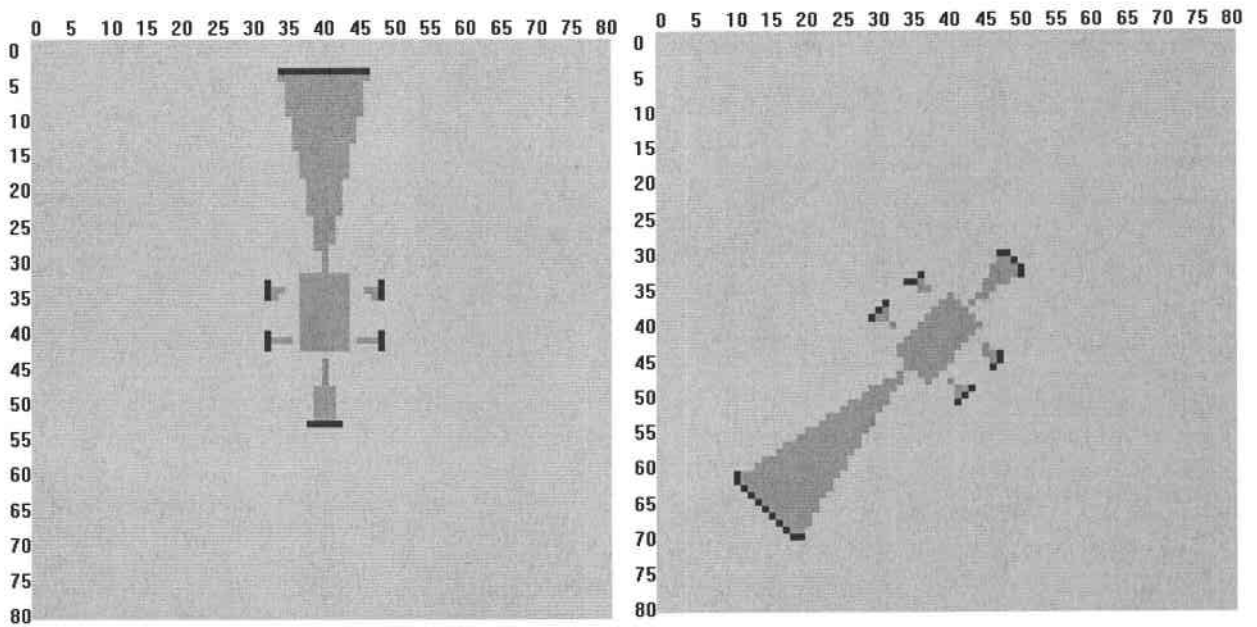


Figure II-15 : Exemple de deux configurations possibles du robot pour la même grille.

Nous différencions deux types de cellules dans la grille locale ; celles représentant le robot et celles correspondant aux mesures ultrasonores.

- Représentation de l'espace occupé par le robot

L'espace occupé par le robot correspond à toutes les cellules recouvertes par la surface au sol du robot (Figure II-16). Les coordonnées des deux cellules extrêmes d et f délimitant la taille du robot sont calculées, puis les cellules ayant leurs coordonnées comprises entre les deux extrêmes représentent le robot (Figure II-17). Comme nous sommes certains que le robot n'est pas placé sur un obstacle, nous affectons la valeur $C_L = -1$ aux cellules correspondantes.

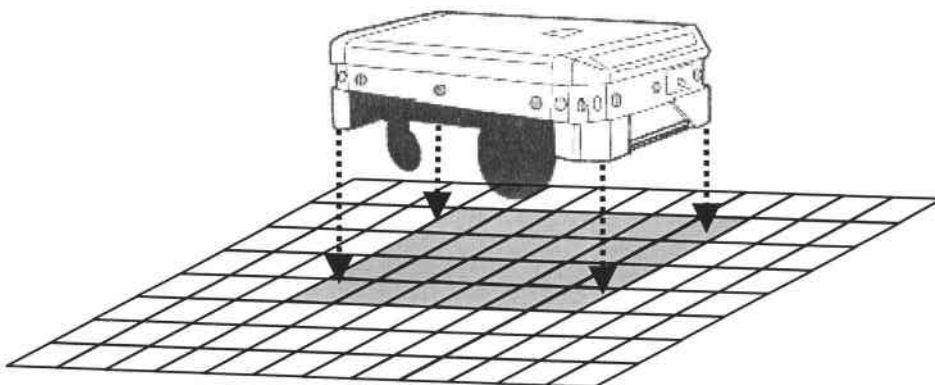


Figure II-16 : Projection de l'espace occupé par le robot sur la grille.

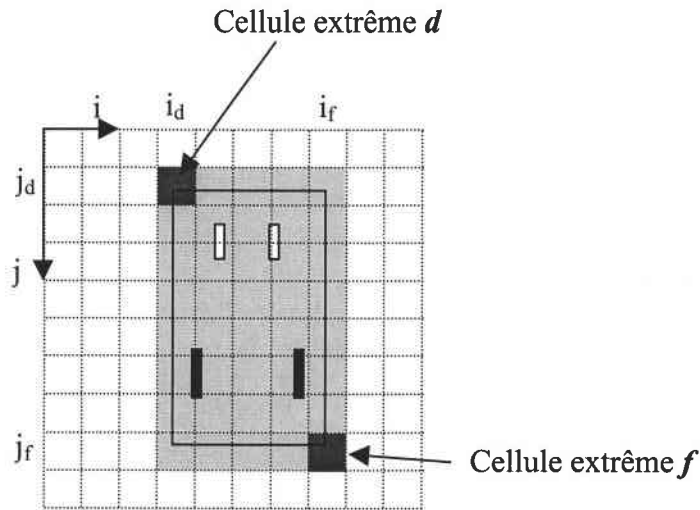


Figure II-17 : Cellules représentant le robot.

- Représentation des mesures ultrasonores

Les cellules de la grille situées dans le cône d'émission d'un capteur à ultrasons avant la distance mesurée ont leur valeur fixée à $C_L = -1$ pour représenter l'espace libre. Puis pour représenter l'obstacle, nous recherchons toutes les cellules se trouvant sur l'arc de cercle décrivant la limite du cône et nous leur donnons la valeur $C_L = 1$ (Figure II-18).

Ceci étant réalisé pour chaque capteur, nous obtenons une grille locale dont nous donnons un exemple à la figure II-19.

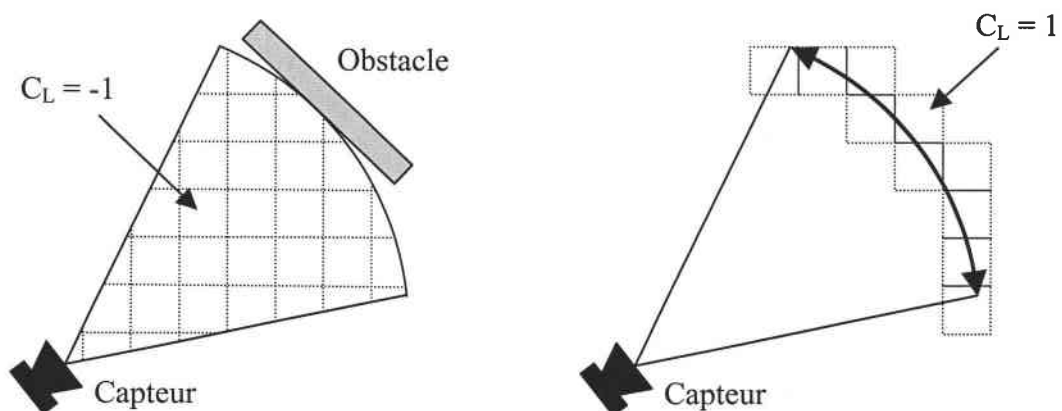


Figure II-18 : Cellules correspondant à une mesure d'un capteur à ultrasons.

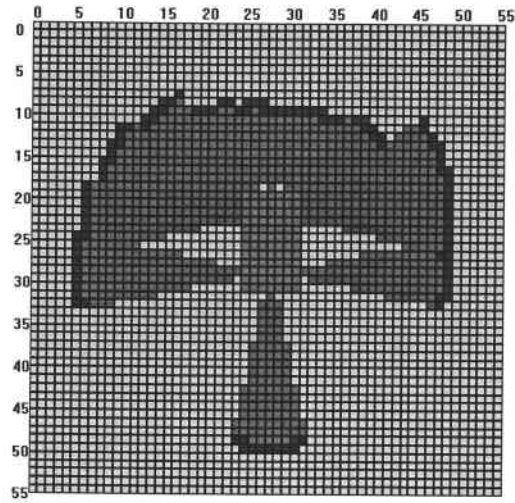


Figure II-19 : Exemple de grille locale (résolution de 10x10 cm par cellule).

II.3.3 Définition de la Grille Globale

La grille globale définit l'environnement dans lequel le robot va se mouvoir. Notre application étant destinée à un fauteuil électrique pour des personnes physiquement handicapées, l'environnement correspond à des pièces telles que celles rencontrées dans un appartement.

Le plan de l'environnement codé en une liste d'objets est fourni à l'application. Chaque objet est défini par cinq paramètres qui sont : son nom, les coordonnées X_0 et Y_0 de son origine (coin supérieur gauche), sa longueur LX et sa largeur LY (Figure II-20). Nous avons également codé des portes correspondant aux passages vers une zone non modélisée et définies par quatre paramètres qui sont : son nom, son mur d'appartenance (Nord, Est, Sud ou Ouest), la distance par rapport à l'origine du mur et sa largeur. Le premier objet de la liste définit la pièce de travail, sa taille (longueur et largeur) et le nombre d'objets et de portes s'y trouvant, et permet de définir la taille de la grille.

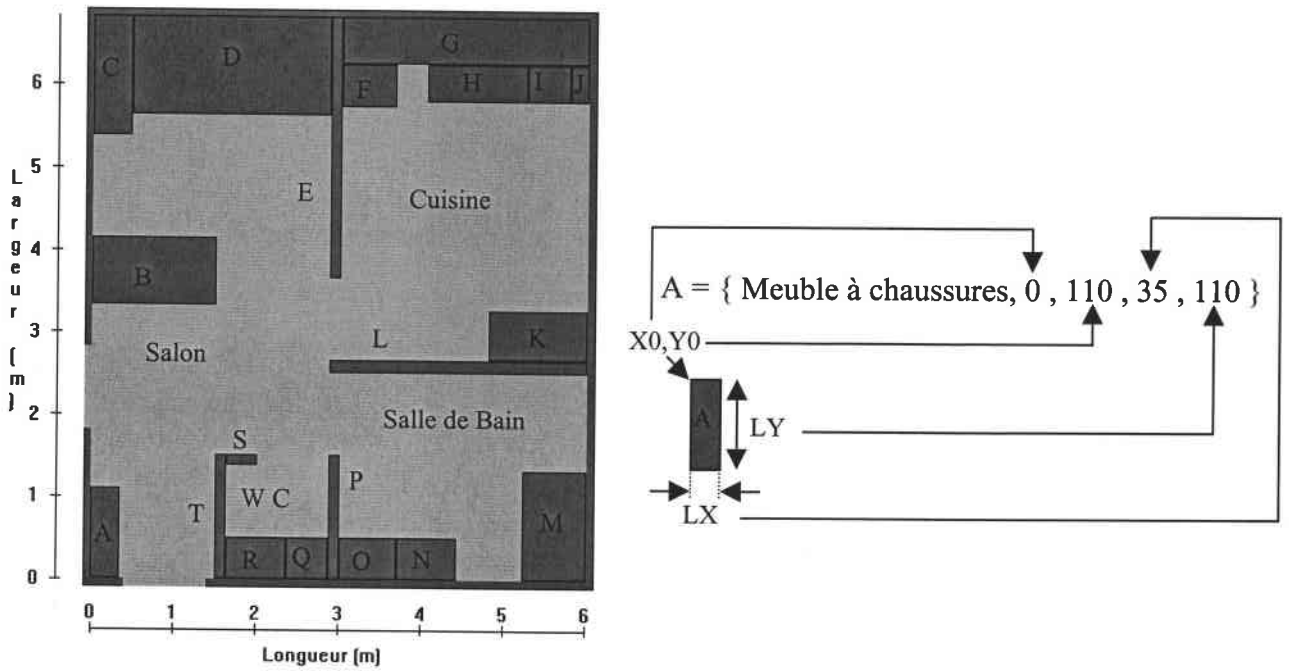


Figure II-20 : Exemple d'environnement et codage d'un objet.

La grille globale est construite suivant trois phases. La première consiste à initialiser les cellules de la grille par les valeurs $C_G = 0$. La seconde phase attribue les valeurs $C_G = -1$ à toutes les cellules représentant le sol de la pièce (sans les objets) puis entoure cette surface par des cellules contenant la valeur $C_G = 1$ représentant les murs délimitant la pièce. Puis, les portes sont placées sur ces murs en attribuant la valeur $C_G = -1$ aux cellules correspondantes. Enfin, la dernière phase ajoute les objets de la liste en attribuant la valeur $C_G = 1$ aux cellules recouvertes par chacun d'eux (Figure II-21).

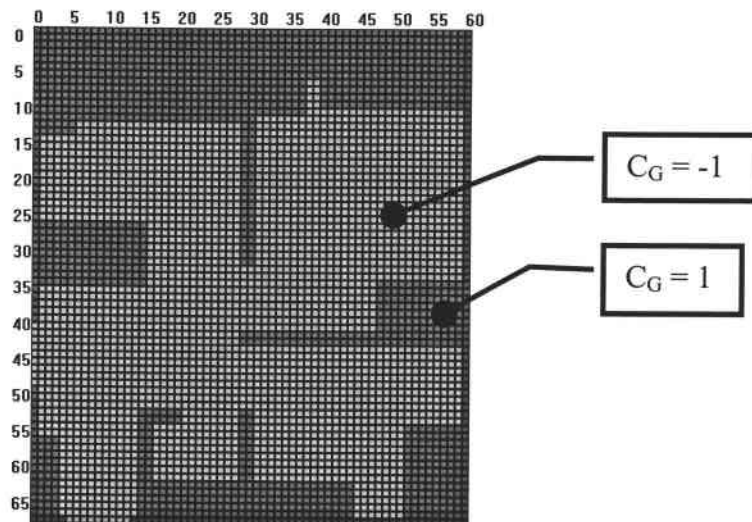


Figure II-21 : Grille globale (résolution de 10x10 cm par cellule).

II.4 MISE EN CORRESPONDANCE DES GRILLES

II.4.1 Principe

Le but de la localisation est de déterminer la position et l'orientation du robot dans l'environnement, ou encore, le robot étant au centre de la grille locale, la position de la grille locale dans la grille globale. Partant d'une position et orientation donnée (Figure II-22), nous recherchons la transformation \mathbf{F} se décomposant en une translation \mathbf{T} et une rotation \mathbf{R} , qui fera coïncider au mieux les deux grilles.

Un point \mathbf{p} dans le repère \mathbf{L} lié au robot donne un point \mathbf{q} dans le repère \mathbf{G} lié à l'environnement par :

$$\mathbf{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \mid \mathbf{q} = \mathbf{F}(\mathbf{p}) = \mathbf{T} + \mathbf{R} \cdot \mathbf{p}$$

avec $\mathbf{q} = [x_G, y_G]^T$: point dans le repère global \mathbf{G} ,

$\mathbf{p} = [x_L, y_L]^T$: point dans le repère local \mathbf{L} ,

$\mathbf{T} = [a, b]^T$: matrice de translation,

$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ -\sin \theta & -\cos \theta \end{bmatrix}$: matrice de rotation dont l'axe est situé sur le robot au centre de la grille locale (voir Annexe 1).

La position du robot est estimée en trouvant l'emplacement de la grille locale dans la grille globale ce qui revient à calculer \mathbf{a} et \mathbf{b} ; l'orientation du robot quant à elle est évaluée en mesurant l'angle de rotation θ de la grille locale par rapport à la grille globale (Figure II-22). Pour trouver \mathbf{a} , \mathbf{b} et θ , il faut donc déterminer parmi toutes les translations \mathbf{T} et rotations \mathbf{R} possibles de la grille locale par rapport à la grille globale la translation \mathbf{T}_M et la rotation \mathbf{R}_M qui maximisent le critère de mise en correspondance.

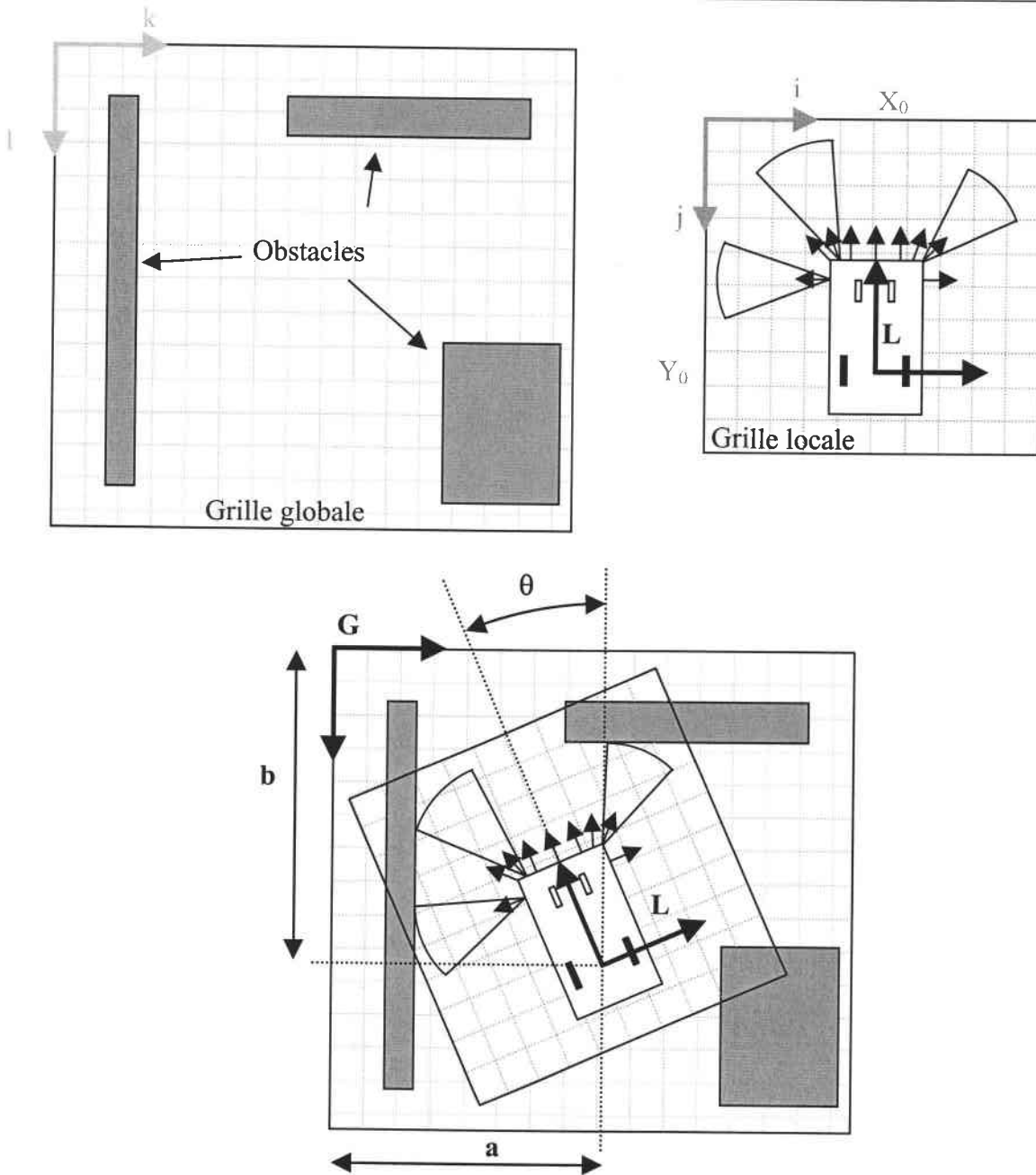


Figure II-22 : Mise en correspondance des grilles.

II.4.2 Critère de mise en correspondance

Considérons une translation T et une rotation R données. Soit $C_L(i, j)$ une cellule de la grille locale. Par la transformation F , cette cellule se superpose à une cellule $C_G(k, l)$ de la grille globale. Soit P le produit des valeurs des deux cellules superposées. Le signe de P indique si les deux cellules ont le même état (occupé ou libre) et donc s'il y a bonne correspondance entre les cellules ou non (Figure II-23).

$C_L \backslash C_G$	Occupée	Libre	Inconnue
Occupée	>0	<0	0
Libre	<0	>0	0
Inconnue	0	0	0

Figure II-23 : Signe du produit des cellules.

Le produit est effectué sur toutes les cellules de la grille locale, la somme S de tous ces produits est calculée et constitue le critère de mise en correspondance. Il informe sur une bonne superposition des grilles puisqu'il est maximal quand le plus grand nombre de cellules de même nature sont superposées.

La position du robot et son orientation seront données par a , b et θ pour lesquels la somme $S(a,b,\theta)$ est maximale, avec :

- $$S(a,b,\theta) = \sum_{i,j \in \text{grille locale}} C_L(i,j) \cdot C_G(k,l)$$
- $$\begin{cases} k = a + (i-i_0) \cdot \cos(\theta) + (j-j_0) \cdot \sin(\theta) \\ l = b - (i-i_0) \cdot \sin(\theta) + (j-j_0) \cdot \cos(\theta) \end{cases}$$
- i_0 et j_0 sont les indices de la cellule centrale de la grille locale (cellule contenant l'origine du repère L et sur laquelle est centrée la rotation) (voir Annexe 1).
- k et l sont les indices de la cellule de la grille globale couverte par la cellule d'indices i,j de la grille locale.

k et l doivent être entiers, or la relation ci-dessus fournit des valeurs réelles. Il est alors nécessaire de préciser la cellule qui servira à réaliser la mise en correspondance. Comme le montre la figure II-24, une cellule de la grille locale sur laquelle a été appliquée la rotation ne coïncide pas exactement avec une seule cellule de la grille globale mais en chevauche plusieurs. Pour notre application, nous levons l'ambiguïté en considérant qu'une cellule de la grille globale est associée à

la cellule de la grille locale qui couvre son point central, ce qui revient à arrondir k et l à l'entier le plus proche.

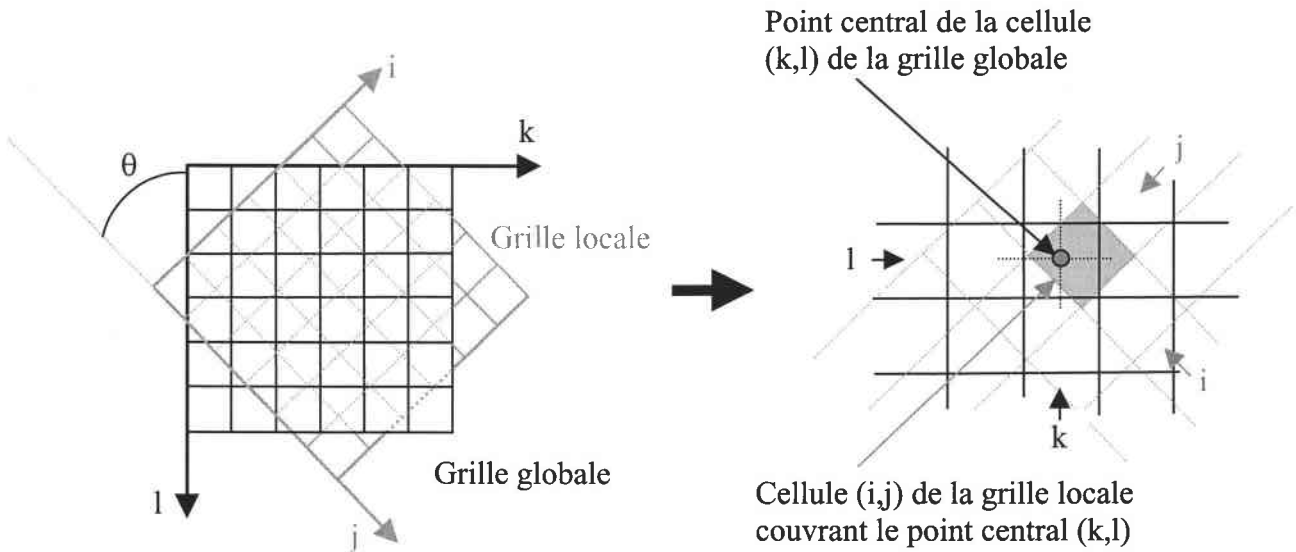


Figure II-24 : Mise en correspondance des cellules.

II.4.3 Zone de recherche

La seule information disponible au début de la trajectoire, en plus des mesures des capteurs à ultrasons, est une information vague de la position du robot. Cette information provient de l'utilisateur qui l'exprime par une formulation simple du type "se situe à l'entrée de la cuisine", "se situe près du téléviseur"... A partir de cette indication, une position et une orientation approximative du robot sont estimées. Nous effectuons ensuite la recherche de la mise en correspondance dans une zone de ± 75 cm autour de la position estimée et pour des orientations comprises dans un cône de ± 45 degrés autour de l'orientation estimée (Figure II-25). Ces valeurs correspondent à une précision raisonnable que peut donner l'utilisateur sur sa position. Dans le cas où la procédure est employée suite à un échec de la localisation dynamique, elles fournissent un intervalle suffisant pour englober la position odométrique du robot. La définition de cette zone de recherche permet de limiter le temps requis pour la localisation, en effet sans cette indication il faudrait calculer le critère de mise en correspondance pour toutes les positions et orientations possible du robot dans la pièce ou même dans l'appartement.

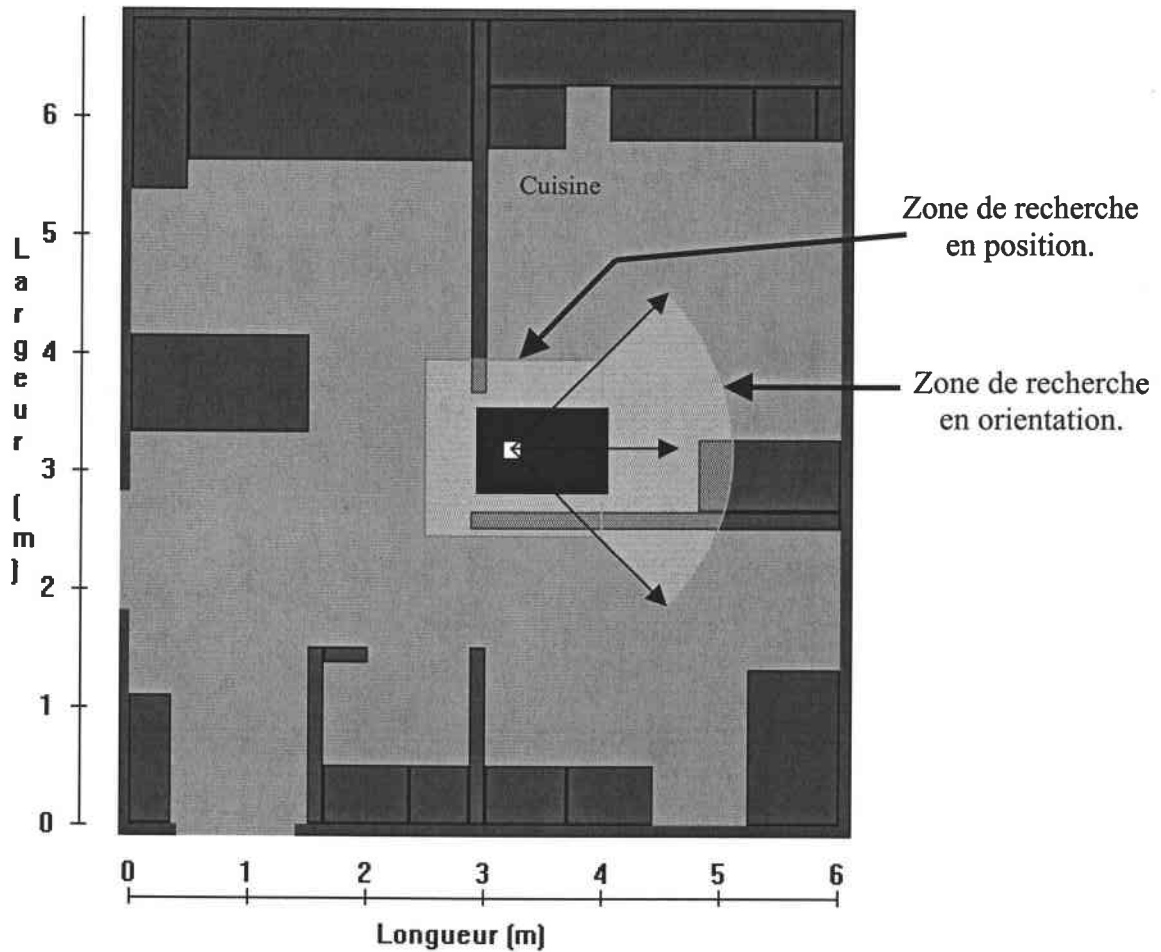
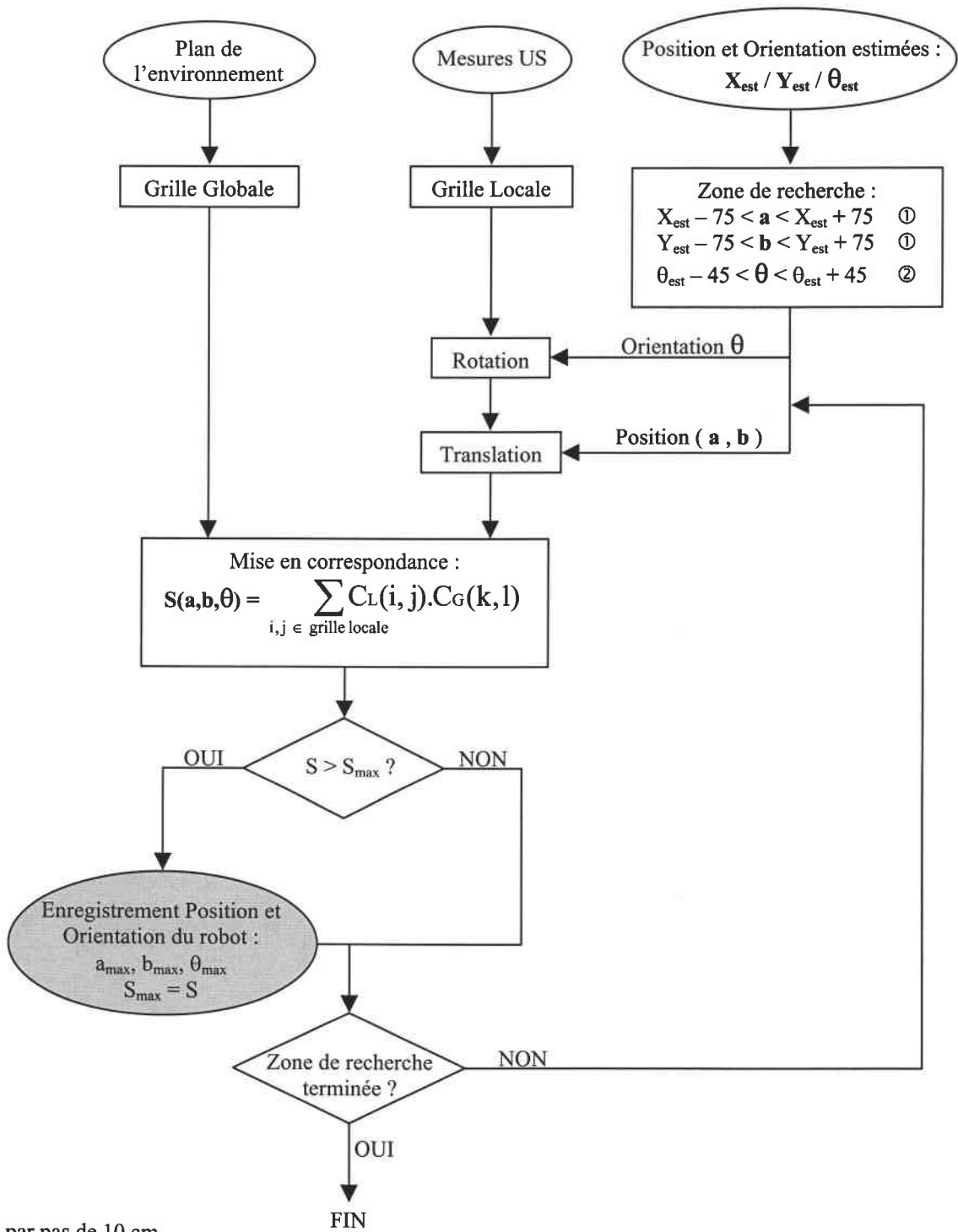


Figure II-25 : Exemple de zones de recherche déduites à partir de l'information "se situe à l'entrée de la cuisine".

Il est aussi possible d'adapter la zone de recherche en fonction de l'information fournie. Dans le cas où l'information provient du module de navigation, l'estimation plus précise de la position du robot peut permettre de réduire la zone de recherche. De même, lors d'un franchissement de porte nous savons que la porte est juste derrière, nous pouvons donc limiter la taille de la zone de recherche en position à la partie située en avant de la porte et réduire la zone de recherche en orientation à quelques degrés.

II.4.4 Algorithme

La figure II-26 présente l'algorithme de localisation que nous avons réalisé. La somme S_{\max} est initialisée à 0, puis chaque fois qu'une somme $S(a,b,\theta)$ est supérieure, la position et l'orientation du robot sont enregistrées et la valeur de S constitue la nouvelle valeur de S_{\max} .



- ① par pas de 10 cm
- ② par pas de 5°

Figure II-26 : Algorithme de localisation.

II.5 RESULTATS PRELIMINAIRES

II.5.1 Simulation

Nous avons, dans un premier temps, testé les performances à partir de mesures simulées des capteurs à ultrasons. C'est à dire qu'à partir de la position et de l'orientation du robot dans l'environnement, nous réalisons pour chaque capteur, un lancer de rayon décrivant le cône ultrasonore (Figure II-27).

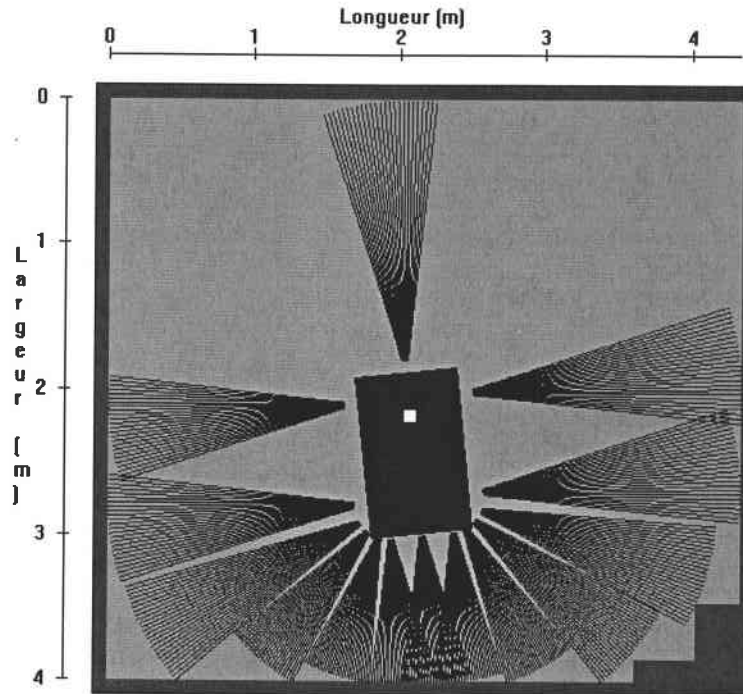


Figure II-27 : Simulation d'une mesure des capteurs à ultrasons.

En supposant que les surfaces sont suffisamment rugueuses pour être détectables quel que soit l'angle d'incidence de l'onde et qu'il n'y a pas de phénomènes de réflexions multiples ou de diaphonie, la méthode de détection d'obstacles prend en compte le premier objet détecté.

Le cône de chaque capteur est décrit par des droites orientées tous les degrés (itération sur α) elles-mêmes constituées de points espacés tous les centimètres (itération sur d) dont les coordonnées sont :

$$\begin{cases} x = x_r - r_c * \sin(\beta_c + \theta_r) - d * \sin(\theta_c + \alpha + \theta_r) \\ y = y_r + r_c * \cos(\beta_c + \theta_r) + d * \cos(\theta_c + \alpha + \theta_r) \end{cases}$$

r_c , β_c et θ_c sont les paramètres définissant la position des capteurs par rapport au robot.

x_r , y_r et θ_r sont la position et l'orientation du robot dans l'environnement.

$\alpha \in [-\alpha_c, \alpha_c]$, α_c est l'angle d'ouverture du cône (Figure II-28).

Un test vérifie si les coordonnées du point du cône sont situées sur un obstacle. Dans ce cas, la distance d est retournée comme étant la distance mesurée par le capteur.

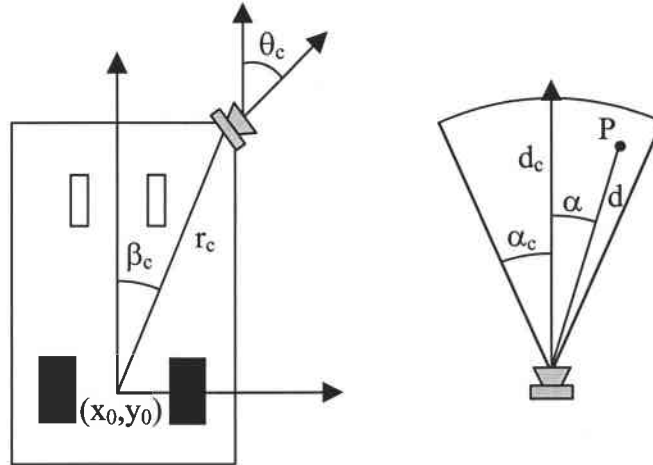


Figure II-28 : Configuration du cône d'émission d'un capteur à ultrasons.

Un exemple de localisation est donné à la figure II-29. Le résultat présenté est obtenu à partir de grilles d'une résolution de 12x12 cm par cellule. La partie de gauche montre le robot sur sa position réelle ainsi que les mesures simulées des capteurs à ultrasons ; la partie de droite le résultat de la localisation et la position estimée du robot. Les positions sont représentées par les indices des cellules ($X=23$ correspond à la position $X=23 \times 12$ cm dans la pièce). L'angle Teta du robot est orienté par rapport à un axe dirigé vers le haut. La position estimée servant à définir la zone de recherche, a été placée avec plus de 70 cm d'erreur en X, plus de 20 cm d'erreur en Y et près de 15° d'erreur en orientation par rapport à la position et l'orientation réelle. La valeur délivrée par l'algorithme recouvre précisément la valeur exacte, l'erreur subsistant sur l'orientation (-69° au lieu de -70°) n'est pas représentative, en effet le pas de recherche en orientation est de 3° car la représentation par grille ne permet pas une meilleure précision angulaire.

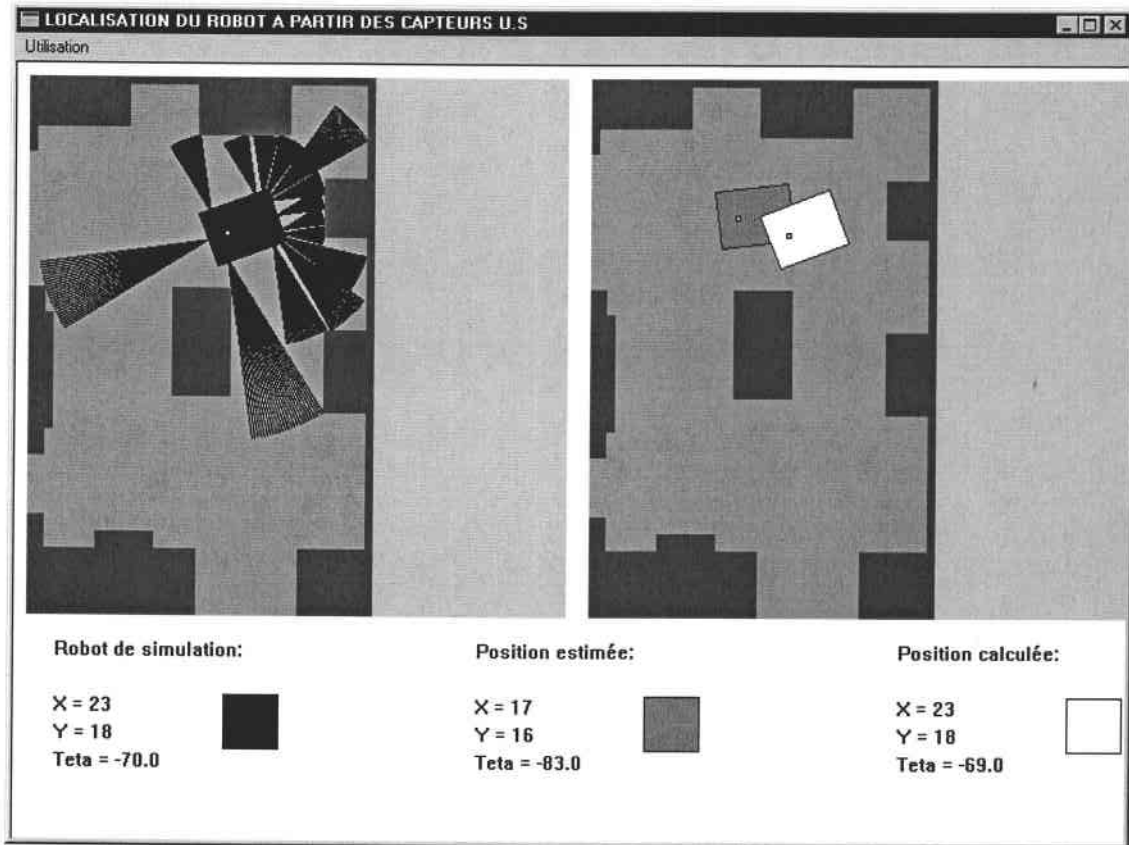


Figure II-29 : Exemple de localisation réalisée en simulation.

II.5.2 Tests réels

Nous avons ensuite réalisé des tests à partir de mesures réelles sur la base mobile Robuter. La position exacte du robot est relevée afin de valider les résultats obtenus, et la zone de recherche est définie de manière à inclure cette position. Ces essais ont été réalisés dans une pièce de petite taille (3*3 m²) avec des grilles d'une résolution de 5cm par cellule.

Un exemple illustre la localisation en conditions réelles (Figure II-30). La grille initiale (à gauche) présente le robot sur la position réelle et la grille finale (à droite) montre le robot sur la position calculée par l'algorithme. Ainsi, nous avons pu constater que les mesures délivrées par les capteurs à ultrasons correspondent pour la plupart à l'obstacle le plus proche et que l'algorithme de localisation est adapté pour une utilisation en conditions réelles.

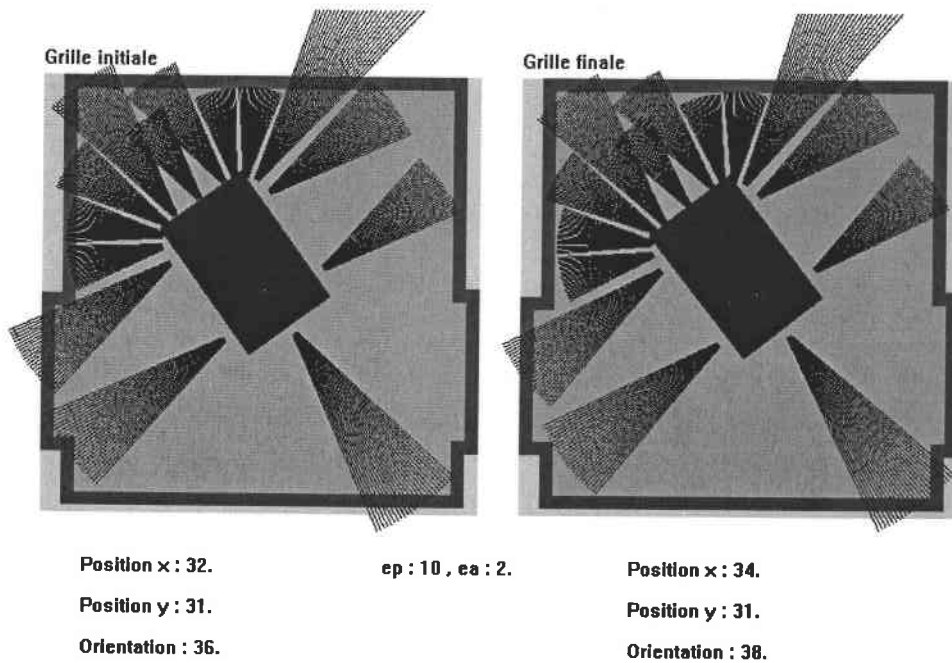


Figure II-30 : Exemple de localisation réalisée en conditions réelles.

Dans cet exemple, l'erreur en position (ep) est de 10 cm soit de deux cases et comme dans le cas de la simulation l'erreur en orientation (ea=2°) n'est pas représentative avec un pas de recherche en orientation de 3°. La mesure du capteur avant droit sort de l'environnement, ce qui révèle la présence d'une spécularité. Lors de la mise en correspondance des deux grilles, les valeurs des cellules sortant de la grille locale sont multipliées par la valeur nulle des cellules inconnues de la grille globale. Ces mesures n'influent donc pas sur le critère de mise en correspondance pour une position donnée. Cependant l'algorithme va essayer de placer les nombreuses cellules vides sur un espace libre afin d'augmenter la valeur du critère qui peut alors être meilleur dans une position erronée que dans la position réelle. Dans ces conditions le résultat peut être faussé. Nous envisagerons les moyens de détecter de telles mesures ultérieurement.

L'exemple suivant (Figure II-31) présente le robot pour une autre configuration où malgré plusieurs mesures erronées, la localisation trouvée est exacte.

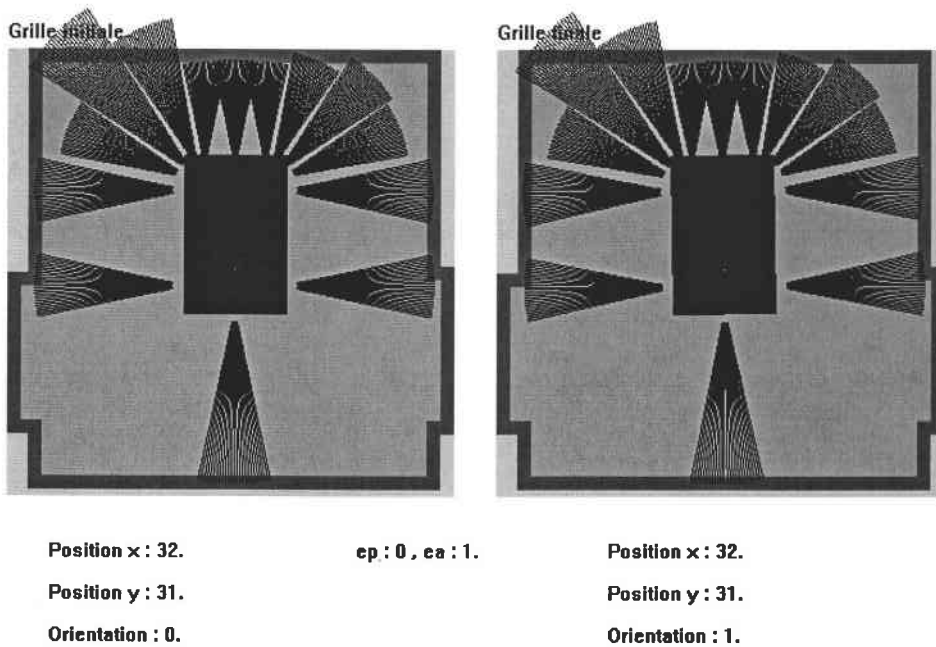


Figure II-31 : Autre exemple de localisation du VAHM.

II.6 CONCLUSION

Dans ce chapitre nous avons expliqué comment à partir des mesures de quatorze capteurs à ultrasons disposés en ceinture autour du robot nous construisons une grille locale. Nous avons ensuite décrit de quelle manière nous obtenons une grille globale à partir d'un plan de l'environnement. Enfin, nous avons détaillé la façon de mettre en correspondance les deux grilles pour déterminer la position et l'orientation du robot.

L'efficacité de la méthode est montrée dans nos essais en simulation où nous retrouvons la position réelle du robot à partir d'une position approximative qui en est éloignée. Les exemples d'essais réalisés sur la base ont également prouvé que l'algorithme peut être utilisé en conditions réelles.

Partant de cette trame, nous allons dans le chapitre suivant établir un affinement des modèles employés pour la création des grilles en vue d'optimiser la méthode.

AFFINEMENT DES
MODELES

III AFFINEMENT DES MODELES

III.1 INTRODUCTION

Dans le chapitre précédent nous avons expliqué le principe de la localisation par une méthode de mise en correspondance de grilles. Au cours de ce chapitre nous étudions plus précisément les valeurs qu'il faut attribuer aux cellules des deux grilles pour deux raisons : d'une part pour obtenir une meilleure cohérence de la grille locale afin de bien représenter les mesures des capteurs à ultrasons, d'autre part pour que les grilles soient bien appropriées pour la mise en correspondance. De nombreux essais réalisés en simulation et sur le robot illustrent les modifications apportées. La résolution des grilles employées est de 10x10 cm par cellule. La recherche de la meilleure correspondance entre la grille locale et la grille globale est réalisée sur une surface de ± 75 cm autour d'une position estimée et pour des orientations comprises dans un cône de ± 45 degrés autour d'une orientation estimée. Le pas de recherche est d'une cellule pour la position et de 5 degrés pour l'orientation. Par ailleurs nous nous assurons que la position réelle se trouve dans la zone de recherche. Pour notre application, nous avons fixé les conditions pour une bonne localisation à :

- une erreur maximale en position d'une cellule (± 10 cm),
- une différence maximale en orientation de 7° .

La localisation du robot sera alors considérée comme réussie si les deux conditions sont satisfaites. Ces valeurs correspondent à une limite pour que la planification de la trajectoire puisse être réalisée et que le module de localisation dynamique puisse ensuite corriger la position du robot lors de la navigation.

III.2 CHOIX DES VALEURS DES CELLULES DE LA GRILLE LOCALE

[COU98a]

III.2.1 Cône d'un capteur à ultrasons

Dans nos travaux préliminaires nous affectons des valeurs entières aux cellules de la grille locale. Ces valeurs sont $C_L = 0$ pour indiquer l'espace inconnu, $C_L = -1$ pour représenter l'espace libre autour du robot et $C_L = 1$ pour décrire la zone dans laquelle un obstacle a été détecté. Ces travaux ont permis de tester la méthode et d'en évaluer les limites. Par exemple, la présence d'une grande mesure de distances à côté de mesures plus petites peut dans certains cas mettre en défaut l'algorithme. L'exemple de la figure suivante (Figure III-1) présente le robot dans un environnement représentant une salle à manger et dans une configuration telle qu'il existe une grande différence entre les mesures.

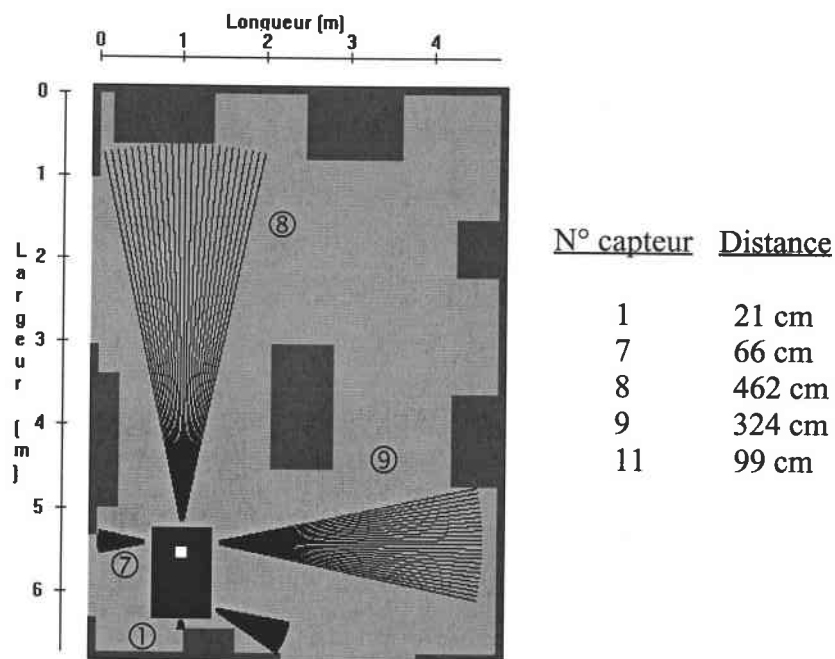
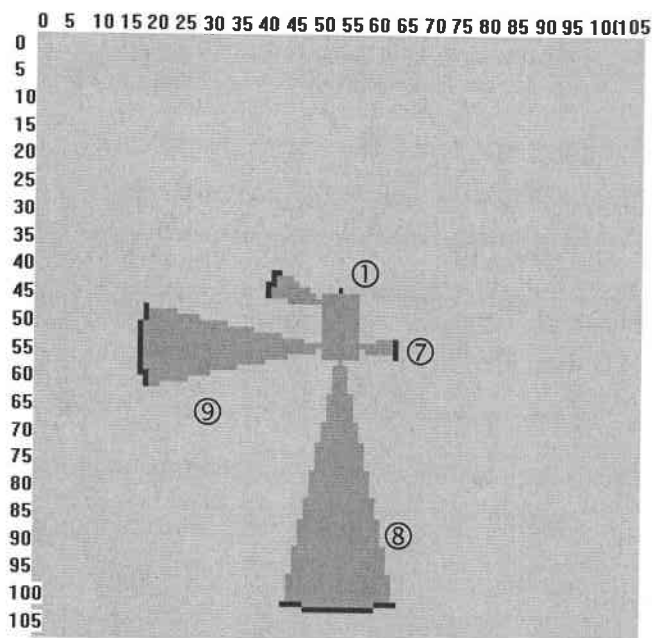


Figure III-1 : Configuration induisant de grandes différences entre les mesures.

Une grande mesure de distance recouvre plus de cellules qu'une petite mesure. En donnant la même valeur à toutes les cellules de chaque cône qu'il soit grand ou petit, la contribution d'une grande mesure pour la localisation sera alors plus importante. Cette idée ne correspond pas au

modèle du capteur à ultrasons pour lequel une grande distance a une incertitude plus grande sur la mesure et risque davantage d'être liée à des problèmes de spécularité qu'une petite distance. Ainsi, quand une grande distance est présente, l'algorithme de localisation place le robot dans une position telle que le nombre de cellules (de la grille locale) mises en bonne correspondance soit maximal, ce qui revient à placer en premier lieu la grande mesure qui, bien qu'étant exacte, ne devrait pas être prépondérante. L'analyse de la grille locale met en évidence l'importance des grands cônes en présentant pour chacune des mesures de l'exemple précédent le nombre de cellules couvrant son espace libre et celui représentant sa couronne (Figure III-2).



N° capteur	Nombre de cellules	
	libres	occupées
1	2	1
7	15	4
8	490	21
9	252	16
11	29	7

Figure III-2 : Poids de chaque mesure dans la grille locale.

Pour réduire l'importance des grandes mesures lors de la localisation, une première approche a été de modifier le poids des petites mesures en donnant à leurs cellules représentant l'espace libre des valeurs plus fortes. Ainsi, nous avons calculé la surface (en nombre de cellules) couverte par chaque secteur pour ensuite pondérer la valeur de chaque cellule par la surface la plus grande :

$$C_L = - \frac{\text{Surface maximale}}{\text{Surface du capteur}}$$

Par exemple, si le plus grand secteur de l'exemple précédent recouvre 490 cellules contenant chacune la valeur $C_L = -1$, les deux cellules du secteur le plus petit auront la valeur $C_L = -245$, ce qui donne le même poids pour la localisation aux deux secteurs.

Ce modèle permet de corriger le poids très important des secteurs des grandes mesures par rapport aux mesures plus petites, mais la grille locale n'est pas homogène. En effet, pour une mesure donnée, les cellules représentant un obstacle n'ont pas le même poids que celles représentant l'espace libre et lors de la mise en correspondance, une importance plus grande sera donnée aux cellules libres. Pour rendre la grille homogène il faut donc que, pour une mesure donnée, les cellules représentant l'obstacle détecté aient globalement le même poids que celles indiquant l'espace libre. Pour cela nous avons travaillé directement avec le nombre de cellules dans les espaces libre et occupé. Par exemple, pour une petite mesure, l'arc est contenu dans une seule cellule, cela signifie qu'elle contient nécessairement le point produisant l'écho ; la valeur 1 peut donc lui être attribuée. Pour de plus grandes mesures, l'arc s'étend sur plusieurs cellules ; la valeur précédente doit donc être uniformément répartie sur l'ensemble de ces cellules. Dans cette logique, n cellules appartenant à un même arc prendront chacune la valeur $1/n$.

En appliquant ce principe, nous définissons la nouvelle valeur C_L d'une cellule de la grille locale comme étant inversement proportionnelle au nombre de cellules couvertes par l'ensemble (libre ou occupé) auquel elle appartient. Par conséquent, pour chaque cône représentant une mesure ultrasonore, nous donnons aux n_1 cellules de l'arc la valeur $C_L = 1/n_1$ et aux n_2 cellules du secteur la valeur $C_L = -1/n_2$, tandis que la valeur des cellules inexplorées est maintenue à 0 (Figure III-3). Ainsi, chaque capteur a une contribution équivalente égale à 2 (en valeur absolue) pour la localisation quelle que soit la mesure.

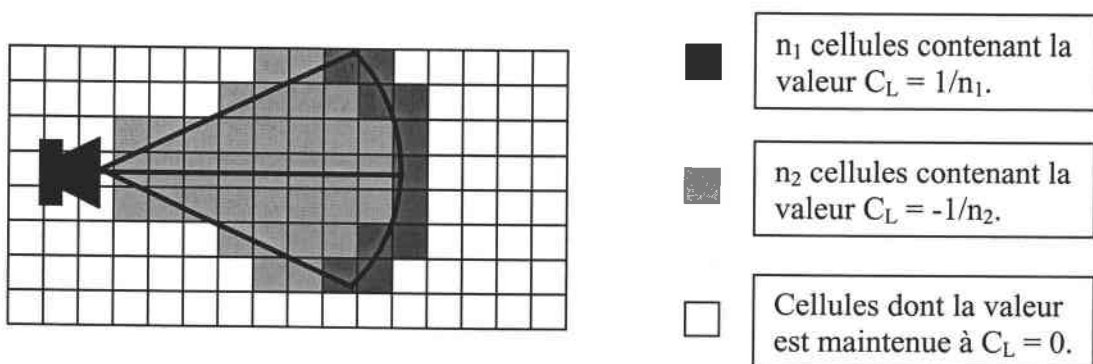


Figure III-3 : Correction de la modélisation du capteur à ultrasons.

Pour tester ce nouveau modèle, nous avons réalisé des essais à la fois à partir de mesures simulées et de mesures réelles dans une pièce vide de 4,3 mètres de long sur 4 mètres de large. Le robot est alors placé dans la pièce pour 34 configurations différentes dont quelques-unes sont représentées dans la figure III-4. En noir sont représentées les positions réelles du robot, en gris les positions calculées à partir des mesures simulées et en blanc les positions calculées à partir des mesures des capteurs sur le robot. Quand seule la flèche blanche est visible c'est que les trois positions se superposent parfaitement. De même, quand les deux positions calculées (en simulation et à partir des mesures des capteurs) sont identiques, la flèche grise n'apparaît pas.

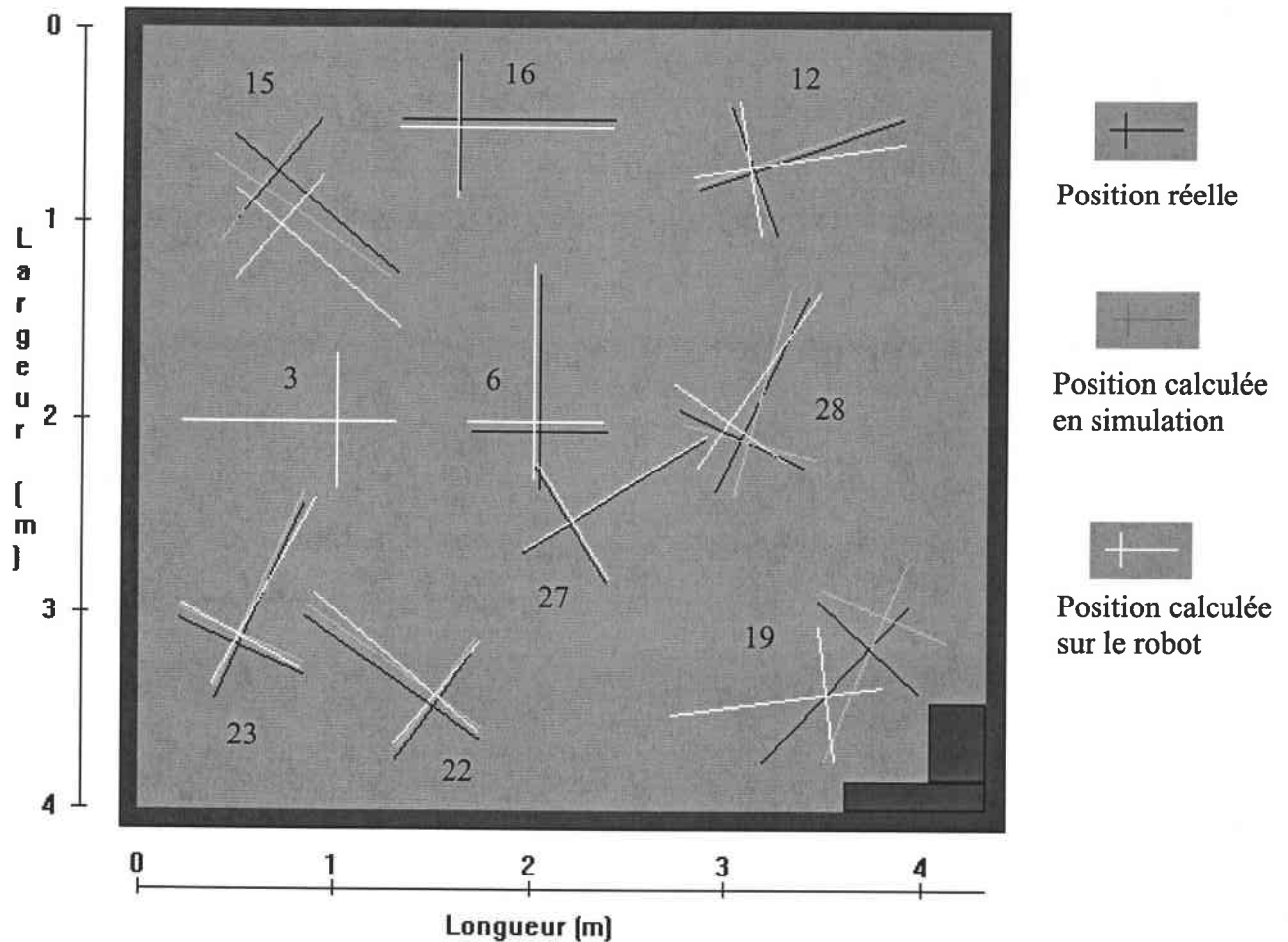


Figure III-4 : Exemple de quelques configurations.

A partir des mesures simulées, 29 configurations sur les 34 testées répondent aux conditions d'une bonne localisation qui sont une erreur maximale en position d'une cellule ($\pm 10\text{cm}$) et une différence maximale en orientation de 7 degrés. Sur le robot, avec les mesures des capteurs à ultrasons, seulement 22 d'entre elles passent ces conditions (Annexe 2). Ceci nous amène à deux

interrogations à savoir pourquoi les essais en simulation n'ont pas plus de réussite et pourquoi les essais sur le robot n'ont pas un taux de réussite comparable à celui de la simulation.

Ce dernier point a été analysé en comparant les essais fournissant un bon résultat en simulation et une mauvaise localisation en conditions réelles. Ces mauvais essais peuvent être expliqués par le fait que certains capteurs à ultrasons renvoient des distances plus grandes que les distances simulées par suite de réflexions multiples. L'algorithme tend alors à ramener les mesures sortant de l'environnement vers l'intérieur de la pièce où la superposition est prise en compte. Il apparaît aussi des mesures trop petites provenant des phénomènes de diaphonie. L'exemple de la figure III-5 montre ces deux phénomènes en comparant les mesures issues de la simulation qui fournissent un bon résultat et celles prises sur le robot qui induisent une erreur d'orientation de 10 degrés.

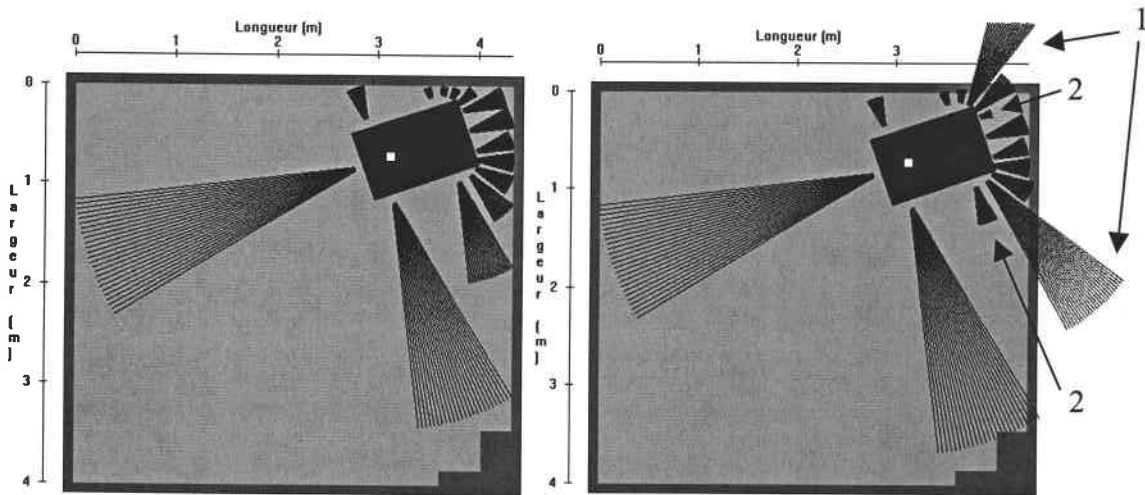


Figure III-5 : Mesures simulées et mesures réelles (1 : réflexions multiples, 2 : diaphonie) de la configuration n°12.

Par ailleurs, l'analyse des échecs en simulation (par exemple la configuration n°19 de la figure III-4), montre que ceux-ci sont liés à l'influence trop forte des cellules couvertes par la surface du robot. Dans la partie suivante nous établirons le mode de résolution de ce problème.

III.2.2 Surface du robot

Après avoir discrétisé les données ultrasonores, nous avons à traiter de la zone du robot. Cette zone correspond à toutes les cellules couvertes par la forme du robot. Chacune de ces cellules doit avoir une valeur C_L indiquant qu'elle est libre (voir chapitre II.3.2). Avec une résolution de la grille de 10x10 cm par cellule cette surface recouvre plus de 80 cellules. Comme nous sommes certains que ces cellules sont libres, dans nos essais préliminaires nous leur avons attribué la valeur -1 , ce qui fait un poids de 80. Suite à la correction des valeurs des cellules des cônes ultrasonores, l'espace libre décrit par les mesures contribue à la localisation avec un poids de 14. Ainsi, les cellules représentant le robot ont un poids six fois plus important que celles issues des mesures. L'exemple de la figure III-6 illustre les conséquences de ce déséquilibre en montrant à gauche le robot sur sa position réelle et à droite le robot sur sa position calculée. Bien que certains cônes sortent de la pièce dans la position issue de l'algorithme, le critère de mise en correspondance (S) est plus élevé pour cette position (89.43) que pour la position réelle (87.22). En décomposant les sommes (S), la contribution des capteurs sur la position réelle est de 7,22 contre 80 pour la surface du robot, et elle est de 6,43 contre 83 pour la position calculée. Deux remarques ressortent de cette analyse : la première est que les mesures n'ont pas un poids suffisant (moins de 10% de la somme) ce qui les rendent négligeables ; la seconde est que la contribution des cellules représentant la surface du robot change suivant la configuration du robot. En effet, les sommes précédentes nous indiquent que le nombre de cellules couvertes par le robot sont au nombre de 80 pour l'orientation réelle du robot et de 83 pour l'orientation calculée, ce qui explique pourquoi l'algorithme choisit cette orientation. Ce point touche le problème des rotations de grilles discrètes qui seront détaillées plus loin (voir chapitre III.3.3).

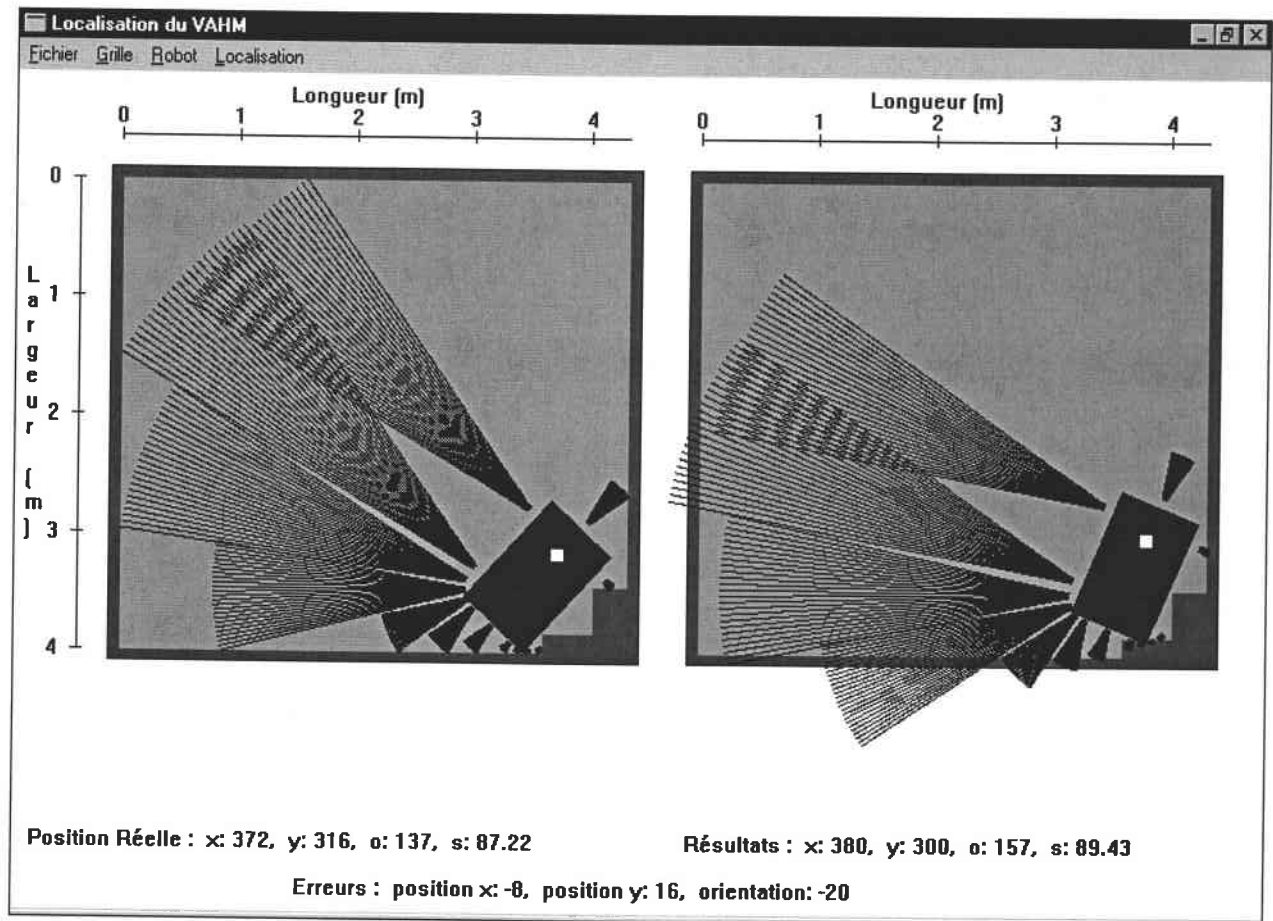


Figure III-6 : Résultat de la localisation simulée pour la configuration n°19.

Dans ce cas, les valeurs trop fortes des cellules de la surface du robot ont induit un résultat faux. Dans tous les cas, il est gênant que le poids relatif à la surface du robot soit très supérieur au poids des mesures car ce sont elles qui déterminent la localisation. Il faut donc diminuer l'influence des cellules de la surface du robot et définir le poids de cette surface de manière à ce qu'il soit indépendant du nombre de cellules qu'elle couvre.

Compte tenu de l'importance relative des mesures et de la certitude que le robot se trouve sur un espace libre, il nous paraît justifié de donner le même poids à l'ensemble des cellules représentant l'espace libre des mesures des capteurs et à celles décrivant le robot. Ainsi, nous affectons la valeur $C_L = -c/n$ à chaque cellule représentant le robot où c est le nombre de capteurs (14 pour nous) et n est le nombre de cellules couvertes par la surface du robot ; ce qui nous a permis de corriger l'exemple précédent (Figure III-7). On remarquera que la somme est abaissée à la valeur de 20,56 dont la répartition est 13,34 pour les cellules de la surface du robot et 7,22 pour les mesures des capteurs à ultrasons ; la valeur non entière de la surface du robot est due au problème de rotation.

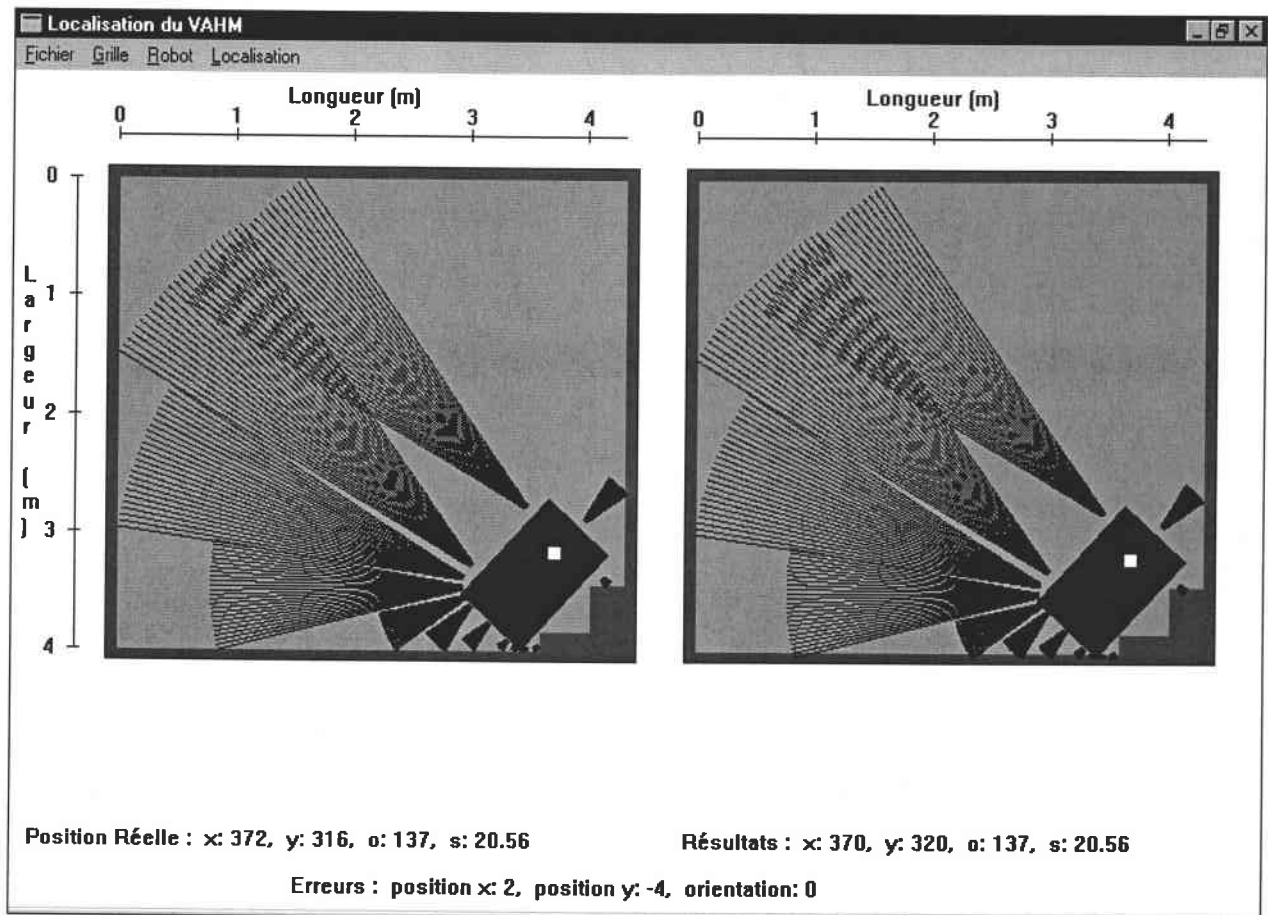


Figure III-7 : Résultat de la localisation en corrigeant la surface du robot.

Nous avons repris les mêmes configurations que pour les essais précédents en appliquant la correction de la surface du robot. Ceci induit une amélioration des résultats en ce qui concerne la simulation puisqu'elle ne présente plus que deux cas ne satisfaisant pas les conditions de réussite sur les 34 essais réalisés (Annexe 2). Le réajustement de la valeur des cellules du robot permet d'éviter certains cas d'échecs mais il est aussi nécessaire de préciser le modèle défini pour les mesures afin d'éliminer certaines incohérences apparaissant dans la grille locale. En effet, l'analyse de différentes grilles a montré certains problèmes notamment : la présence de cellules libres derrière des cellules occupées provenant de la superposition des cônes, la définition d'arcs ne présentant pas une épaisseur constante en raison de la discrétisation des mesures (Figure III-8).

L'objet de la partie suivante concernera le traitement de ce type de configurations.

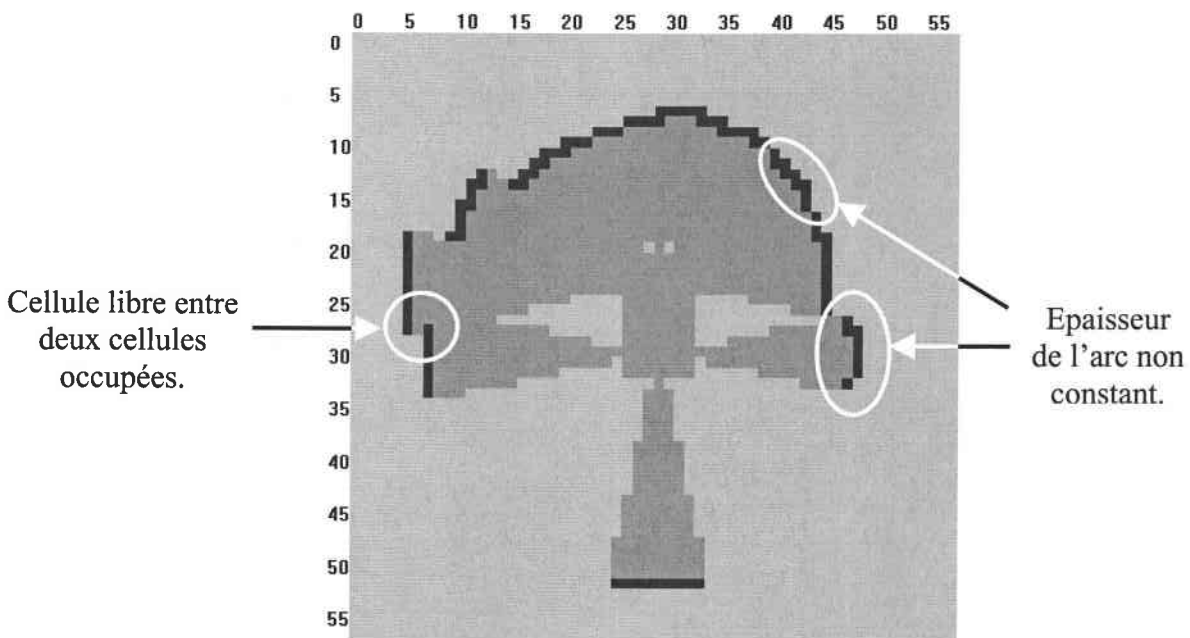


Figure III-8 : Grille locale présentant des incohérences.

III.3 RECHERCHE DE COHERENCE POUR LA GRILLE LOCALE

[COU98b]

III.3.1 Superposition des cônes ultrasonores

III.3.1.1 Incohérences dans la grille locale

Les valeurs réelles données aux cellules de la grille locale s'appliquent individuellement sur chaque mesure des capteurs à ultrasons. Or notre robot mobile est équipé d'une ceinture de 14 capteurs à ultrasons ; il arrive donc que certains d'entre eux aient leurs cônes d'émission qui se superposent. Ainsi, quand deux capteurs voisins renvoient des mesures suffisamment grandes pour se superposer, il apparaît dans la carte locale des espaces libres derrière des obstacles. Ce cas ne peut être considéré comme une représentation acceptable de l'état des cellules puisqu'il est impossible d'avoir une connaissance sur l'état de l'environnement derrière un obstacle s'il est perçu par des capteurs fixes dans une position fixe. Par ailleurs, il faut définir la valeur à attribuer à une

cellule de la carte locale quand elle est recouverte par les cônes de deux capteurs différents. La figure suivante (Figure III-9) montre les différentes configurations de recouvrement de cônes.

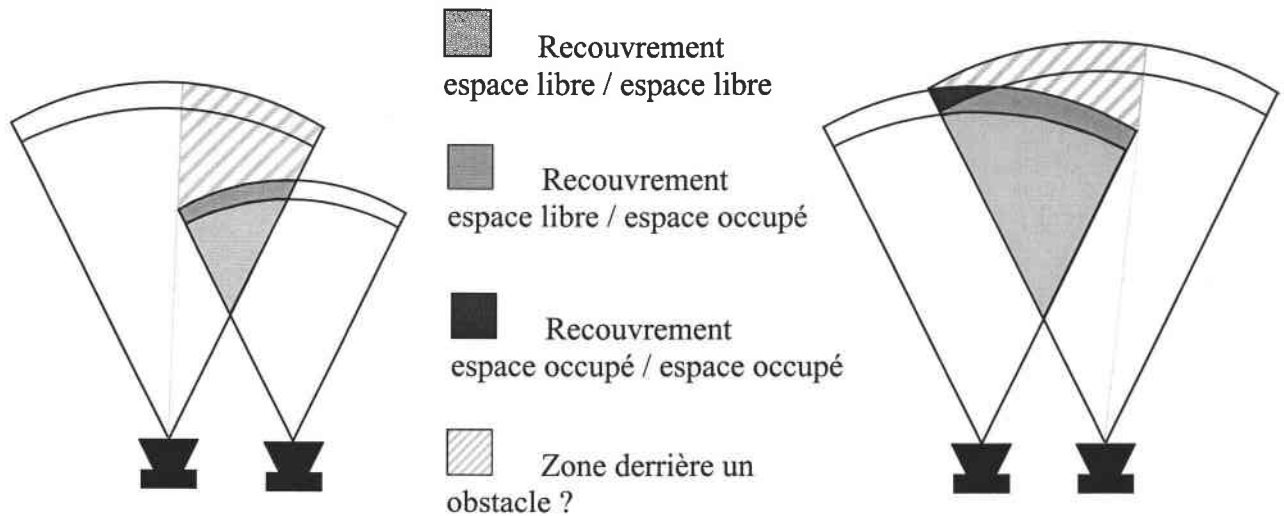


Figure III-9 : Configurations de recouvrement de cônes possibles.

III.3.1.2 Analogie avec les méthodes de mise à jour d'une grille d'occupation

Pour traiter du problème de la superposition des cônes d'émissions de deux capteurs à ultrasons, il nous semble opportun de faire l'analogie avec les méthodes de mise à jour d'une grille d'occupation. En effet, la mise à jour d'une grille dynamique, souvent utilisée pour l'exploration d'environnements inconnus, consiste à déterminer la nouvelle valeur à attribuer à la cellule connaissant la valeur actuelle de cette cellule et la nouvelle valeur provenant de la mesure. Si l'on se place au niveau de la cellule, le fait de lui donner une première valeur à un instant t puis une seconde à un instant $t+1$ à partir d'une mesure issue du même capteur situé à deux emplacements différents, est similaire au fait de lui attribuer une première valeur par un capteur et une deuxième par un autre capteur (Figure III-10).

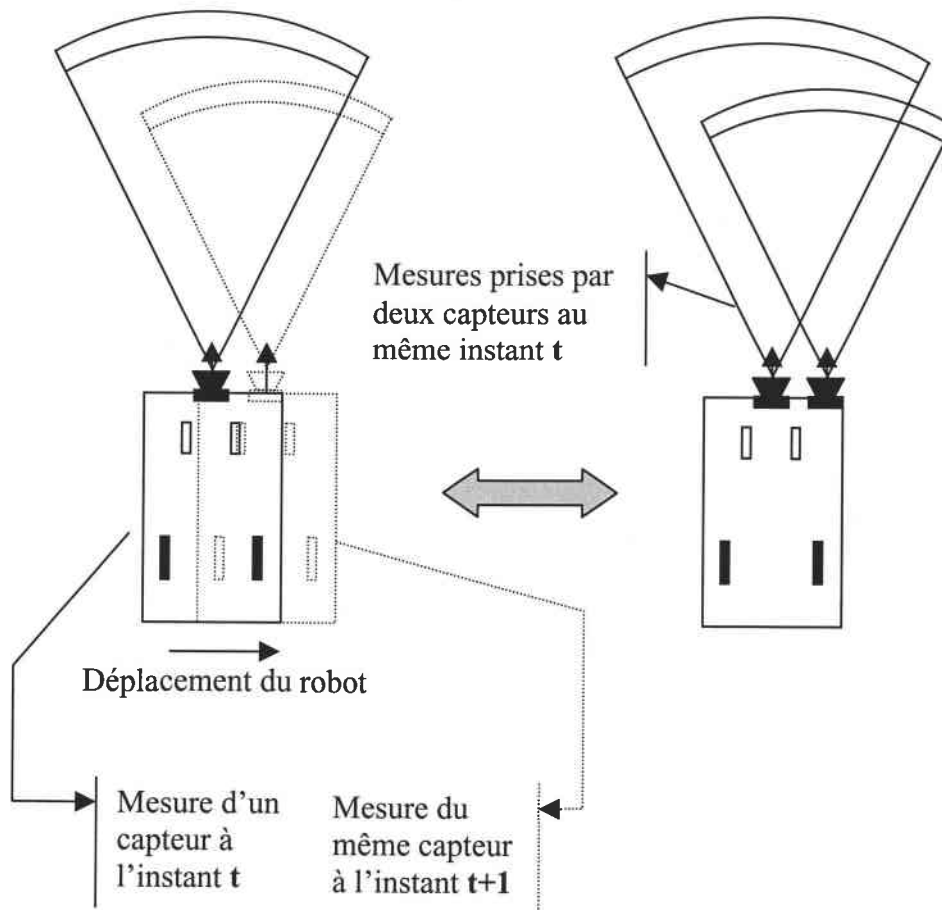


Figure III-10 : Analogie entre mesures successives d'un capteur et mesures de deux capteurs.

Parmi les méthodes traitant de la mise à jour de grilles d'occupations il y a celles utilisant des profils de probabilités. Ces méthodes ne suppriment pas un obstacle détecté par un capteur et situé dans l'espace libre d'un autre capteur. Elles réduisent l'influence de l'obstacle en diminuant la probabilité d'occupation de la cellule concernée. Les différents modèles existants reposent sur de nombreuses mesures et peuvent alors utiliser des lois probabilistes. Dans notre cas, la superposition de deux cônes ressemble par le principe à ces modèles mais n'est basée dans la plupart des cas que sur deux mesures. Nous devons alors interpréter les mesures qui se superposent pour obtenir un résultat qui soit cohérent avec le modèle des capteurs à ultrasons que nous utilisons et qui soit adapté pour la mise en correspondance des grilles lors de la localisation.

III.3.1.3 Principes de superposition pour notre méthode

Comme notre modèle n'utilise pas de probabilités mais des valeurs réelles positives et négatives, une méthode simple de discrétisation serait de faire la somme des valeurs des cellules

superposées. Or pour une mesure, les cellules représentant l'obstacle détecté sont moins nombreuses que celles représentant l'espace libre, elles ont alors une valeur plus forte (en valeur absolue). Dans ce cas, un obstacle détecté par un capteur et situé dans l'espace libre d'un autre capteur n'est pas supprimé puisque la somme est positive, la valeur de la cellule est simplement réduite. Le problème serait alors sur l'interprétation d'une telle représentation, que signifie une cellule un peu occupée ?

Il nous faut donc traiter séparément les deux cas de recouvrements, à savoir le cas où les deux cellules superposées sont contradictoires et le cas où elles sont semblables.

- Recouvrement espace libre / espace occupé

Ce cas de recouvrement apparaît quand un capteur retourne une grande distance à côté d'un autre capteur renvoyant une distance plus petite. La grande mesure définit un espace libre alors que la petite représente un espace occupé. Sans aucune autre information que les quatorze mesures des capteurs à ultrasons, nous ne pouvons pas traiter les phénomènes de spécularité. Nous partons alors du principe qu'un capteur voit nécessairement un obstacle s'il est situé dans son cône d'émission. Par conséquent, si un obstacle existait réellement dans le cône d'émission du capteur renvoyant la grande mesure, ce dernier l'aurait détecté (Figure III-11). La partie de l'arc d'espace occupé créé par la plus petite mesure peut alors être écrasée par les cellules libres correspondantes de l'autre capteur. Ainsi, le nombre de cellules occupées provenant de la petite mesure est réduit et la valeur à attribuer à ces cellules doit être ajustée. Si nous gardons les valeurs initiales, le poids de cette mesure lors de la localisation sera différent par rapport à une autre mesure ; ce qui est contraire à notre modèle qui donne la même contribution à chaque mesure. Par contre, en tenant compte du nombre de cellules restantes, leurs valeurs sont augmentées ; s'il reste n'_1 cellules, on leur donne la valeur $1/n'_1$. Cela représente le fait que l'obstacle est certainement dans cette petite zone. De cette façon lors de la localisation, les cellules correspondant à la petite mesure gardent un poids plus important. Ce choix est en adéquation avec notre modèle puisqu'il respecte les principes utilisés ; c'est à dire que chaque mesure a une contribution égale lors de la localisation, et une cellule représentant un obstacle d'une petite mesure a plus de poids qu'une autre provenant d'une grande mesure.

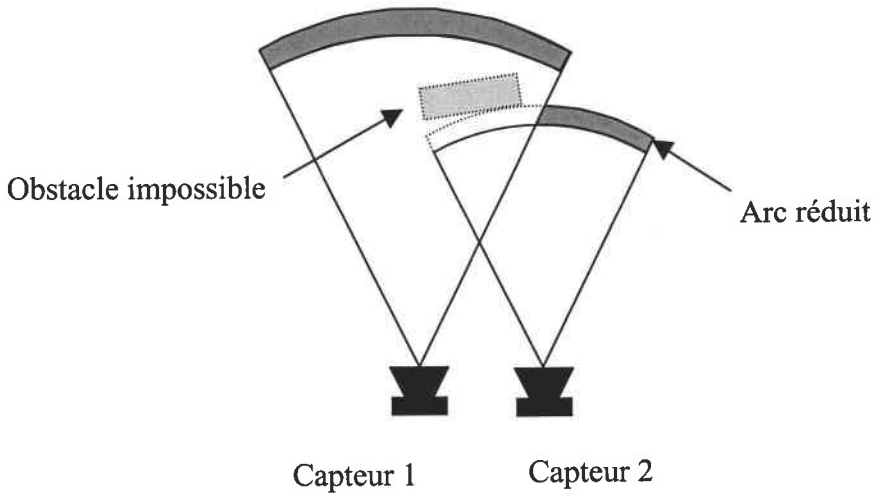


Figure III-11 : Recouvrement de cônes.

- Recouvrement espace libre / espace libre ou espace occupé / espace occupé

L'étude qui suit concerne la superposition de deux cellules semblables. Nous la détaillons pour le cas de deux cellules libres qui est le cas le plus fréquent puis par analogie nous l'appliquons à la superposition de deux espaces occupés. Quand deux cellules libres de deux capteurs différents se superposent, nous choisissons celle munie de la plus grande valeur pour favoriser la mesure la plus petite qui est la plus précise. Ainsi, le nombre de cellules définissant l'espace libre de la petite mesure reste le même alors que celui de la grande mesure est réduit. Dans le cas du recouvrement de deux cellules contradictoires nous ajustons la nouvelle valeur des cellules de l'arc réduit, mais ce principe ne peut pas être appliqué dans le cas présent. En effet, s'il reste n'_2 cellules libres pour la grande mesure et que nous leur donnons la valeur $1/n'_2$, le poids d'une cellule de la mesure la plus grande pourrait être alors plus important pour la localisation qu'une mesure plus petite, ce qui est en contradiction avec notre modèle. Nous ne pouvons pas non plus attribuer la somme des deux valeurs à la cellule puisque ceci donnerait plus de poids à la zone superposée. Dans le cas de deux cellules occupées superposées cela signifierait que nous considérons qu'un obstacle est à cet endroit alors que rien ne le confirme. L'exemple de la figure III-12 montre le cas où la zone superposée ne correspond justement pas à la localisation d'un éventuel obstacle.

En conclusion, lors de la superposition de deux cellules de même nature nous choisissons la valeur la plus importante en valeur absolue sans réajustement des cellules voisines.

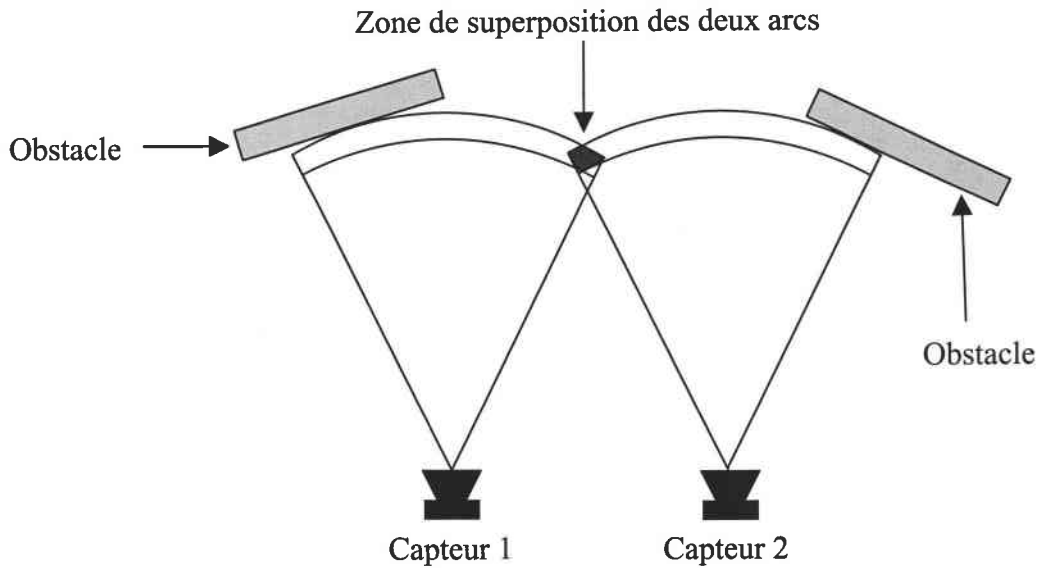
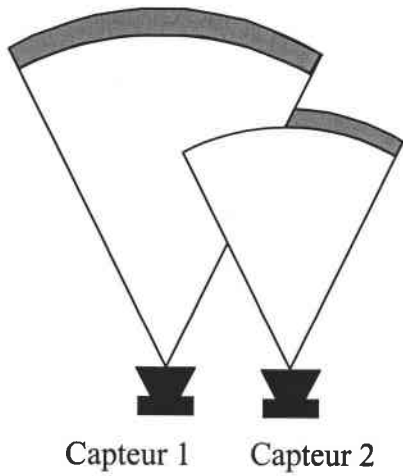
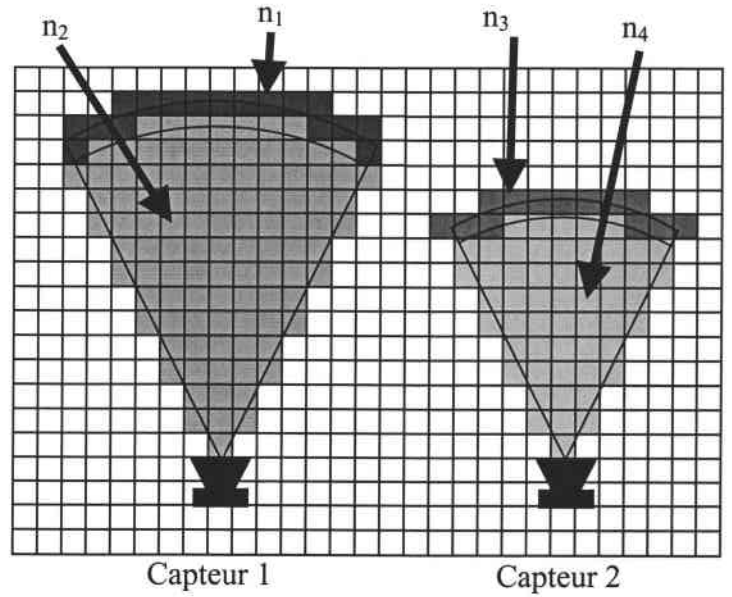
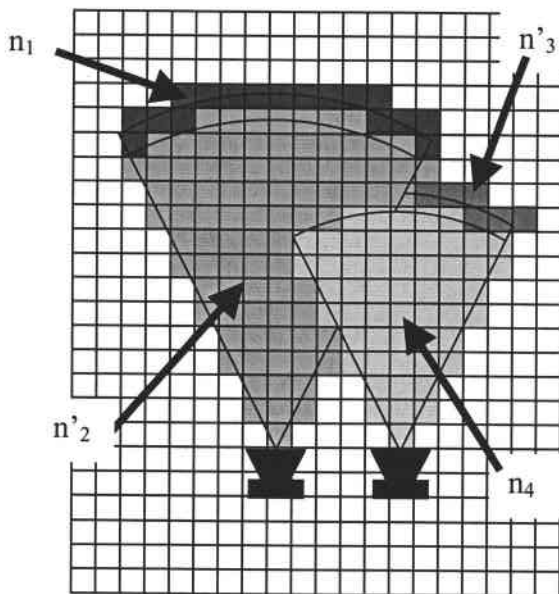


Figure III-12 : Exemple où la zone commune des arcs ne correspond pas à la localisation d'un obstacle.

- Exemple de superposition de cônes ultrasonores

Les choix que nous avons fait donnent la priorité à la mesure la plus grande pour la perception de l'obstacle et la priorité à la mesure la plus petite pour décrire l'espace libre. Cela nous a permis de conserver le principe de notre modélisation qui affecte des valeurs plus importantes aux cellules issues de mesures plus petites. Ces choix nous ont permis également de résoudre les problèmes de cohérence de la carte comme la présence de cellules libres ou occupées derrière un obstacle. La figure III-13 résume notre modèle de superposition en indiquant le modèle dans sa forme continue, les mesures discrétisées indépendamment et les mesures discrétisées en traitant le recouvrement de cônes.

Modèle continuMesures discrétisées indépendammentMesures discrétisées en traitant le recouvrement

- $n_1 (=17)$ cellules occupées du capteur 1 munies de la valeur $1/n_1$
- $n_2 (=102)$ cellules libres du capteur 1 contenant la valeur $1/n_2$
 - $n'_2 (=87)$ cellules libres du capteur 1 contenant la valeur $1/n_2$
- $n_3 (=13)$ cellules occupées du capteur 2 contenant la valeur $1/n_3$
 - $n'_3 (=6)$ cellules occupées du capteur 2 contenant la valeur $1/n'_3$
- $n_4 (=54)$ cellules libres du capteur 2 munies de la valeur $1/n_4$

Figure III-13 : Modélisation des recouvrements de cônes.

III.3.2 Discrétisation des arcs

III.3.2.1 Influence de l'orientation du capteur

Pour représenter dans la grille locale un obstacle détecté par un capteur à ultrasons nous sélectionnons toutes les cellules traversées par l'arc de la mesure. Ainsi, le nombre de cellules occupées le long d'un arc dépend de l'orientation de l'axe acoustique (Figure III-14). Deux capteurs retournant la même distance mais orientés différemment sélectionneront un nombre de cellules différent ; les valeurs attribuées aux cellules ne seront donc pas les mêmes pour les deux arcs. Lors de la mise en correspondance, une cellule d'un arc n'aura pas le même poids qu'une cellule de l'autre arc alors que les deux arcs représentent la même distance.

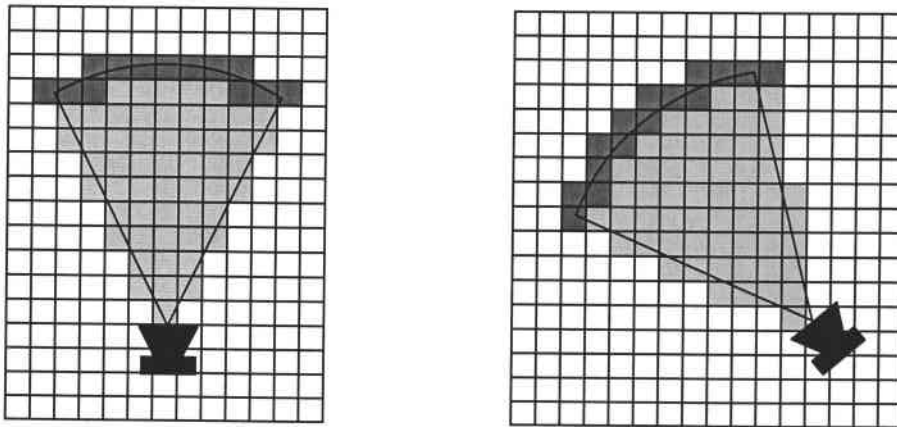


Figure III-14 : Nombre de cellules dans l'arc en fonction de l'orientation du capteur (à droite : 0 degré, arc de 13 cellules / à gauche : 40 degrés, arc de 15 cellules).

Le problème rencontré est le même que celui de la représentation d'une image continue par un ensemble de pixels discrets. Pour rechercher la représentation optimale de l'arc nous avons emprunté aux méthodes d'analyse d'image, des notions de géométrie discrète telles que le «voisinage», le «chemin connexe», la «composante connexe» et enfin «l'arc discret» [CHA91].

III.3.2.2 Notions de géométrie discrète

Le voisinage :

Chaque cellule de la grille possède 4 voisins directs qui sont les quatre cellules ayant une arête commune, et 4 voisins diagonaux (ou indirects) qui sont les quatre cellules ayant un sommet commun (Figure III-15).

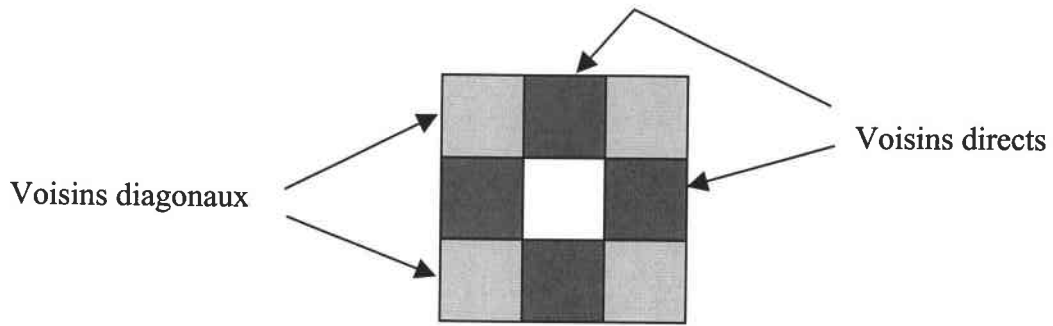


Figure III-15 : Voisins d'une cellule.

Le chemin connexe :

Pour relier deux cellules, il existe deux types de chemins, à savoir des chemins 4-connexes ou 8-connexes suivant que l'on utilise les 4 voisins (directs) ou les 8 voisins (4 directs et 4 diagonaux) (Figure III-16).



Figure III-16 : Chemins 4-connexe (a) et 8-connexe (b).

La composante connexe :

La composante connexe est un ensemble de cellules pouvant toutes être reliées par un chemin connexe. En fonction de la notion de chemin connexe utilisée, on définit des composantes 4-connexes (associées aux chemins 4-connexes) ou 8-connexes (associées aux chemins 8-connexes).

L'arc discret :

Soit S un ensemble de points (cellules), S est un arc si et seulement si S est connexe et si chacun de ses points (cellules), excepté deux d'entre eux, possède exactement deux voisins dans S . Les deux points (cellules) n'ayant pas deux voisins sont appelés les extrémités de l'arc S .

III.3.2.3 Application aux arcs des mesures ultrasonores

Ainsi, il existe des arcs 4-connexes comme ceux présentés à la figure III-14 qui ne nous conviennent pas puisqu'ils n'ont pas une épaisseur constante d'une cellule, et des arcs 8-connexes qui sont en adéquation avec notre modèle (Figure III-17). Bien sûr, le nombre de cellules est réduit et les valeurs de ces cellules sont alors mises à jour ; le nombre de cellules de l'arc ne dépend plus de l'orientation du capteur.

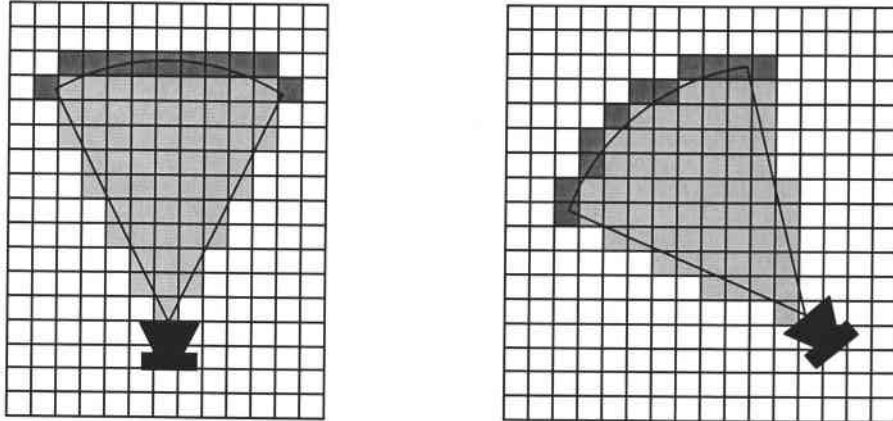


Figure III-17 : Correction des arcs de la figure III-14 en utilisant la 8-connexité.

III.3.3 Problèmes liés à la rotation des grilles

III.3.3.1 Non-invariance de la rotation discrète

Pour trouver la meilleure correspondance entre la grille locale et la grille globale nous devons essayer toutes les translations et rotations possibles d'une grille par rapport à l'autre dans l'aire de recherche. De ce fait, nous créons une deuxième grille locale correspondant à la grille locale initiale sur laquelle nous appliquons la rotation, puis nous effectuons la translation sur cette nouvelle grille pour réaliser la superposition avec la grille globale.

Pour l'application de transformations géométriques telles que la translation et la rotation, lorsque les arguments de ces transformations correspondent à des valeurs entières pour la translation ou à des angles multiples de 90 degrés pour la rotation, on conserve les caractéristiques de la grille. Les problèmes non résolus concernent les cas généraux avec des arguments quelconques. Dans notre application, l'algorithme de localisation effectue des translations d'arguments entiers, nous déplaçons la grille par pas de 1 cellule ; par contre les arguments de la

rotation sont quelconques, le pas de recherche étant de 5° autour de l'orientation estimée, ce qui ne permet pas de préserver la grille initiale et induit des erreurs dans le modèle.

En effet, la rotation directement appliquée à une grille associe deux fois certaines cellules de la grille initiale alors que d'autres ne sont pas associées. Après rotation, les critères de connexité que nous avons définis pour un arc ne seront plus vérifiés. Or le résultat de l'algorithme de localisation dépend de la somme des valeurs des cellules. Donc si le nombre de cellules diffère selon l'orientation de la grille, la comparaison entre les différentes grilles tournées sera faussée.

La figure III-18 illustre le cas où deux cellules de la grille tournée obtiennent la valeur issue de la même cellule de la grille initiale et le cas où une cellule de la grille initiale ne recouvre aucun point central des cellules de la grille tournée.

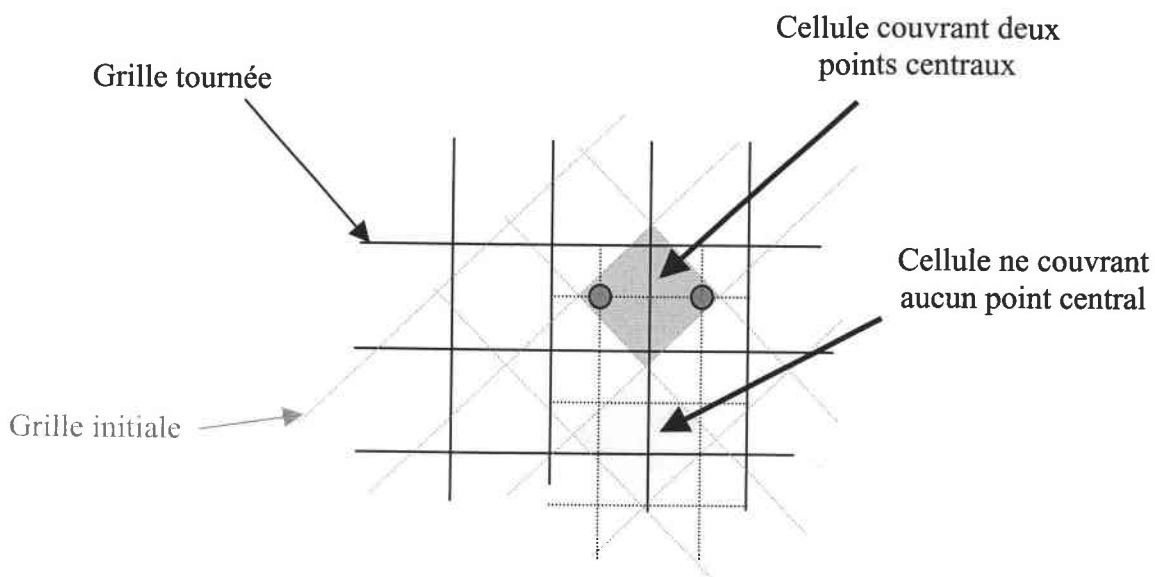


Figure III-18 : Non-invariance de la rotation d'une grille.

III.3.3.2 Application d'une rotation analogique

Pour résoudre ce problème nous créons à partir des mesures un modèle analogique sur lequel nous effectuons la transformation, ensuite nous le discrétisons et obtenons la nouvelle forme discrète. Il nous faut donc créer la grille directement suivant l'orientation désirée. Il est alors nécessaire de créer autant de grilles qu'il y a d'orientations à considérer.

Le modèle analogique que nous utilisons est défini par quinze régions graphiques indépendantes décrivant le robot et les quatorze cônes d'émission de chaque capteur à ultrasons. Ce modèle n'est utilisé que pour définir les valeurs à affecter aux cellules de l'espace libre car les outils graphiques dont nous disposons ne nous permettent pas de définir un arc. Les cellules

correspondant aux obstacles détectés sont alors sélectionnées comme nous l'avons décrit au chapitre II.3.2 puis nous leur appliquons les règles de connexité vues précédemment.

Pour discrétiser ce modèle nous testons pour chaque cellule de la grille, son appartenance à une région. Quand une cellule appartient à deux régions il faut alors choisir celle qui suit les règles de recouvrements de cônes décrites plus haut. Or pour ces règles il faut connaître la surface de chaque cône et tous les cônes auxquels la cellule appartient. Afin de réaliser cette opération de discrétisation rapidement nous avons créé un codage particulier qui permet d'avoir pour chaque cellule l'identification de tous les capteurs qui la concernent. Quand une cellule appartient à une région elle est étiquetée par le numéro du capteur concerné. Si cette même cellule appartient à une deuxième région elle est étiquetée par une nouvelle valeur définie par :

nouvelle valeur = $100 * \text{numéro du premier capteur} + \text{numéro du deuxième capteur}$.

Par exemple si le capteur 5 et 6 ont leurs cônes d'émissions qui se superposent, la valeur $100 * \underline{5} + \underline{6} = 506$ sera affectée aux cellules concernées.

Cette règle est encore appliquée quand cette même cellule est située dans une troisième région ; ce cas arrive notamment pour les trois capteurs situés en avant du robot, ce qui nous donne la valeur : $100 * (100 * \underline{1} + \underline{2}) + \underline{14} = 10214$.

Connaissant les capteurs agissant sur une cellule, nous pouvons attribuer à cette dernière la valeur calculée à partir du capteur dominant défini par les règles de superpositions.

Par cette méthode nous obtenons la grille orientée suivant l'orientation de calcul désirée. L'inconvénient est qu'il est nécessaire de construire autant de modèles et de grilles qu'il y a d'orientations différentes à considérer. Néanmoins pour une grille locale donnée, la somme de toutes les cellules est approximativement constante quelle que soit l'orientation de la grille. L'exemple de la figure III-19 compare un modèle discrétisé puis tourné de 69° par rotation discrète, à un modèle discrétisé après avoir orienté le modèle analogique dans la même orientation. On remarque que dans le cas de la rotation discrète certaines cellules occupées apparaissent derrière des obstacles et d'autres cellules qui devraient être occupées deviennent libres (cellules encerclées), alors que dans la grille orientée ces incohérences n'existent pas.

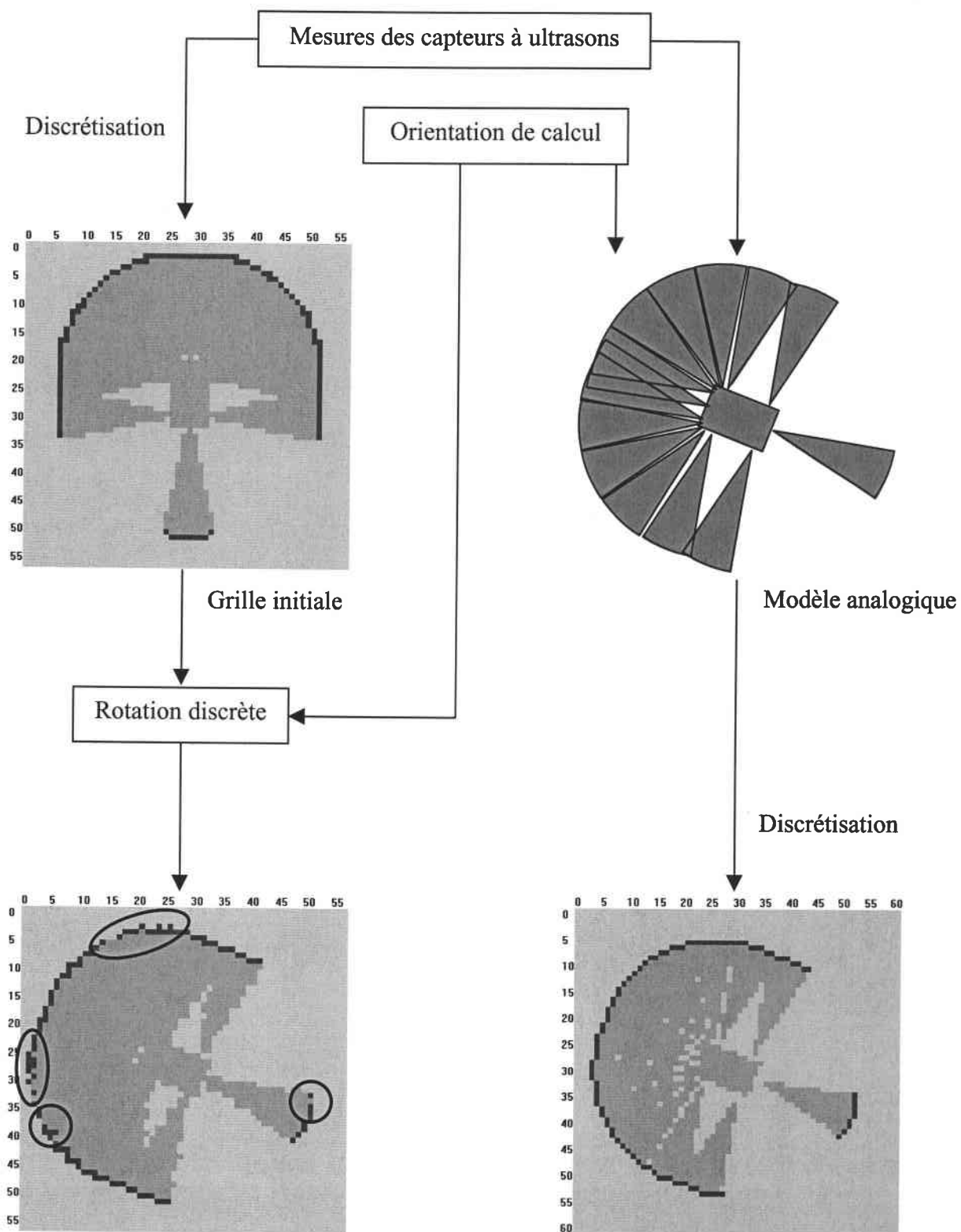


Figure III-19 : Comparaison de la rotation discrète à la rotation analogique.

III.3.4 Essais

Nous avons mis en œuvre les modifications décrites dans les paragraphes précédents : la discrétisation tenant compte de la superposition des cônes, la correction des arcs de mesures à l'aide d'un modèle 8-connexe, et le remplacement de la rotation discrète par une rotation analogique ; ce qui a permis de réduire les erreurs moyennes pour les mêmes configurations du robot (Annexe 2). Pour nous permettre d'évaluer plus précisément notre algorithme, nous avons alors décidé de réaliser des essais dans trois environnements plus complexes que celui utilisé jusqu'à présent. Ainsi, nous avons effectué 33 essais dans une première pièce carrée de 6 mètres de côté et peu encombrée, puis nous avons exécuté 47 essais dans la pièce réduite à environ 5 mètres de côté mais munie de plus d'objets, et enfin 20 derniers essais dans cette même pièce avec un obstacle supplémentaire en son centre (Figure III-20).

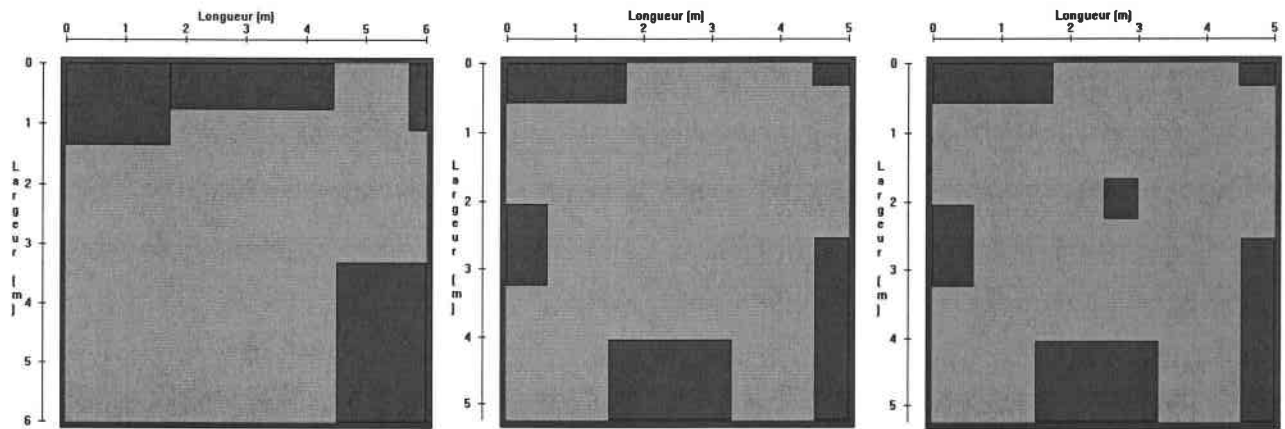


Figure III-20 : Les trois environnements utilisés pour évaluer l'importance de l'environnement dans la réussite de la localisation.

Sur les 100 essais réalisés, 75 cas en simulation et 54 en conditions réelles sur le robot respectent les conditions d'une bonne localisation (Annexe 3). Ce faible taux de réussite en simulation nous a amenés à étudier de plus près l'impact des environnements sur la localisation. Pour ce faire, nous avons repris l'environnement présenté à la figure III-4 auquel nous avons ajouté un mur fictif d'une épaisseur de 50 cm à gauche de la pièce afin de modifier la grille globale sans changer l'environnement du robot. Pour une position et orientation donnée du robot, les mesures des capteurs et donc la grille locale sont alors exactement les mêmes pour les deux environnements. Les résultats en simulation présentés à la figure III-21 illustrent l'importance de l'environnement pour la localisation puisque dans le premier cas l'erreur est nulle alors que dans le second il apparaît

une erreur en position de 30 cm. Ainsi, nous avons pu mettre en évidence l'importance du choix des cellules de la grille globale.

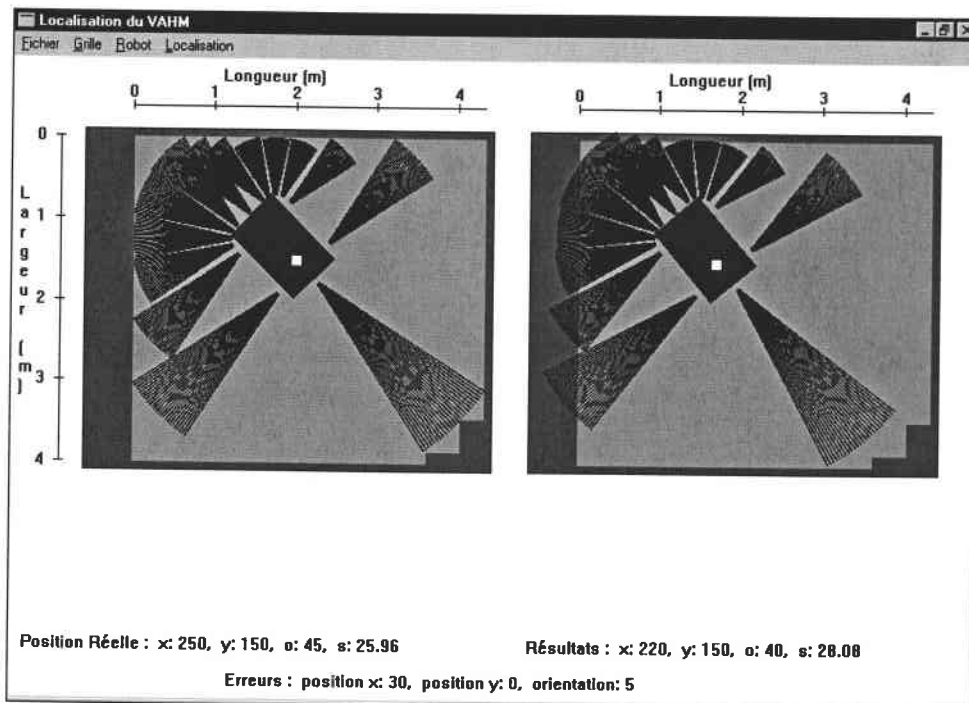
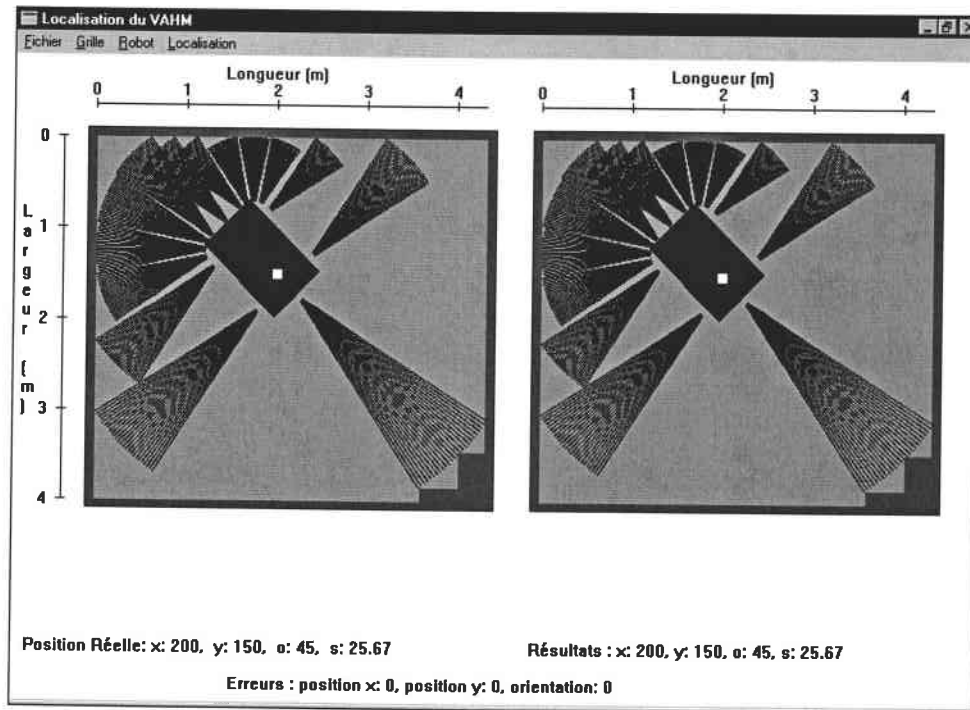


Figure III-21 : Influence de l'environnement sur le résultat de la localisation.

III.4 AFFINEMENT DE LA GRILLE GLOBALE ET ESSAIS

[COU99]

Dans les essais précédents nous avons constaté que l'environnement, et donc la grille globale, agit sur la localisation. L'analyse des résultats révèle que contrairement à la grille locale, la grille globale n'est pas appropriée pour la localisation. En fait dans certains cas comme le montre la figure suivante (Figure III-22), certains cônes apparaissent avoir une bonne superposition à l'intérieur des objets alors qu'il est impossible d'avoir une information sur ce qu'il y a derrière un obstacle, ce qui engendre dans l'exemple présenté une erreur de localisation de plus de 40 cm.

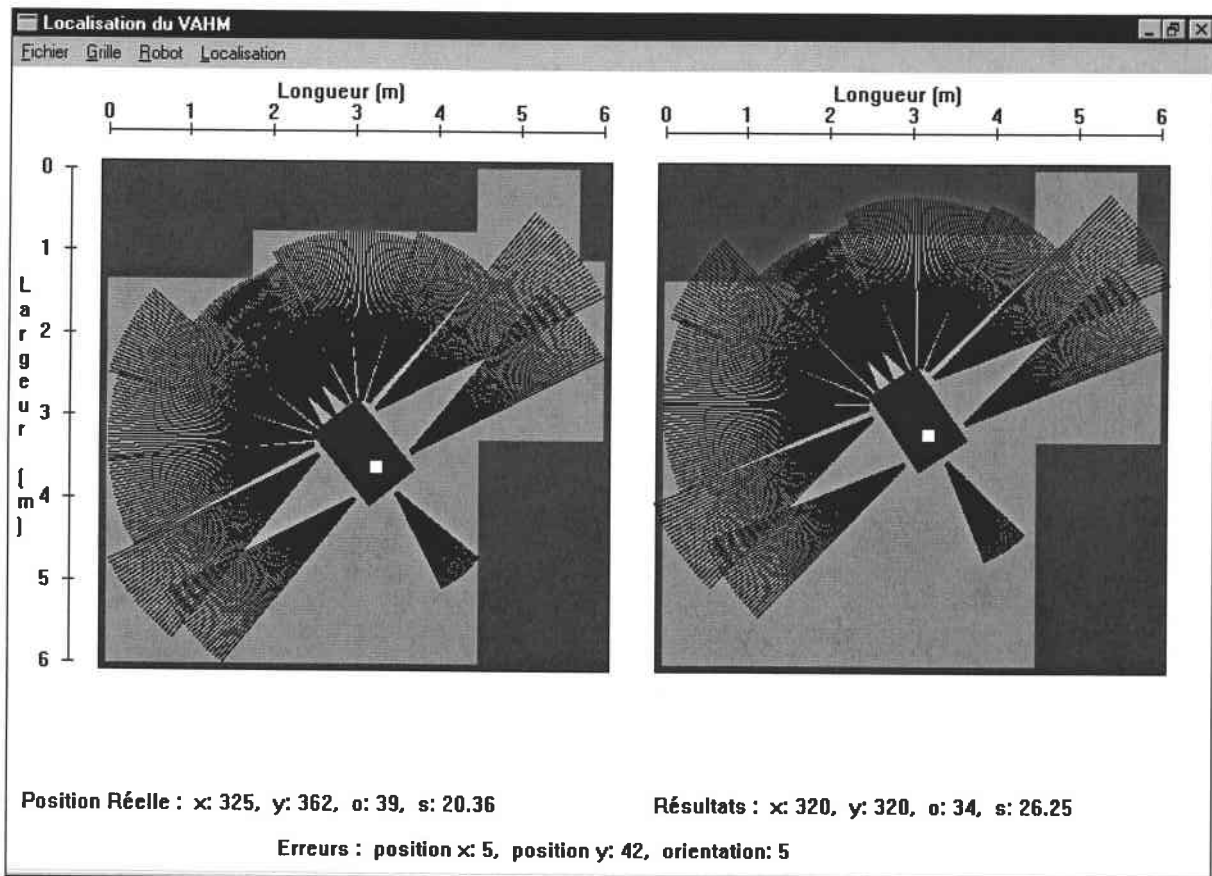


Figure III-22 : Exemple de localisation faussée par la grille globale.

La grille globale est construite à partir du plan de l'environnement dans lequel les objets sont codés. La grille est constituée de cellules libres contenant la valeur $C_G = -1$ et de cellules occupées contenant la valeur $C_G = 1$. Par ailleurs, toutes les cellules correspondant à un objet sont

utilisées. Or, dans le cas de la grille locale, l'arc d'un capteur à ultrasons utilise un arc dont l'épaisseur n'est que d'une cellule du fait du manque de connaissance sur ce qu'il y a derrière l'obstacle. En plaçant cet arc à l'intérieur d'un obstacle, il y a plus de cellules occupées qui coïncident que s'il était tangent à l'obstacle. L'exemple de la figure III-23 illustre ce cas en analysant l'exemple de la figure III-22. Nous n'avons représenté qu'une seule mesure pour laquelle les cellules en bonne correspondance sont de couleur claire et les cellules en contradictions sont de couleur noire. En plaçant le robot sur sa position réelle, toutes les cellules de l'arc ne se superposent pas sur l'obstacle ce qui donne une somme de -0,11, alors que le secteur fournit une valeur égale à 0,99, ce qui donne une somme totale de 0,88. Sur la position calculée par contre, les cellules représentant l'obstacle contribuent à la somme avec un poids égal à 1 (puisque toutes les cellules de l'arc sont bien placées) alors que les cellules en contradiction du secteur ne retranchent qu'une valeur égale à 0,27, ce qui globalement donne une somme égale à 1,46 supérieure à la somme pour la position réelle. En appliquant le même raisonnement aux autres mesures cela explique pourquoi nous avons une différence sur la somme S dans l'exemple de la figure III-22.

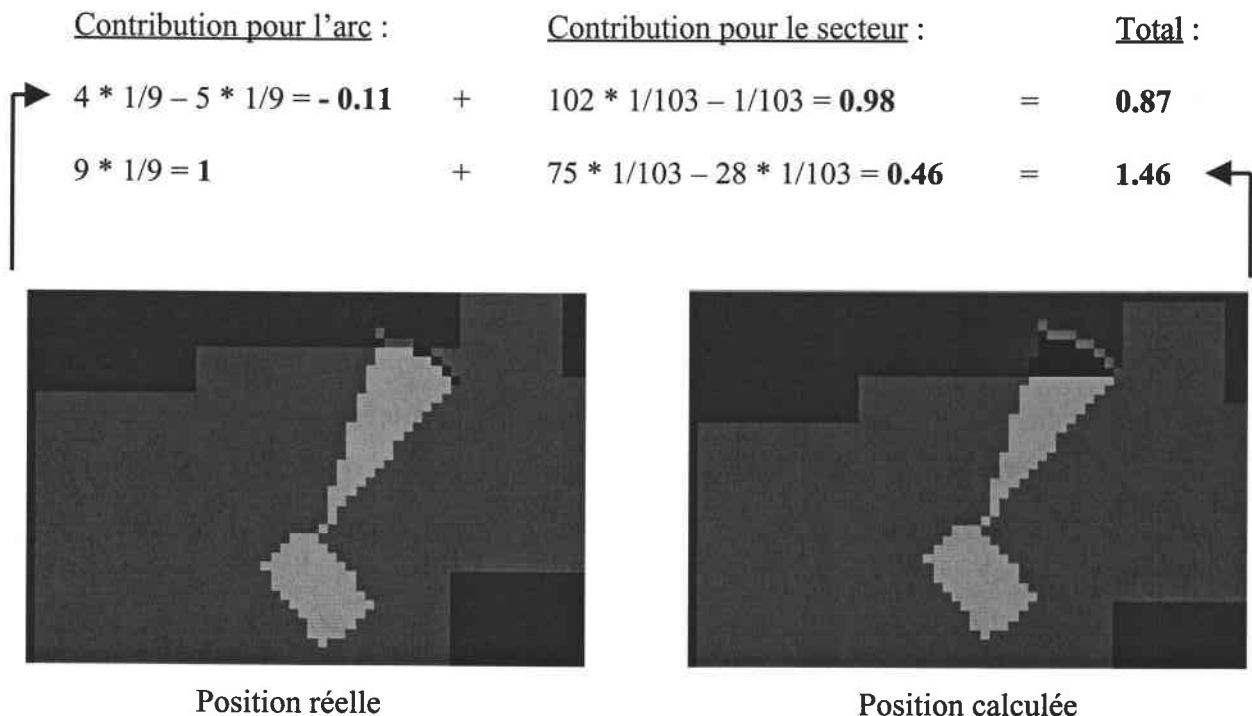


Figure III-23 : Analyse de la superposition pour un cône.

Ainsi, même si la grille globale correspond à la réalité, elle induit des erreurs car la grille locale représente une limite de l'espace libre. Donc, si nous voulons superposer les cellules occupées de la grille locale sur un objet de la grille globale, il est nécessaire d'appliquer le même principe sur les deux grilles.

Pour éviter d'avoir une information derrière un obstacle, nous modifions la grille globale pour obtenir le contour des obstacles avec une épaisseur d'une seule cellule. La grille de départ est créée comme précédemment (voir chapitre II.3.3), puis pour chaque cellule occupée $C_G(i,j)$ un test de voisinage vérifie la non-occupation des huit cellules voisines $C_G(k,l)$ (avec $k=i-1, i, i+1$ et $l=j-1, j, j+1$). Si l'une d'entre elles est libre alors la cellule occupée $C_G(i,j)$ est un contour et ne doit donc pas être modifiée. Par contre, si aucune des cellules voisines n'est libre, alors la cellule occupée $C_G(i,j)$ représente l'intérieur d'un objet et la valeur inconnue $C_G = 0$ doit donc lui être attribuée. La figure III-24 illustre la correction de la grille globale.

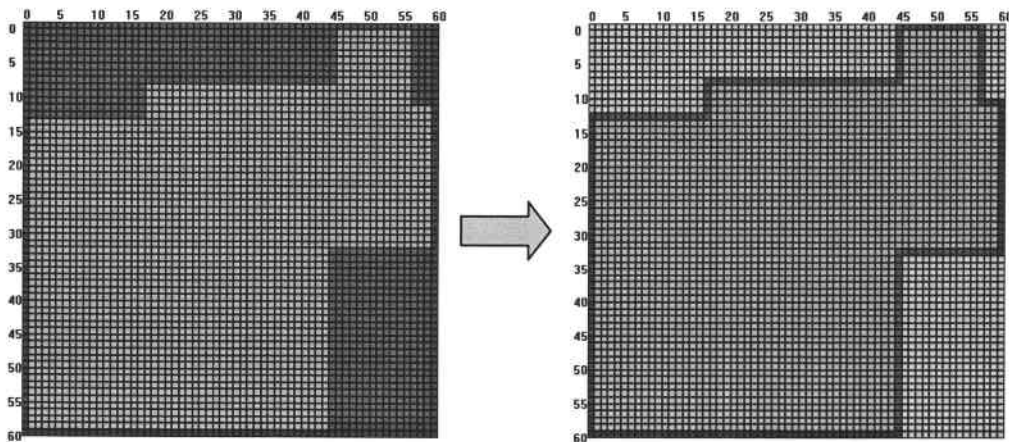


Figure III-24 : Obtention des contours des objets.

En reprenant l'exemple de la figure IV-23, la position réelle donne une somme totale de 0,77 alors que la position calculée a une somme de 0,84. La différence entre les deux positions est atténuée mais ne suffit pas. En effet, pour la position réelle, la partie de l'arc de la mesure située dans l'espace libre de la pièce est en contradiction, ce qui décrémente la somme. Pour la position calculée, la partie de l'arc placée sur les cellules intérieures de l'objet ne modifie pas la somme, il faudrait que ces cellules soient aussi en contradiction. Pour corriger cela, nous retranchons alors systématiquement la valeur de la cellule de la grille locale qui se superpose à une cellule de la grille globale contenant la valeur nulle. L'apport de l'ensemble des corrections de la grille globale sur l'exemple présenté à la figure III-22 est montré à la figure III-25.

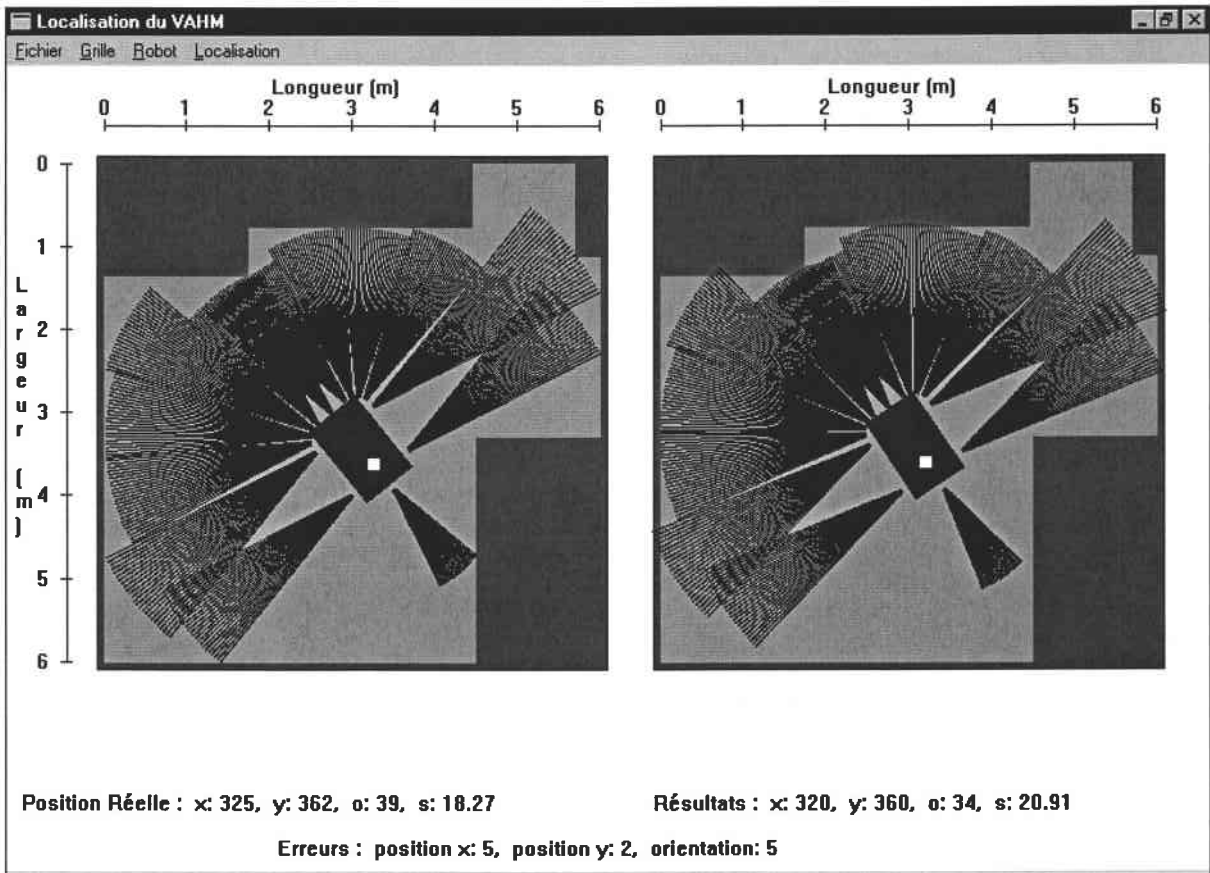


Figure III-25 : Exemple de localisation corrigée grâce à la correction de la grille globale.

Pour montrer la contribution de la correction de la grille globale, nous avons alors réalisé les mêmes essais pour les mêmes positions. Les résultats sont nettement améliorés en simulation en corrigeant 25 configurations sur les 26 présentant une trop grande erreur en position ou en orientation, le seul cas de mauvaise localisation restant révèle une erreur d'orientation de 10 degrés (Annexe 3). Les erreurs moyennes ont également été réduites pour les essais réalisés sur le robot, cependant il reste des configurations pour lesquelles le résultat est trop éloigné de la position réelle. La seule différence entre la simulation et les conditions réelles sont les données ultrasonores. Les cas réels pour lesquels la localisation est mauvaise présentent tous des mesures aberrantes. Nous avons alors décidé d'évaluer l'algorithme final dans un environnement se rapprochant de la réalité. Nous avons donc simulé un appartement constitué de plusieurs pièces et nous avons placé le robot dans une centaine de configurations différentes (Figure III-26). De plus, dans cette série nous avons fourni la position et l'orientation estimées du robot par une évaluation grossière (à vue) de la place du robot dans la pièce. Nous sommes ici dans les conditions réelles d'utilisation.

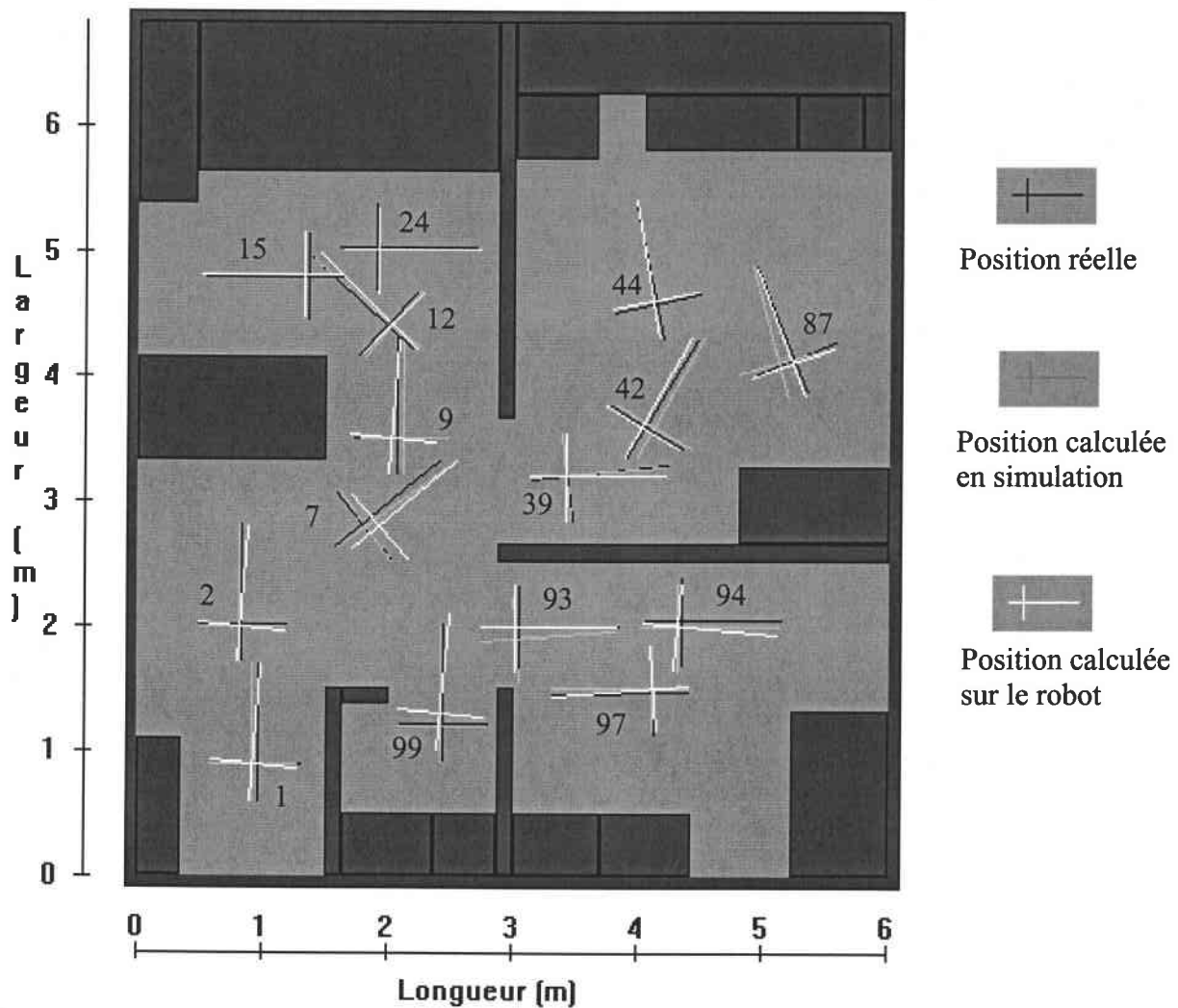


Figure III-26 : Essais réalisés dans un appartement simulé.

Ainsi, sur les 100 essais réalisés sur le robot, 80 sont acceptables alors que tous les essais simulés sont réussis (Annexe 3). Généralement la localisation est demandée après avoir franchi une porte pour démarrer un déplacement autonome dans la pièce et dans ce cas, le positionnement est réussi, comme pour les essais n°1, n°39 ou n°93. Dans le cas où la personne démarre l'application et se place dans une configuration telle qu'un objet ou un mur soit proche et droit (dans le sens où la droite portée par l'obstacle et la droite suivant l'orientation du fauteuil sont parallèles ou perpendiculaires), la localisation est assurée ; c'est le cas par exemple des essais n°2, n°15, n°24 ou n°97. Les configurations pouvant poser des difficultés sont celles pour lesquelles le fauteuil est placé au centre d'une pièce ou orienté d'un angle proche de 45 degrés. En effet, ces cas sont plus facilement sujet aux mesures trop longues dues aux réflexions multiples et à celles trop courtes dues aux phénomènes de diaphonie.

III.5 CONCLUSION

Les corrections apportées sur les différentes grilles ont permis d'améliorer non seulement les essais en simulation mais aussi les essais réalisés sur notre prototype. Pour ce faire, nous avons utilisé des valeurs réelles dans la grille locale permettant de donner la même contribution pour la localisation à chaque mesure des capteurs à ultrasons et de donner un poids raisonnable à la surface recouverte par le robot par rapport au poids des mesures. Par ailleurs la discrétisation employée qui corrige l'arc d'une donnée ultrasonore avec un modèle 8-connexe et qui corrige les cônes superposés à l'aide d'un algorithme approprié a permis d'obtenir une grille locale plus cohérente. De plus, la rotation discrète de la grille locale a été remplacée par une rotation du modèle analogique ce qui évite d'avoir des grilles différentes suivant l'orientation du robot. Enfin, les modifications réalisées sur la grille globale la rendent compatible avec le principe utilisé pour la grille locale, à savoir qu'il est impossible d'être informé de ce qu'il y a derrière un obstacle détecté par une seule mesure.

Toutes ces modifications nous ont amené à réaliser des essais en conditions réelles fournissant de bons résultats. Les configurations du robot pour lesquelles la localisation a échoué présentent tous des problèmes de réflexions multiples et des phénomènes de diaphonie. Pour cette dernière, l'application d'un tir programmé par la méthode EERUF permettrait de diminuer ce taux d'échec. En ce qui concerne les réflexions multiples, un filtrage des mesures permettrait d'y remédier en supprimant les mesures dépassant une distance seuil. Une étude sur les mesures des capteurs à ultrasons est alors nécessaire pour fixer ce seuil, ce qui a été réalisé et détaillé dans le chapitre suivant.

RECHERCHE D'UNE
OPTIMISATION POUR LA
MISE EN ŒUVRE DE LA
METHODE

IV RECHERCHE D'UNE OPTIMISATION POUR LA MISE EN ŒUVRE DE LA METHODE

IV.1 INTRODUCTION

Pour réaliser la localisation du robot mobile, nous avons développé une méthode de mise en correspondance de grilles. Une première grille appelée grille globale représente l'environnement d'évolution du robot, la seconde appelée grille locale enregistre les informations de distances des capteurs à ultrasons. Pour évaluer une configuration possible du robot, nous superposons les deux grilles et nous calculons le critère de mise en correspondance par la somme des produits des valeurs des cellules associées. Le produit étant réalisé pour toutes les cellules de la grille locale, si celle-ci a une dimension de $n \times n$ cellules, il faut alors effectuer n^2 produits. La recherche de la position de la grille locale dans la grille globale est limitée à une zone définie à partir d'une détermination approximative de la position du robot (voir chapitre II.4.3). Si cette zone est de dimension $p \times p$, le nombre de positions possibles de la grille locale est p^2 . Si le nombre d'orientations possibles est o , le nombre de calculs requis est donc en $O(o \cdot p^2 \cdot n^2)$. En prenant $o = 19$, $p = 17$ et $n = 70$ on a près de 3.10^7 produits à traiter, ce qui peut prendre quelques minutes sur le PC 486 embarqué sur le fauteuil. Ces valeurs correspondent à des cas réels :

- $o = 19$: correspond à la recherche de l'orientation du robot parmi celles possibles dans la zone de recherche. Cette zone est centrée sur l'orientation estimée et le calcul est réalisé par pas de 5° jusqu'à une limite de 45° , ce qui fait 9 calculs de part et d'autre de l'orientation estimée plus elle-même.
- $p = 17$: indique la recherche de la position du robot parmi toutes celles possibles dans la zone de recherche. Cette zone étant de ± 75 cm autour de la position estimée, en ayant une résolution de la grille globale de 10×10 cm par cellule, cela fait 8 cellules à tester de chaque côté de la cellule incluant la position estimée plus cette dernière.
- $n = 70$: représente une taille moyenne de la grille locale. Cette taille est obtenue lorsque la distance maximale détectée par les capteurs à ultrasons est d'environ 3 mètres et en utilisant une résolution de la grille locale de 10×10 cm par cellule.

Afin de replacer la charge de calcul dans une limite raisonnable nous mettons en œuvre une méthode de recherche hiérarchisée en réalisant une réduction des grilles que nous détaillerons dans la première partie de ce chapitre. Dans une seconde partie, nous parlerons du filtrage des mesures que nous avons réalisé et qui nous permet de limiter les mesures aberrantes dues aux réflexions multiples et aux phénomènes de diaphonie. Enfin nous finirons par des perspectives sur l'optimisation des grilles locale et globale en présentant les axes de recherche des travaux futurs.

Les valeurs des temps de calculs présentés dans ce chapitre sont données à titre indicatif. Le but n'est pas d'avoir un traitement en temps réel comme les applications de localisation dynamique ; dans notre cas il faut que la méthode fournisse le résultat en quelques secondes.

IV.2 REDUCTION DE GRILLES

[HOR97]

IV.2.1 Principe

Les temps de calculs dépendent de la résolution des grilles utilisées. Dans l'exemple cité en introduction elle est de 10x10 cm par cellules. En prenant 20x20 cm par cellules, le nombre de calculs est réduit puisque p devient égal à 9 au lieu de 17 et n passe de 70 à 35 cellules. Cependant avec des cellules d'une telle taille la précision du résultat n'est pas suffisante. C'est pourquoi, nous avons réalisé une méthode de recherche hiérarchisée utilisant des grilles définies pour différentes résolutions. Nous appelons niveau élevé un niveau utilisant une résolution faible, les grilles sont alors définies par des cellules de tailles plus grandes. Le principe de la méthode est alors de réaliser la recherche de la position et de l'orientation du robot à un niveau élevé, puis d'utiliser le résultat pour affiner la recherche à un niveau plus faible (Figure IV-1).

Pour ce faire, il faut construire les grilles locale et globale à tous les niveaux de résolution utilisés. Les deux grilles (locale et globale) du niveau le plus haut (niveau N) sont mises en correspondance en recherchant toutes les superpositions possibles dans la zone de recherche suivant la méthode décrite au chapitre II.4. Les grilles du niveau inférieur sont ensuite traitées en restreignant la recherche pour la mise en correspondance autour de la meilleure position trouvée au niveau supérieur. Ainsi, le résultat obtenu à un niveau sert de base pour traiter le niveau inférieur. La position délivrée au niveau le plus faible (niveau 1) représente la position et l'orientation du robot.

Le nombre de niveau de compression est un paramètre adaptable en fonction des conditions dans lesquelles le positionnement sera réalisé, par exemple une grande pièce peu encombrée peut utiliser plus de réduction qu'une petite pièce encombrée. Le nombre de fusion doit être tel que la grille de niveau le plus élevé soit encore représentative de la topologie de la pièce. La complexité de la méthode dépend du nombre de niveaux, de la résolution choisie à chaque niveau et de la définition des zones de recherche ; elle peut être exprimée par $O(\sum_{i=1}^N o_i * p_i^2 * n_i^2)$ mais nous ne pouvons pas donner son ordre de grandeur avant d'avoir fixé les paramètres.

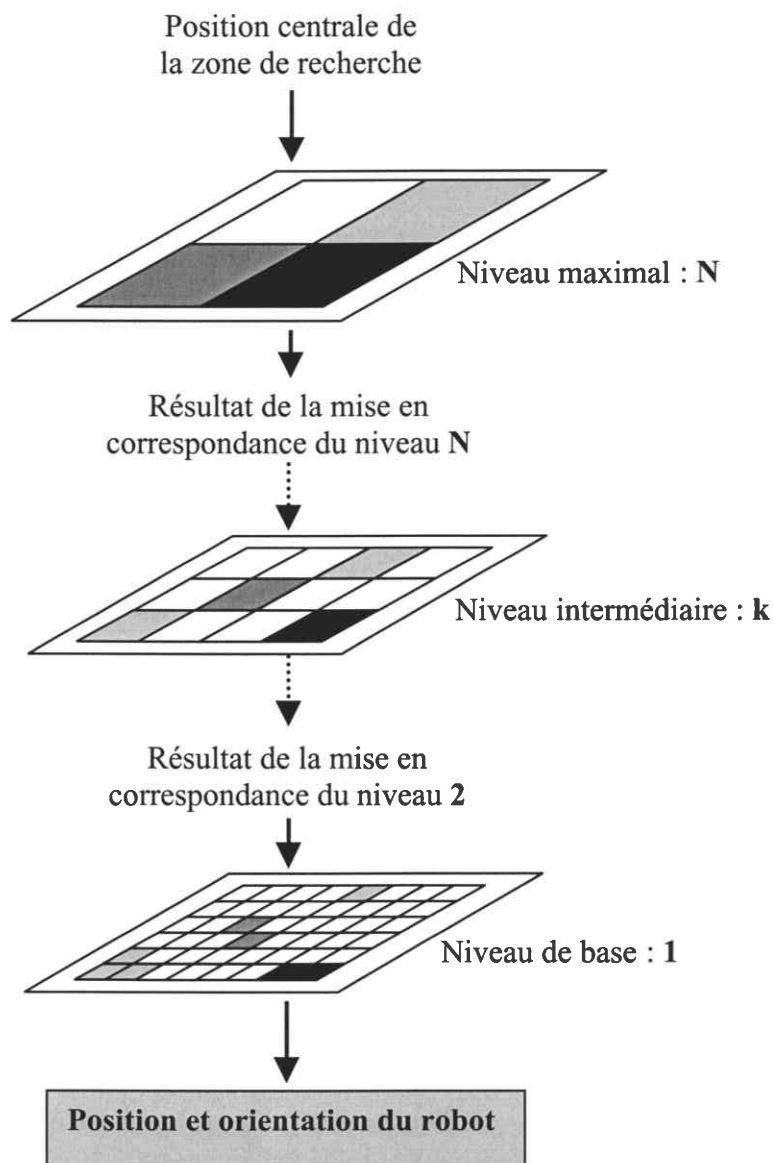


Figure IV-1 : Principe de la recherche hiérarchisée.

IV.2.2 Mise en œuvre

IV.2.2.1 Définition des niveaux

Pour notre application la position du robot doit être précise à 10 cm près, ce qui fixe la taille des cellules des grilles du niveau de base à 10x10 cm. La zone de recherche initiale (du niveau maximal) est fixée par celle de l'application à savoir une zone de recherche en position de ± 75 cm autour de la position estimée et une zone de recherche en orientation de $\pm 45^\circ$ autour de l'orientation estimée. Toutes les cellules de la zone sont testées et le pas de recherche en orientation est de 5 degrés.

L'environnement que nous utilisons ne permet pas d'avoir une résolution supérieure à 20x20 cm par cellule. Certains objets sont de petite taille et placés le long d'un mur ce qui a pour conséquence de n'avoir qu'une cellule occupée dans la grille globale indiquant la présence de l'objet et du mur. La résolution des grilles du niveau maximal est alors fixée à 20x20 cm par cellule. Dans ce cas, l'utilisation d'un niveau intermédiaire n'est pas utile, N est alors égal à 2.

Nous considérons ensuite qu'en plaçant le robot au centre de la cellule donnant la meilleure correspondance, la position de celui-ci est connue à la taille d'une cellule près. Ainsi, au niveau inférieur, la zone de recherche en position est de ± 20 cm autour de la position trouvée.

La zone de recherche en orientation du niveau de base est plus délicate à choisir. Une rotation de 5 degrés de la grille locale au niveau supérieur n'est pas significative et en réduisant trop la zone de recherche en orientation il se peut que l'orientation réelle du robot ne soit pas comprise dans cette zone. Pour avoir l'orientation réelle du robot comprise dans la zone de recherche, nous l'avons alors fixé la recherche à $\pm 30^\circ$ (par pas de 5 degrés) autour de l'orientation trouvée au niveau supérieur.

L'algorithme correspondant à tous ces choix est présenté à la figure IV-2. Dans ce cas le nombre de calculs au niveau 2 est : $o_2 * p_2^2 * n_2^2$ et au niveau 1 : $o_1 * p_1^2 * n_1^2$. En reprenant l'exemple de l'introduction cela représente environ $3,5.10^6$ calculs (avec $o_2=19$, $p_2=9$, $n_2=35$, $o_1=13$, $p_1=5$ et $n_1=70$). Ainsi, le nombre de calculs est divisé par un facteur 10, ce qui permet d'avoir le résultat de la localisation en moins d'une minute.

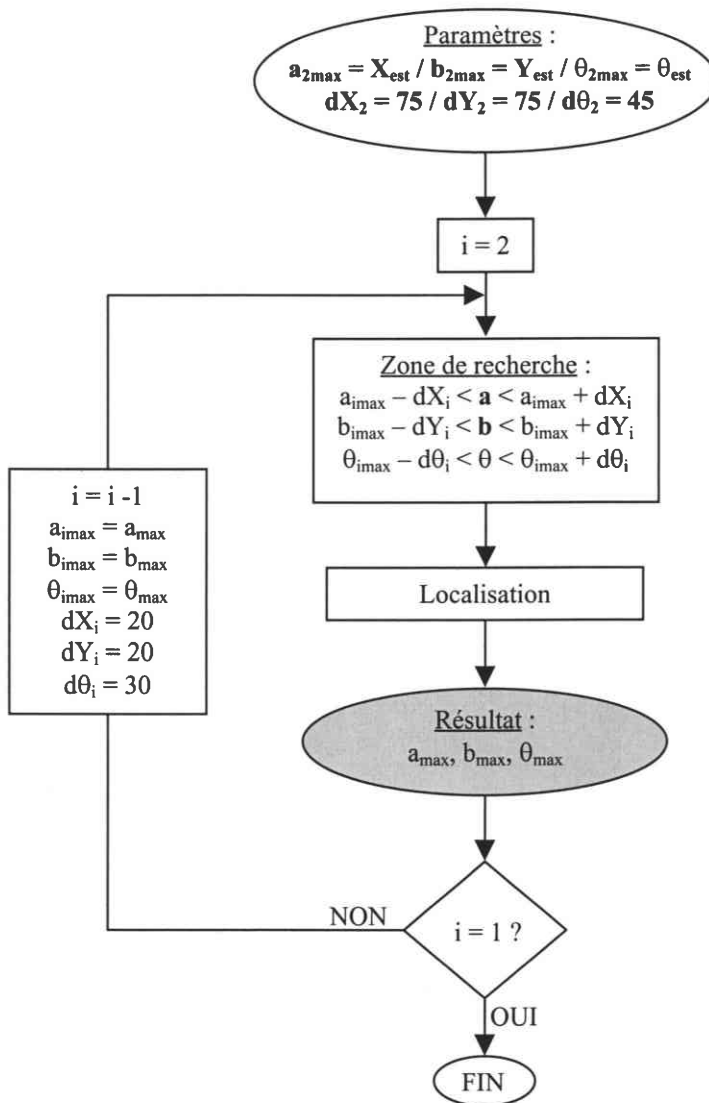


Figure IV-2 : Algorithme de la localisation hiérarchisée.

IV.2.2.2 Première évaluation

A partir des grilles créées au niveau de base suivant les procédés décrits au chapitre III, nous construisons les grilles du niveau 2 par une méthode de compression selon les critères suivants :

- quatre cellules voisines C_i au niveau k produisent une cellule C_j au niveau $k+1$ (Figure IV-3).
- Si l'une des quatre cellules est occupée (la valeur de C_i est positive), la valeur de la cellule C_j sera la valeur maximale des quatre cellules C_i .

- Si aucune cellule n'est occupée et si l'une des quatre est vide (la valeur de C_i est négative), la valeur de la cellule C_j sera la valeur minimale des quatre cellules C_i .
- Si les quatre cellules sont inconnues (les valeurs de C_i sont toutes nulles) la valeur de la cellule C_j sera 0.

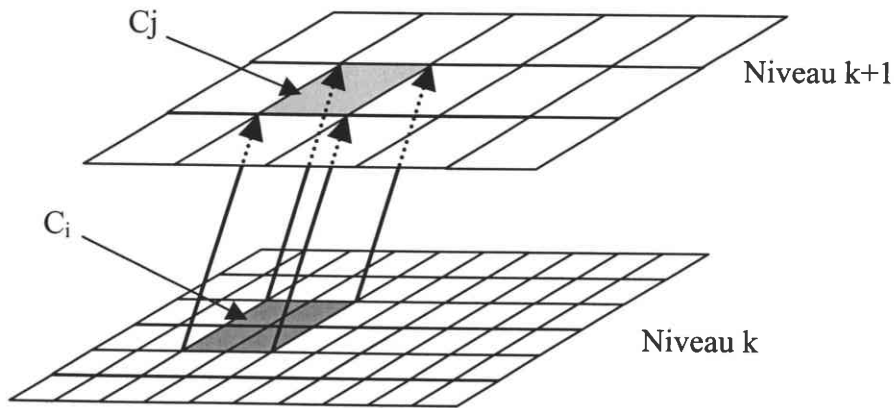


Figure IV-3 : Principe de la compression de grilles.

Ce mode de compression qui donne plus d'importance aux cellules occupées, est fait dans le but de ne pas éliminer des obstacles existants. En effet, augmenter la taille d'un obstacle risque de décaler légèrement le résultat de la recherche (décalage qui sera réajusté lors du traitement ultérieur), alors qu'éliminer un obstacle est susceptible de faire disparaître un repère indispensable à la localisation (erreur non compensable). Sous cette forme, certaines places libres de la pièce pourront paraître occupées à un niveau haut de la hiérarchie, c'est pourquoi nous avons décidé pour la recherche des places possibles du robot de considérer aussi les zones occupées, excepté au dernier niveau où une cellule contenant une valeur positive contiendra nécessairement un objet. Un exemple de compression d'une grille globale est présenté à la figure IV-4.

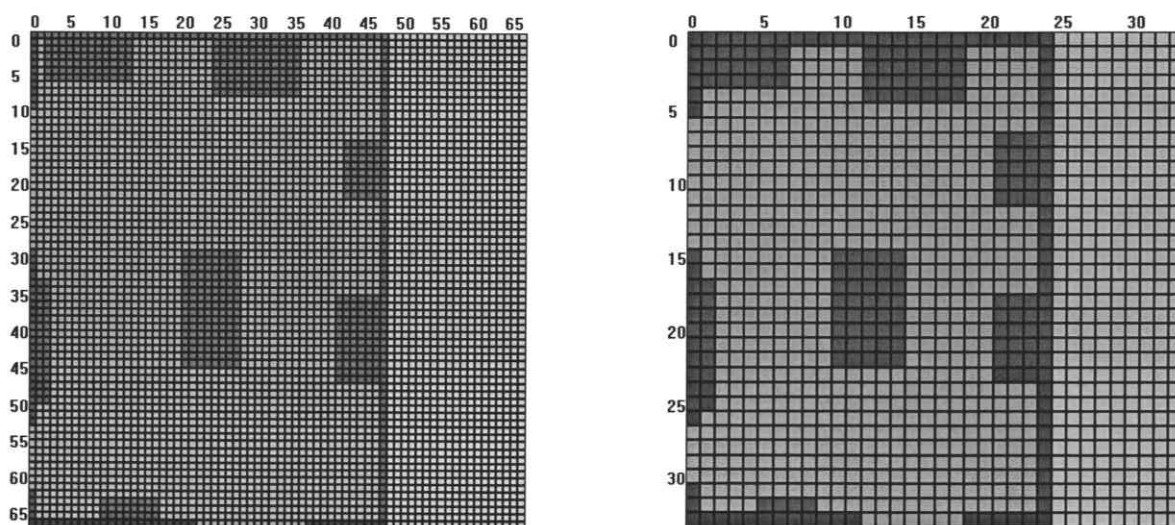


Figure IV-4 : Exemple de compression d'une grille globale.

Cette méthode de compression présente l'avantage d'être rapide puisque nous utilisons des grilles existantes pour créer les suivantes ; la discrétisation des mesures (procédé demandant beaucoup de temps), n'est alors réalisée que pour le niveau de base.

Pour évaluer l'apport de la recherche hiérarchisée, nous avons réalisé des essais en simulation sur un Pentium I 133MHz. Pour une configuration donnée, le calcul a été fait avec, puis sans, l'utilisation des réductions, en relevant le temps nécessaire pour chaque cas. La figure IV-5 donne un exemple numérique étape par étape en montrant le résultat obtenu au niveau supérieur puis au niveau de base et en indiquant les zones de recherche dans chaque niveau. Dans cet exemple, l'utilisation des réductions a permis de retrouver exactement le même résultat en 3 secondes (contre 10 secondes sans les réductions), ce qui diminue le temps de calcul de 70%.

Lors de la mise en place de la méthode nous réalisons les essais correspondant à ceux tenant compte de la correction de la surface du robot présentés dans l'annexe 2. A ce stade, la recherche de cohérence pour la grille locale ainsi que l'affinement de la grille globale n'avaient pas été réalisés. Dans ces conditions, nous avons évalué l'apport de la compression sur l'ensemble des essais simulés sur le Pentium I comme précédemment. Le temps moyen est de 9 secondes sans réduction et de 1,9 secondes en utilisant la compression des grilles, ce qui représente un gain de temps de 79%.

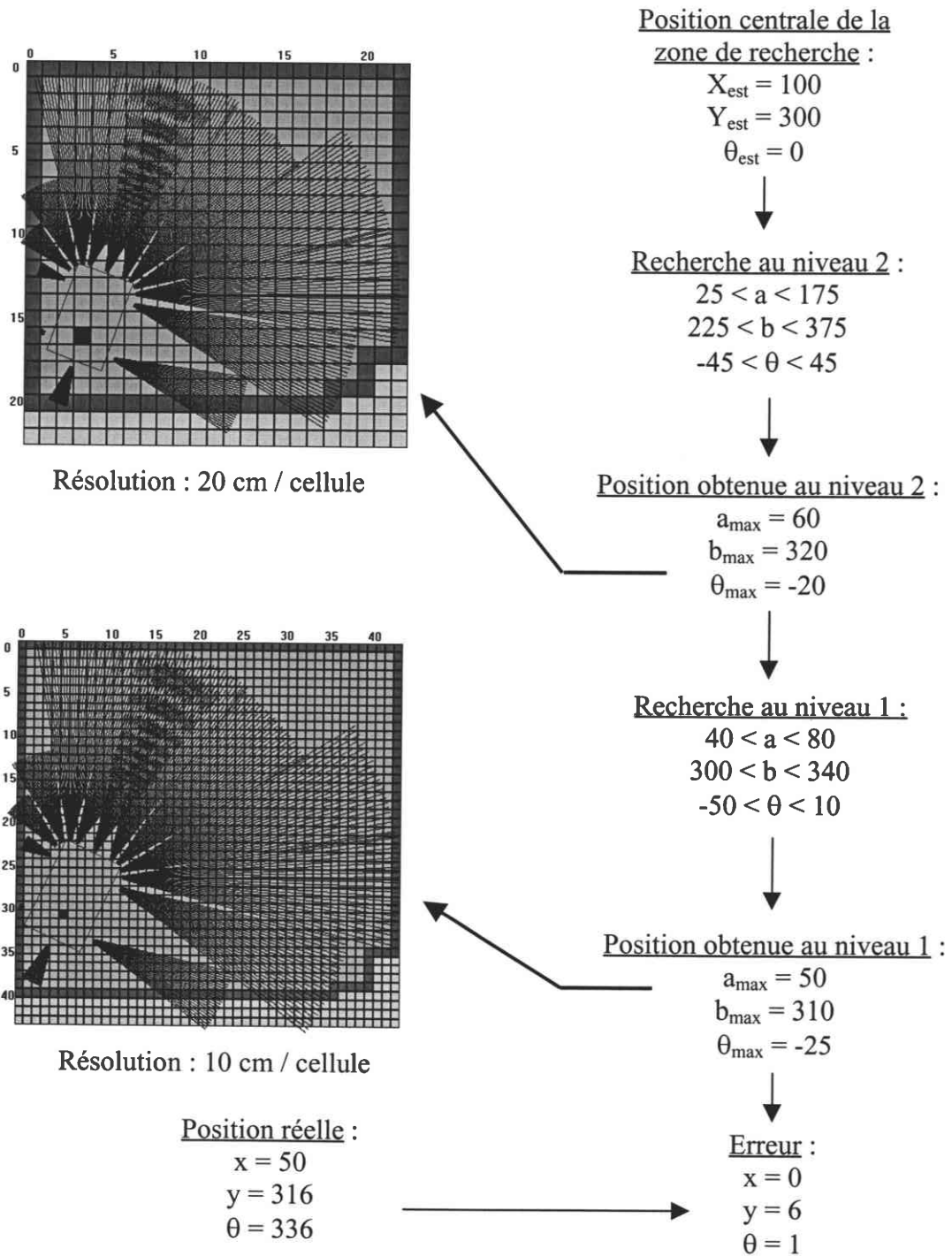


Figure IV-5 : Exemple sur l'utilisation des réductions.

IV.2.2.3 Correction et résultats

L'utilisation de la compression des grilles a prouvé son efficacité en réduisant les temps de calculs. Par ailleurs, pour des questions de cohérence de la grille locale nous avons mis en place une méthode de discrétisation des mesures ainsi qu'une rotation analogique (voir chapitre III-3). La

méthode de compression qui réduit quatre cellules voisines en une seule en utilisant l'ordre de priorité obstacle-libre-inconnu ne conserve pas les choix définis et comme le montre l'exemple de la figure IV-6, des cellules libres peuvent apparaître derrière des obstacles.

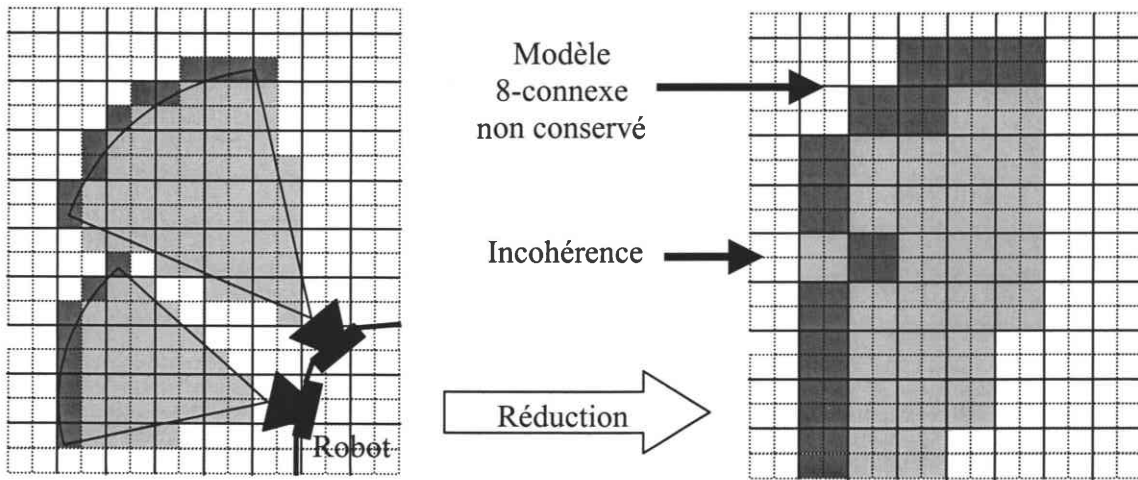


Figure IV-6 : Exemple de réduction induisant des incohérences dans la grille réduite.

Comme nous l'avons décrit au paragraphe III.3.3.2, la mise en place de la rotation analogique nous a amené à construire une nouvelle grille à chaque nouvelle orientation du robot. Dans ce contexte, il est logique de créer les grilles directement réduites en utilisant un paramètre définissant la résolution de la grille (au même titre que le paramètre définissant l'orientation) et en l'introduisant lors de la création de la grille locale (Figure IV-7). Les grilles globales sont également construites suivant ce principe.

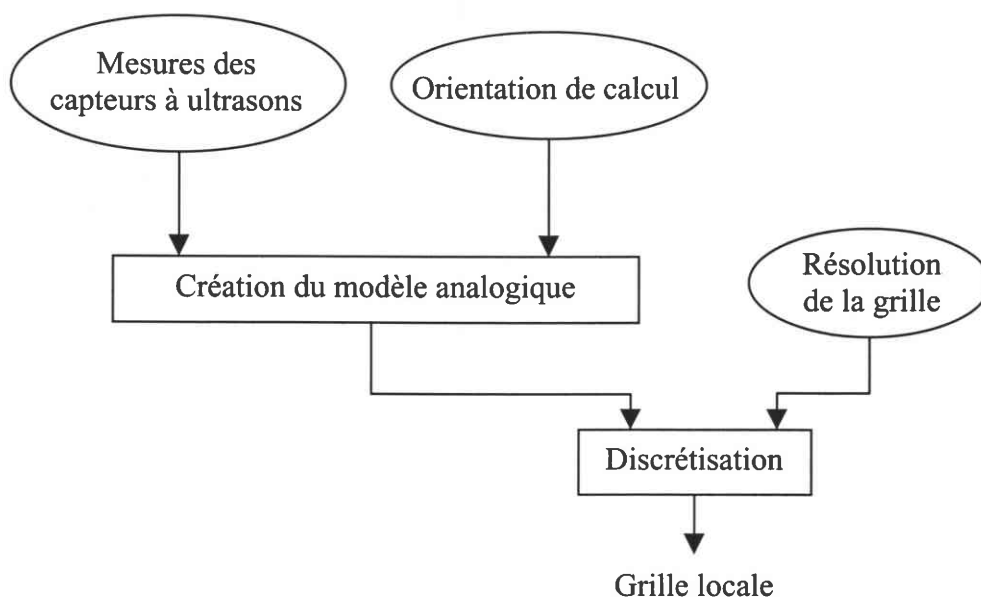


Figure IV-7 : Création des grilles locales réduites.

En reprenant l'exemple présenté à la figure IV-5, le temps pour réaliser la localisation est de 43 secondes en utilisant les réductions (58 secondes sans les réductions), ce qui ne donne qu'un gain de 25% (contre 70% en utilisant les rotations discrètes).

Pour évaluer l'apport des réductions sur un nombre plus important de situations et en conditions réelles, nous avons évalué les temps de calculs pour les 100 configurations du robot dans l'appartement (Annexe 3). A ce stade, les corrections de la grille globale ont également été utilisées. La localisation a été réalisée sur le PC486 embarqué sur la base Robuter, l'ensemble des essais nous fournissent les résultats présentés à la figure IV-8.

	Temps moyen	Ecart Type	Temps minimal	Temps maximal
Sans réductions	72 s	34 s	25 s	219 s
Avec réductions	48 s	25 s	15 s	159 s

Figure IV-8 : Comparaison des temps de calculs sur le PC 486 embarqué.

Il en ressort que l'utilisation des réductions permet en moyenne de réduire les temps de calculs d'un tiers. Nous avons alors évalué l'importance du matériel informatique en calculant les temps mis pour réaliser la localisation en simulation pour ces configurations et sur des PC de puissances différentes (Figure IV-9). Ainsi, on remarque que les réductions permettent d'abaisser les temps de calculs d'environ 30% quel que soit le PC utilisé et que ce temps devient tout à fait acceptable pour l'application avec moins de 6 secondes à partir du PII à 400MHz.

	486 66MHz	Pentium I 133MHz	Pentium II 400MHz	Pentium III 450MHz
Sans réductions	65 s	25 s	5,7 s	5,1 s
Avec réductions	45 s	17 s	4,2 s	3,7 s

Figure IV-9 : Comparaison des temps de calculs pour réaliser la localisation.

La réduction de grilles nous permet donc d'obtenir les mêmes résultats pour la localisation en consommant 30% du temps en moins. La méthode de création de grilles réduites nous permet de conserver la qualité des modèles que nous avons développés au chapitre précédent pour tous les niveaux. Pour réduire davantage les temps de calculs et améliorer la fiabilité de l'algorithme, nous avons envisagé différentes possibilités qui constituent les perspectives décrites par la suite.

IV.3 FILTRAGE DES MESURES

Comme nous l'avons vu au chapitre III.2.2, la différence entre les résultats obtenus en simulation et ceux réalisés sur le robot provient des mesures des capteurs à ultrasons qui sont sujets aux phénomènes de spécularité et de diaphonie. Nous allons essayer de détecter ces phénomènes pour limiter leur influence sur la localisation du robot.

IV.3.1 Seuillage

Afin de limiter l'influence de la spécularité, le premier filtrage que nous avons introduit consiste à supprimer toutes les mesures supérieures à un seuil fixé par la taille de la pièce (ou de l'appartement). Nous avons déterminé la distance maximale mesurable dans une pièce donnée par :

$$D_{MAX} = \sqrt{l^2 + L^2} - l_R$$

avec l la largeur de la pièce, L sa longueur et l_R la largeur du robot ; ce qui revient à placer le robot dans un coin de la pièce de façon à ce que l'un de ses capteur situé sur son côté vise le coin opposé de la pièce (Figure IV-10).

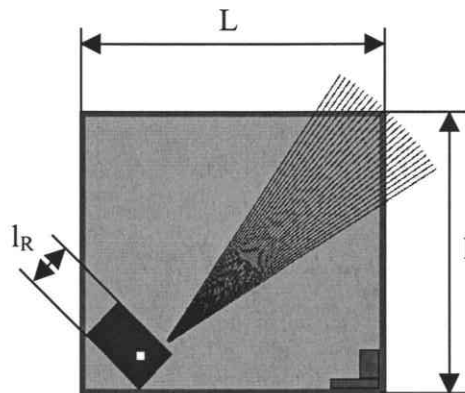


Figure IV-10 : Configuration du robot pour la distance maximale mesurable.

Les mesures supérieures à cette distance sont forcément liées à des phénomènes parasites et sont mises à 0 afin de ne pas les utiliser pour la localisation. Ce filtre permet en outre de limiter la

taille de la grille locale, ce qui réduit le temps de calcul. L'exemple présenté à la figure IV-11 illustre un cas où le robot est proche d'un mur ce qui induit des réflexions multiples pour les capteurs situés à l'avant droit du robot. Le filtre supprime alors la mesure du capteur n°4 et permet de réduire la taille de la grille locale à 80x80 cellules au lieu de 140x140 cellules, ce qui se traduit par un gain de temps de 70% (on passe de 126 à 37 secondes de calcul sans réductions et de 89 à 26 secondes avec une réduction).

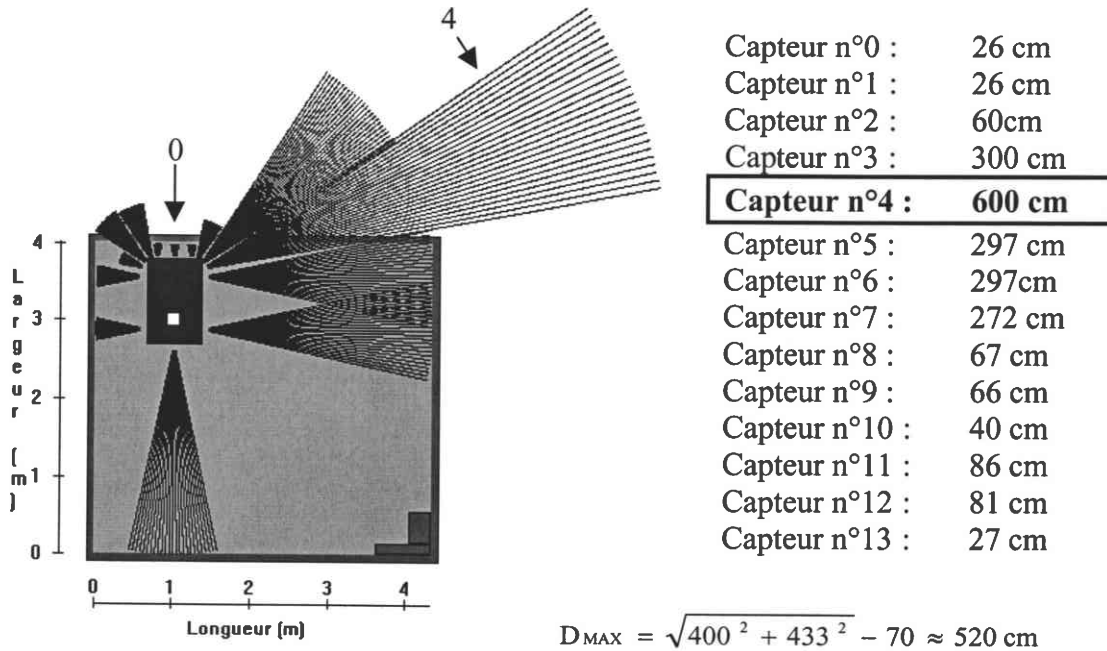


Figure IV-11 : Filtrage des mesures par seuillage.

On remarque également dans cet exemple que la mesure n°3 mériterait d'être supprimée au même titre que la mesure n°4. Dans cet exemple, cette mesure n'est pas dominante et le programme a su trouver le bon résultat ; mais il arrive que ce genre de mesures mettent en défaut l'algorithme en plaçant le robot sur une fausse position. En effet, les réflexions multiples induisent des mesures qui sortent de l'environnement, l'algorithme tend alors à ramener ces mesures à l'intérieur de la pièce où la superposition est correcte. C'est le cas de l'exemple de la figure IV-12 où la mesure du capteur n°7 (en haut à droite) est sujette aux réflexions multiples, ce qui fausse le résultat en position de près de 80 cm et induit une erreur en orientation de 35 degrés ; en supprimant cette mesure l'algorithme peut alors trouver un bon résultat.

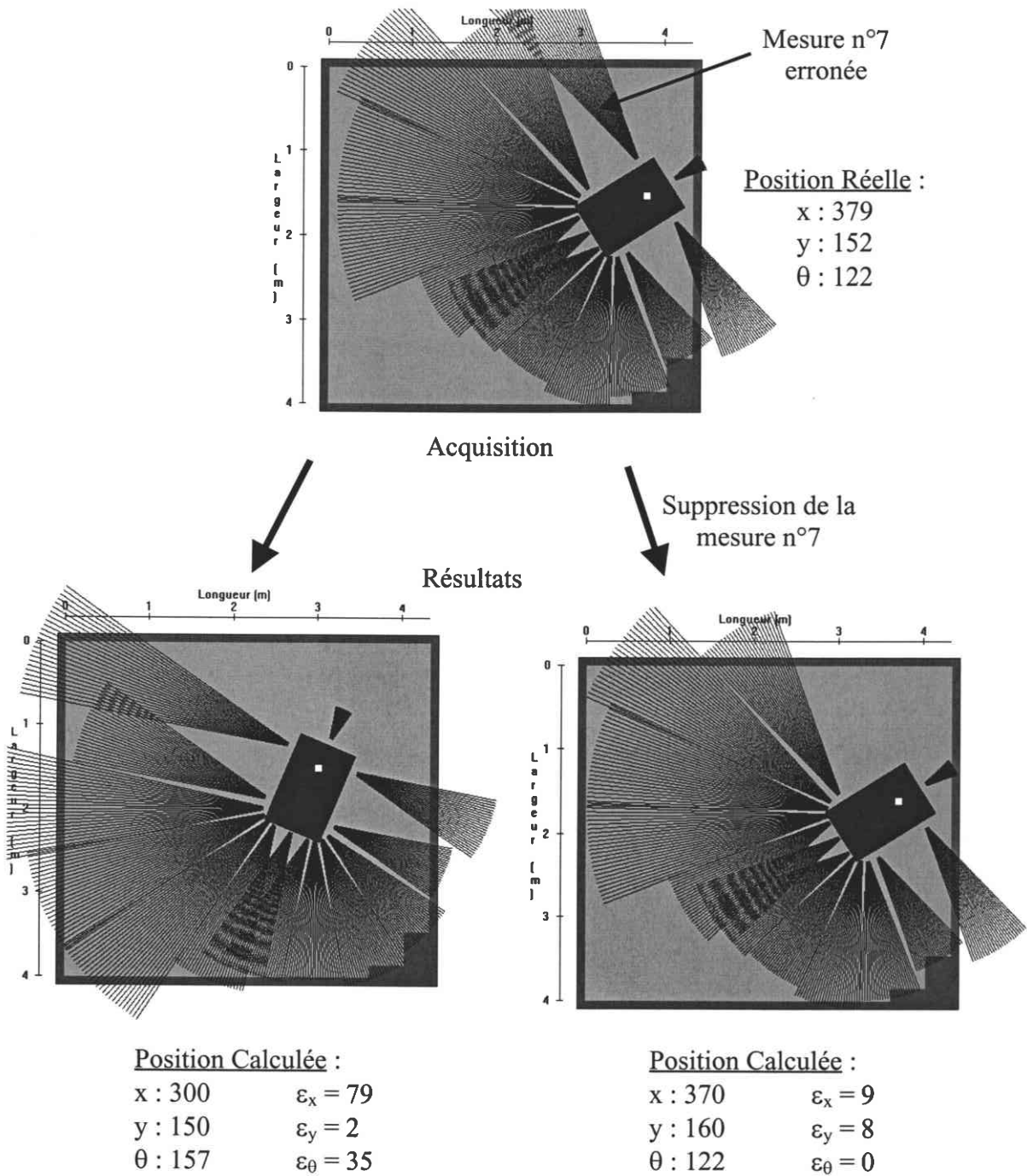


Figure IV-12 : Réflexions multiples faussant la localisation.

Pour trouver un moyen de supprimer ces mesures aberrantes nous avons étudié les données issues des capteurs en relevant plusieurs mesures successives, ce qui fait l'objet de la partie suivante.

IV.3.2 Moyenne

Pour voir si les mesures aberrantes se reproduisent, nous avons réalisé un deuxième filtrage consistant à moyenner cinq mesures consécutives. Nous avons constaté que pour la majorité des acquisitions, les mesures au sein d'un même tir sont sensiblement égales. Pour les cas où il apparaît une mesure aberrante, la moyenne des cinq permet d'avoir une distance convenable. Les exemples de la figure IV-13 présentent différents cas rencontrés, les exemples 1 et 2 montrent que la moyenne permet de limiter la présence d'une mesure trop grande dû aux réflexions multiples, l'exemple 3 illustre le cas où une mesure est sujette à un phénomène de diaphonie, enfin l'exemple 4 illustre le cas où une mesure dépassant la distance D_{max} est alors mise à 0 et n'est pas utilisée pour la moyenne.

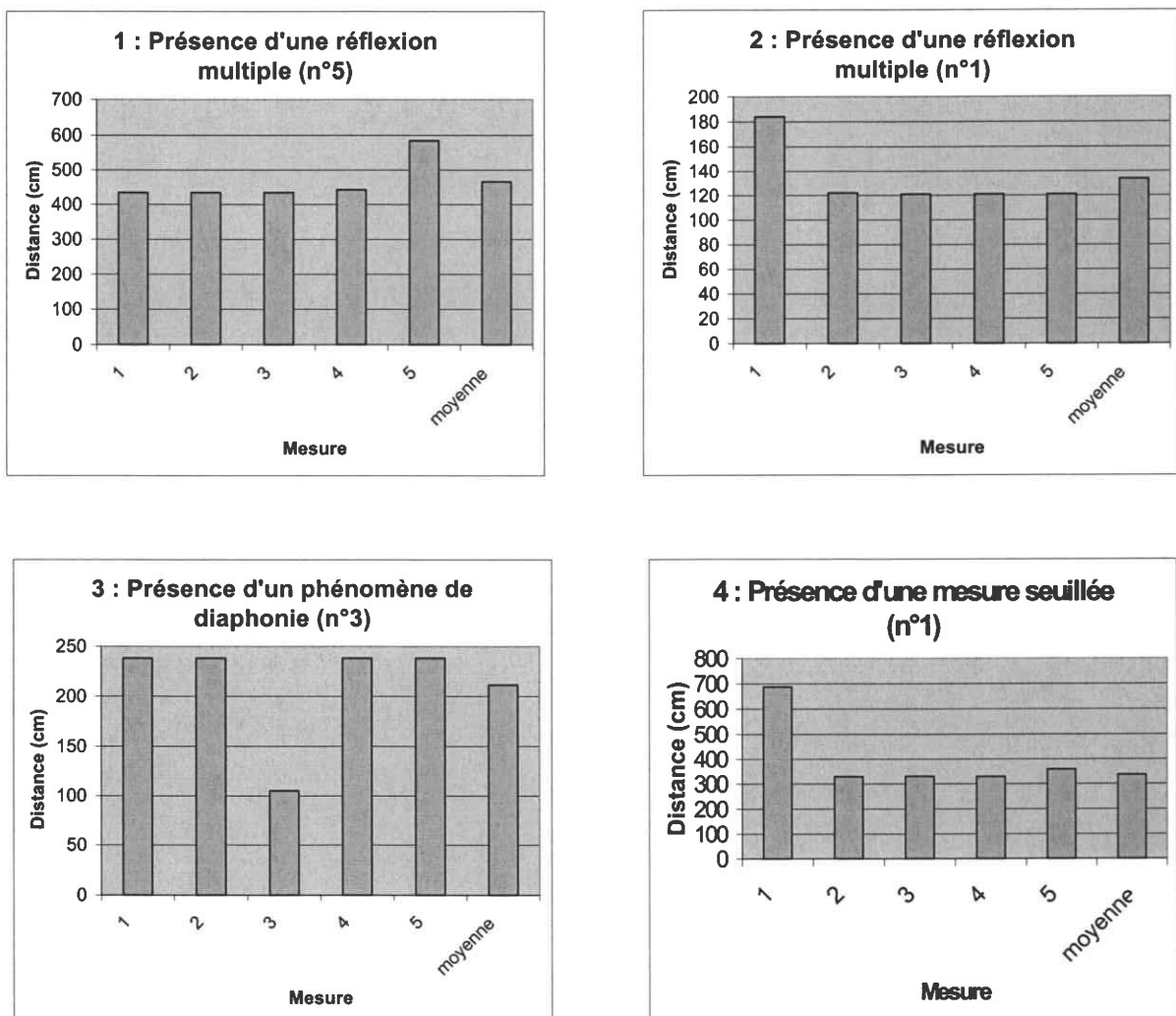


Figure IV-13 : Filtrage des mesures par moyenne.

La moyenne de plusieurs mesures permet de limiter l'influence des erreurs occasionnelles. Dans le dernier cas présenté (n°4), les acquisitions successives permettent d'avoir une mesure pour le capteur, mesure qui aurait été nulle suite au seuillage en ne faisant qu'une acquisition. L'effet des phénomènes parasites est limité en moyennant plusieurs mesures successives, cependant il arrive que les cinq mesures fournissent une distance sensiblement égale mais fausse ; la moyenne dans ce cas n'est pas efficace.

Par ailleurs, lors de nos essais en conditions réelles (Annexe 3), nous avons pu mettre en évidence que le tiers des mauvaises localisations sont dues aux phénomènes de diaphonie. En remplaçant dans les essais les mesures trop courtes issues de la diaphonie par des mesures simulées, nous corrigeons les résultats présentant de grandes erreurs et obtenons 87 cas satisfaisant les conditions de réussite sur la centaine de configurations testées sur le robot. Les problèmes liés à la diaphonie peuvent être réduits en appliquant la méthode EERUF développée par BORENSTEIN [BOR92]. Cette méthode consiste à réaliser des tirs programmés en définissant un ordre pour un groupe de capteurs voisins. Pour chaque capteur du groupe, une temporisation variable est introduite avant le tir et plusieurs mesures successives sont comparées. Si un phénomène de diaphonie est présent, deux mesures successives ne seront pas égales, la dernière acquisition est alors ignorée. L'application développée sur un fauteuil électrique prévoit la mise en place de cette méthode.

IV.4 PERSPECTIVES D'OPTIMISATION POUR LES GRILLES LOCALE ET GLOBALE

Dans cette partie nous allons donner les axes de recherche que nous envisageons dans nos futurs travaux pour réduire les temps de calculs et améliorer les résultats. La première idée concerne une diminution de la taille de la grille locale en éliminant les cellules ne contenant pas d'informations. Le second sujet s'attache à améliorer les résultats en conditions réelles en ne considérant dans la grille globale que la partie de l'environnement perceptible depuis le robot. Le dernier thème porte sur la méthode de mise en correspondance : il s'agit d'appliquer les translations et les rotations sur la grille globale afin de les exécuter à l'avance, ce qui réduira les temps de calcul.

IV.4.1 Dimension de la grille locale

Lors de notre étude sur le filtrage des mesures dans la partie précédente, nous avons mis en évidence l'importance de la taille de la grille locale. En effet, en passant de 140x140 cellules à 80x80 cellules, le gain de temps était de 70%. Il est donc intéressant d'ajuster les dimensions de la grille au minimum.

La taille de la grille locale est définie par la distance la plus grande mesurée par les capteurs à ultrasons. Dans nos essais préliminaires nous avons défini cette grille carrée pour permettre sa rotation lors de la mise en correspondance. L'utilisation du modèle analogique par la suite ne justifie plus cette taille surdimensionnée. En effet, le robot est directement orienté dans la direction voulue ; à chaque nouvelle orientation une nouvelle grille est créée (voir chapitre III-3). Aussi, quand une mesure se détache des autres, il apparaît dans la grille des zones qui ne sont pas utilisées. L'exemple de la figure IV-14 présente ce cas avec une grille locale pour laquelle la surface utile représente moins de 20% de la surface totale.

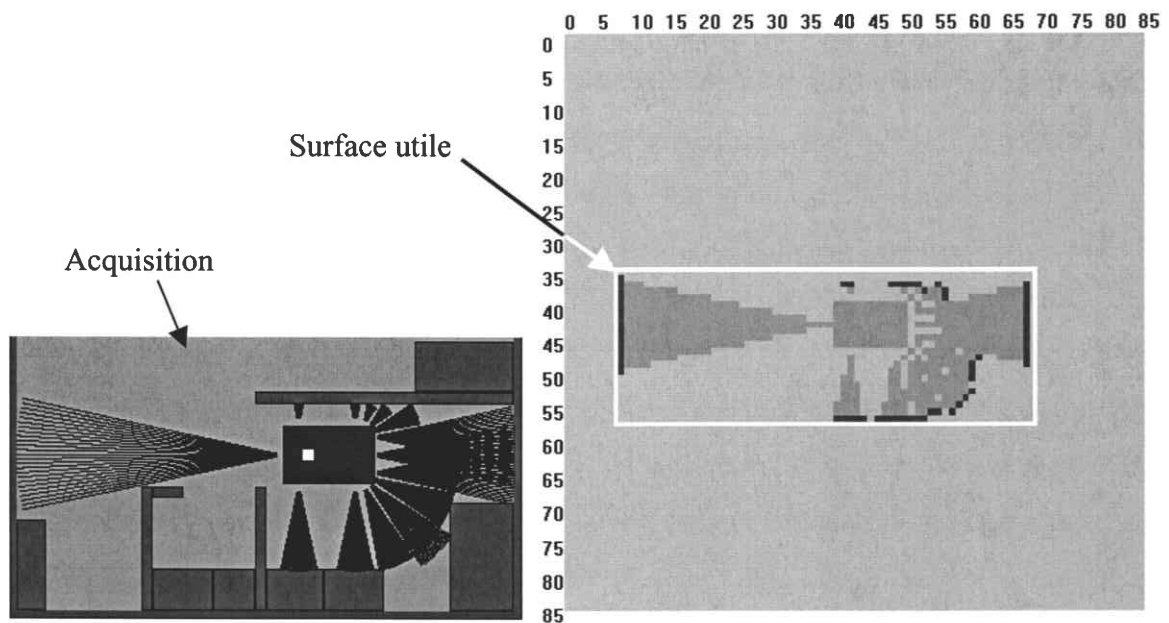


Figure IV-14 : Surface utile d'une grille locale.

La solution que nous proposons est de construire une grille locale initiale suivant la méthode présentée au chapitre III.3.3.2, puis de la réduire à la surface utile en supprimant toutes les colonnes et les lignes ne contenant que des 0, ce qui rend la grille rectangulaire. Ceci peut se faire

simplement grâce à un algorithme comme celui présenté en annexe 4. Nous avons estimé l'apport en sélectionnant un ensemble de grilles locales caractéristiques des situations rencontrées pour lesquelles nous calculons le rapport de la surface utile sur la surface totale. Sur les 15 grilles sélectionnées, la surface utile représente en moyenne 35% de la surface totale, dans une plage de 23% à 48%, cependant, nous ne pouvons pas évaluer l'apport sur les temps de calculs. En effet, la taille de la grille et donc le temps de calcul dépend de l'orientation du robot. Pour une localisation, il est nécessaire de comparer plusieurs orientations du robot et donc de créer plusieurs grilles locales de tailles différentes. En utilisant des grilles rectangulaires, la surface peut varier du simple au double comme le montre l'exemple de la figure IV-15 qui reprend l'exemple de la figure IV-14 en orientant le robot à 325°. Le robot placé à 270° permet d'obtenir une surface utile de 61x22 = 1342 cellules, à 315° elle est de 52x52 = 2704 cellules. Ainsi, nous pouvons seulement dire qu'en se plaçant dans le pire des cas il est possible de réduire les surfaces des grilles locales d'au moins la moitié et nous sommes certains de diminuer les temps de calcul.

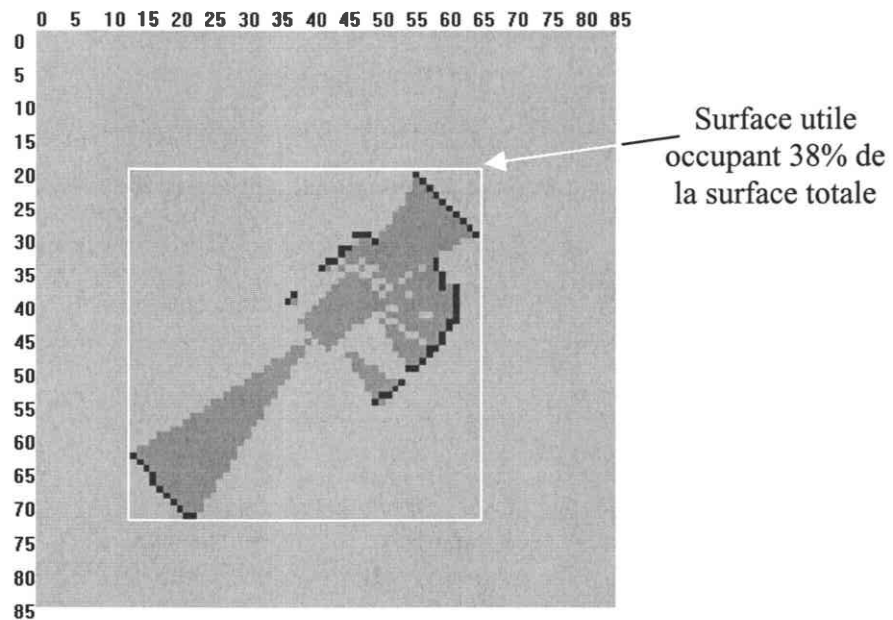


Figure IV-15 : Surface utile en plaçant le robot à 315°.

IV.4.2 Adaptation de la grille globale

IV.4.2.1 Partie visible

La grille globale représente l'environnement d'évolution du robot dans sa totalité. Lorsque cet environnement comprend de nombreux objets ou bien différentes pièces comme c'est le cas de notre appartement reconstruit, certaines parties sont cachées par les obstacles. Ainsi, nous pouvons réduire la grille globale à la partie visible depuis la position du robot (Figure IV-16).

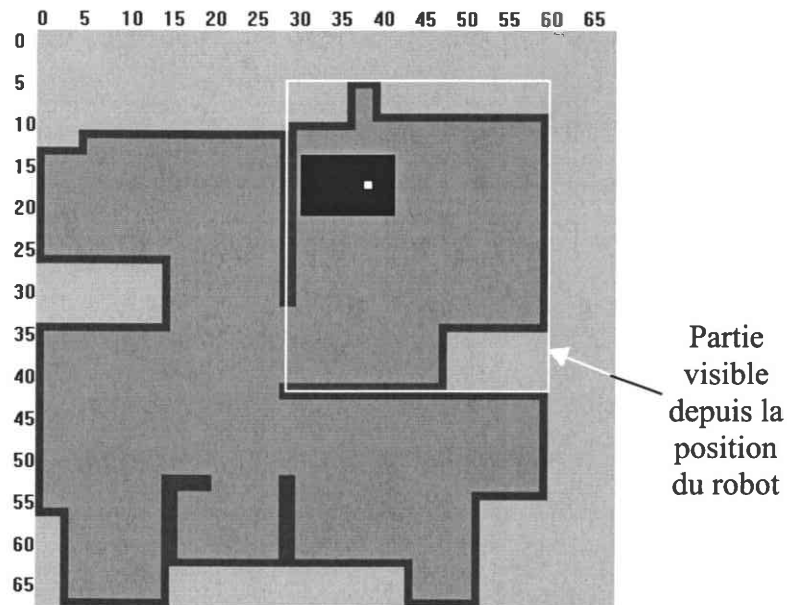


Figure IV-16 : Réduction de la grille globale.

Ne considérer que la partie visible a un intérêt majeur pour la réussite de la localisation. En effet, dans certains cas, les mesures des capteurs à ultrasons sont mises en correspondance au-delà de la limite perceptible (Figure IV-17). L'exemple suivant présente le cas où l'algorithme trouve une meilleure position en plaçant les mesures sur un mur qui n'est pas visible ce qui induit une erreur en position de 40 cm et une erreur en orientation de 27 degrés.

Par ailleurs, le mur séparant deux pièces est d'une largeur de 14 cm, la résolution de la grille étant de 10 cm par cellule le mur est alors représenté par deux rangées de cellules occupées. Il arrive alors que des mesures soient placées sur la deuxième rangée, ce qui peut induire des erreurs lors de la localisation. Le robot ne pouvant être dans les deux pièces en même temps, l'une des deux rangées est superflue pour la localisation.

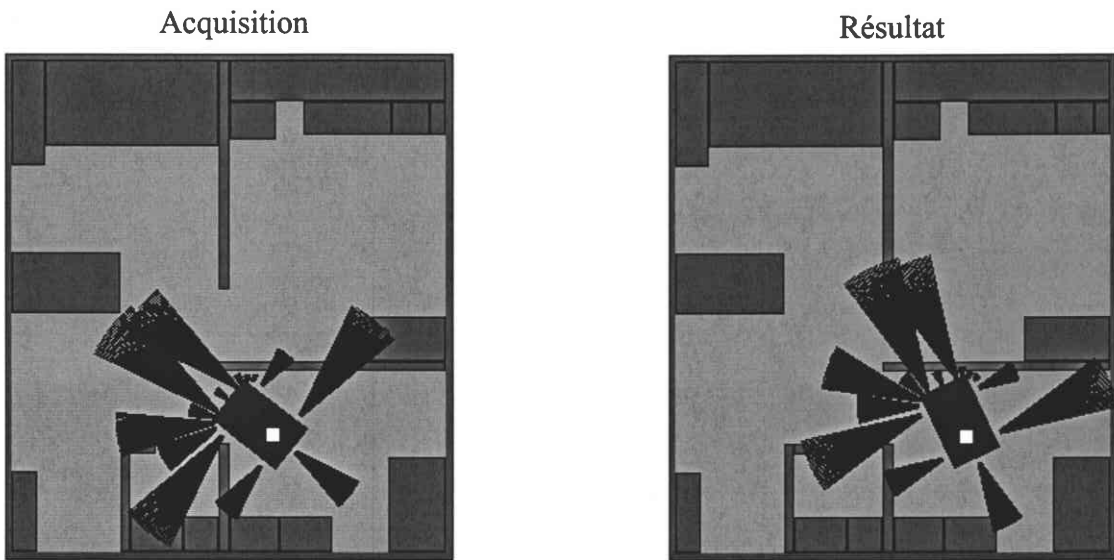


Figure IV-17 : Configurations où les mesures traversent les murs.

La partie visible de la grille globale dépend de la position et l'orientation du robot, or tout ce dont nous disposons c'est une position approximative permettant de définir une zone de recherche. Pour obtenir cette partie, nous plaçons le robot sur les quatre positions extrêmes de la zone de recherche, ne connaissant pas son orientation nous réalisons alors un '*lancer de rayon*' dans toutes les directions comme si nous avons un capteur parfait tournant (Annexe 5). Tant qu'un rayon traverse des cellules libres, ces dernières sont gardées dans la nouvelle grille, quand le rayon touche un obstacle et donc une cellule occupée, il prend en compte cette cellule puis s'arrête. La nouvelle grille représente alors la partie visible de la grille initiale, sa taille peut ensuite être limitée suivant le même principe que celui exposé pour la grille locale.

En limitant ainsi la grille globale nous prévoyons d'améliorer sept cas dans les essais réalisés dans l'appartement (Annexe 3), ce qui nous permettrait d'obtenir un taux de réussite de 94% (avec le filtrage de la diaphonie). Le temps de calcul en serait aussi réduit puisque nous n'aurions plus à effectuer le produit de la superposition des grilles au-delà des murs.

IV.4.2.2 Rotation

Nous pouvons également réduire le temps de calcul en réalisant les rotations sur la grille globale afin d'éviter celles de la grille locale. En effet, la rotation de la grille locale nécessite la création d'un nouveau modèle analogique puis la création d'une nouvelle grille. Contrairement au modèle local qui n'est disponible qu'après avoir réalisé l'acquisition des mesures, le modèle global

est défini et figé. Nous pouvons donc, lors de la phase d'enregistrement du modèle global, réaliser toutes les rotations possibles de l'environnement et les mémoriser. En discrétisant une pièce de 6x6 mètres par des cellules de 10x10 cm, la place mémoire requise serait de moins de 8 Moctets pour les 360 grilles correspondant à toutes les rotations possibles. Les grilles globales n'étant constituées que de 0 de 1 et de -1 il est facile d'utiliser un algorithme de compression qui réduirait la place mémoire requise à quelques centaines d'octets. Ainsi, lors de la localisation, il suffit de rappeler les grilles correspondant aux orientations voulues et de créer un seul modèle et une seule grille locale pour réaliser la mise en correspondance. Pour évaluer l'apport de cette méthode nous avons estimé le temps nécessaire à la création d'une grille locale et le temps requis pour obtenir une grille globale. La difficulté de cette estimation réside dans le choix de la grille locale qui fixe les temps de calculs. Les distances fournies par les 14 capteurs et l'orientation du robot en sont des paramètres variables suivant la localisation. Nous avons alors décidé de réaliser l'estimation sur trois cas particuliers parmi les essais réalisés dans l'appartement. Ces trois cas sont celui pour lequel le calcul est le plus rapide (8 secondes), celui qui est le plus lent (65 secondes), ainsi qu'un essai fournissant un temps moyen (25 secondes). Toute l'étude a été réalisée sur le Pentium I 133Mhz pour des cas simulés avec une résolution de 10 cm par grilles et en se plaçant au niveau de base de la hiérarchie. Les temps de calculs présentés sont une moyenne obtenue en réalisant un millier de fois le même événement et donnent ainsi un ordre de grandeur.

Dans la méthode actuelle, nous créons 1 grille globale et 19 grilles locales (car il y a 19 orientations à comparer). De plus, chaque grille locale nécessite la création d'un modèle analogique. Ainsi, la création de la grille globale correspondant à l'appartement requiert moins d'une dizaine de millisecondes ; la grille locale demande des temps de l'ordre de 0,3 secondes pour l'essai le plus rapide, de 1 seconde pour une configuration fournissant un temps moyen et de 2,5 secondes pour l'essai le plus lent. Dans la nouvelle méthode, il faut lire sur le disque dur 19 grilles globales mémorisées. Le temps de lecture dépend de la taille de la grille qui est fonction de son orientation. Nous nous sommes alors placés dans le pire des cas, c'est à dire que pour l'appartement qui nécessite un codage de 60x60 cellules, à 45 degrés cet environnement demande une grille d'environ 85x85 cellules. Le temps requis est alors de l'ordre d'une cinquantaine de millisecondes. Ainsi, en comparant les deux procédés (Figure IV-18), il s'avère que la méthode que nous proposons permettrait d'avoir le résultat trois fois plus rapidement. En extrapolant, nous pouvons alors prévoir l'exécution de la localisation en moins d'une seconde sur un ordinateur de dernière génération et en utilisant la méthode de recherche hiérarchisée.

Type de localisation	Méthode actuelle			Méthode proposée		
	+ rapide	moyenne	+ lente	+ rapide	moyenne	+ lente
Temps pour 1 grille globale	10 ms			55 ms		
Temps pour 1 grille locale	0,3 s	1 s	2,5 s	0,3 s	1 s	2,5 s
Temps pour créer toutes les grilles	6 s	19 s	48 s	1,3 s	2 s	3,5 s
Temps total de la localisation	8 s	25 s	65 s	3,3 s	8 s	20 s

Figure IV-18 : Estimation et comparaison des temps de calculs pour la méthode proposée.

IV.5 CONCLUSION

Dans ce chapitre nous avons expliqué comment nous avons réduit les temps de calcul de près d'un tiers par une procédure de recherche hiérarchisée qui consiste à réaliser la localisation sur plusieurs niveaux de grilles à différentes résolutions. Nous avons également développé un filtrage des mesures qui permet d'une part de limiter les distances trop grandes par rapport à la taille de l'environnement et d'autre part d'éviter les erreurs ponctuelles en moyennant plusieurs mesures successives. Concernant la taille de la grille locale, les essais permettent d'envisager de la réduire de moitié. Enfin, l'application d'une rotation sur la grille globale et la mémorisation de ces dernières réduirait les temps de calculs à moins de 10 secondes sur un Pentium I à 133 MHz, ce qui est dans un ordre de grandeur acceptable pour l'application.

CONCLUSION

CONCLUSION

L'objet du travail décrit dans ce mémoire est la réalisation d'une méthode de localisation d'un véhicule mobile autonome qui réponde aux caractéristiques suivantes :

- ☞ utilisable dans tout type d'environnement intérieur sans nécessiter d'aménagements particuliers,
- ☞ réalisable quel que soit l'emplacement du robot et sans le déplacer,
- ☞ employant les mesures de capteurs à ultrasons fixes et une information vague de la position du robot,
- ☞ donnant une précision suffisante pour réaliser une trajectoire,
- ☞ fournissant le résultat en quelques secondes.

L'un des premiers problèmes qui se posait était de déterminer le mode de représentation des mesures et de l'environnement ; nous avons opté pour les grilles d'occupation qui permettent de réunir les trois premières conditions. La méthode de localisation consiste à mettre en correspondance deux grilles : la première représentant l'environnement de travail, la seconde décrivant l'espace du robot perçu par les capteurs. La meilleure correspondance est évaluée par un critère permettant de déterminer la translation et la rotation d'une grille par rapport à l'autre. Connaissant la position du robot dans la grille issue des mesures, nous en déduisons la configuration du robot dans l'environnement. Nous avons étudié en détail les grilles pour qu'elles représentent au mieux les mesures des capteurs à ultrasons et qu'elles s'adaptent au mode de localisation par mise en correspondance. Les temps de calculs sont rendus acceptables grâce à la mise en place d'un mode de recherche hiérarchisée. L'efficacité de la méthode est vérifiée dans nos essais en simulation où nous trouvons la position réelle du robot à partir d'une position approximative qui en est éloignée. Les essais réalisés sur la base mobile montrent que l'algorithme peut être utilisé en conditions réelles. Toutefois, les problèmes inhérents à l'utilisation des capteurs à ultrasons tels que les réflexions multiples et la diaphonie mènent à quelques cas d'échec.

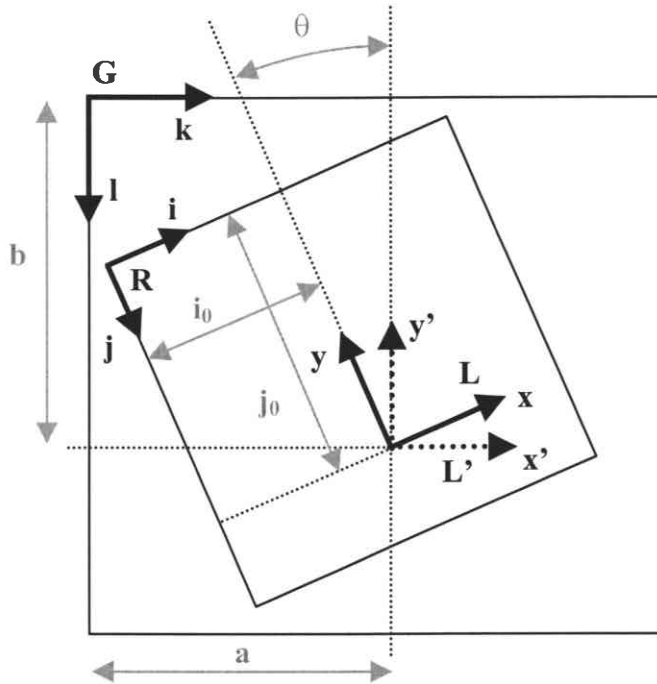
Compte tenu de cette incertitude et de la particularité de l'application, le résultat de la localisation est présenté à l'écran afin que la personne puisse voir si le résultat est acceptable, dans le cas contraire elle a la possibilité d'indiquer la présence d'une aberration. Même si la procédure de localisation demande une confirmation à l'utilisateur, cela est moins contraignant pour le pilote que s'il avait à déterminer tout seul l'emplacement exact du fauteuil.

La méthode que nous avons développée ne se cantonne pas uniquement à notre projet. Toute application de robotique mobile d'intérieur peut prévoir sa mise en place moyennant quelques modifications :

- ↳ La première concerne les capteurs. La méthode est applicable quelle que soit la disposition des capteurs sur la ceinture, et peut aussi convenir à un capteur à ultrasons tournant. L'utilisation d'autres capteurs télémétriques (par exemple laser ou infrarouges) est également possible, il faut alors adapter les représentations des mesures aux capteurs utilisés.
- ↳ La seconde porte sur la personne utilisant le fauteuil. Dans notre cas, elle indique une position vague pour démarrer la recherche et contrôle la cohérence du résultat. Pour certaines applications de robotique mobile cette personne n'existe pas et il faut pallier ces deux manques :
 - La position de départ peut être fournie grossièrement par l'opérateur ce qui évite de déterminer un point précis et de devoir placer le robot sur cette position avant de réaliser une tâche de navigation, ce qui se fait dans beaucoup d'applications actuelles. Si aucune information n'est disponible, il faudrait réaliser la recherche dans l'environnement entier, il serait alors intéressant d'optimiser le temps.
 - Pour contrôler la cohérence du résultat, une étude sur un critère de qualité indiquant la réussite de la localisation serait indispensable, il permettrait de donner une fiabilité à la localisation.

ANNEXES

ANNEXE 1 : Formules de changement de repères



Passage du repère R à L :

$$\begin{cases} x = i - i_0 \\ y = j_0 - j \end{cases}$$

Passage du repère L à L' :

$$\begin{cases} x' = x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ y' = x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{cases}$$

Passage du repère L' à G :

$$\begin{cases} k = a + x' \\ l = b - y' \end{cases}$$

D'où le passage du repère R à G :

$$\begin{cases} k = a + (i - i_0) \cdot \cos(\theta) + (j - j_0) \cdot \sin(\theta) \\ l = b - (i - i_0) \cdot \sin(\theta) + (j - j_0) \cdot \cos(\theta) \end{cases}$$

ANNEXE 2 : Essais relatifs à la grille locale

Le tableau ci-dessous présente les résultats obtenus dans une pièce vide de 4,3 x 4 mètres en simulation et en conditions réelles. Les positions X et Y du robot sont exprimés en cm, les orientations θ du robot sont en degrés (0 degré correspond à une orientation dirigée vers le haut). La différence entre la position réelle et la position calculée est indiquée par les erreurs ϵ_x et ϵ_y en cm pour les erreurs en position, par l'erreur en orientation ϵ_θ en degrés et par l'erreur en position ϵ_c en nombre de cellules. Les cases grisées indiquent les localisations pour lesquelles au moins une des deux conditions de réussite n'est pas respectée à savoir une erreur en position supérieure à une cellule et une erreur en orientation supérieur à 7 degrés.

La colonne « *Modélisation de base en 1/n* » correspond aux essais décrit au chapitre III.2.1, celle intitulée « *Correction de la surface du Robot* » montre les essais exposés au chapitre III.2.2.

N° d'essai	Position Réelle			Modélisation de base en 1/n								Correction de la surface du Robot							
				Erreurs en Simulation				Erreurs en Réel				Erreurs en Simulation				Erreurs en Réel			
	X	Y	θ	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c
1	100	100	0	0	0	0	0	0	-30	0	3	0	0	0	0	0	-30	0	3
2	100	100	45	0	0	5	0	0	0	5	0	0	0	0	0	0	0	5	0
3	100	200	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	200	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	200	200	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	203	205	0	3	5	0	1	3	5	0	1	3	5	0	1	3	5	0	1
7	300	200	270	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	300	300	240	0	0	0	0	10	-10	10	1	0	0	0	0	-10	30	40	3
9	350	250	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	378	257	180	8	-3	-5	1	-2	-3	5	0	8	-3	-5	1	-2	-3	5	0
11	146	204	40	6	4	5	1	6	4	10	1	6	4	5	1	6	4	10	1
12	312	72	289	2	2	0	0	2	2	10	0	2	2	0	0	2	2	10	0
13	223	185	222	3	-5	0	0	3	-5	0	1	3	-5	0	0	3	-5	0	1
14	103	189	93	3	-1	5	0	3	-1	0	0	3	-1	5	0	3	-1	0	0
15	69	72	230	9	-8	-5	1	-1	-28	0	3	9	-8	-5	1	-1	-28	0	3
16	161	46	270	1	-4	0	0	1	-4	0	0	1	-4	0	0	1	-4	0	0
17	344	71	236	4	1	0	0	74	-59	15	7	4	1	0	0	84	-39	-25	8
18	379	152	122	-1	-8	0	1	19	-18	0	2	-1	-8	0	1	19	-18	0	2
19	372	316	137	-8	16	-20	2	22	-24	40	2	2	-4	0	0	22	-24	40	2
20	310	357	90	0	-3	0	0	0	7	0	1	0	-3	0	0	0	7	0	1

21	312	280	209	2	-10	0	1	2	0	0	0	2	-10	0	1	2	0	-5	0
22	149	346	55	-1	6	0	1	-1	6	5	1	-1	6	0	1	-1	6	5	1
23	50	316	336	0	6	0	1	0	6	5	1	0	6	0	1	0	6	0	1
24	65	181	270	-5	1	0	0	-5	1	0	1	-5	1	0	0	-5	1	0	1
25	141	146	342	1	-4	0	0	1	-4	0	0	1	-4	0	0	1	-4	-5	0
26	162	246	152	12	-4	-10	1	-8	-4	-10	1	2	-4	0	0	-8	-4	0	1
27	219	252	302	-1	2	0	0	-1	2	0	0	-1	2	0	0	-1	-8	-5	1
28	306	209	335	-4	-1	-10	0	6	9	10	1	6	-1	0	1	6	9	10	1
29	285	159	13	5	-1	5	1	5	-1	5	1	5	-1	5	1	5	-1	5	1
30	192	195	53	2	5	0	1	2	5	5	1	2	5	0	1	2	-5	10	1
31	207	217	186	-3	-3	5	0	-3	7	5	1	-3	-3	5	0	-3	7	5	1
32	213	121	208	-7	1	10	1	-7	-9	10	1	-37	-9	25	4	3	-9	-5	1
33	229	221	251	-1	-19	0	2	-1	-9	10	1	-1	-19	0	2	-1	-9	10	1
34	168	263	330	8	3	5	1	-2	3	5	0	8	3	0	1	-2	3	5	0

	Simulation				Réal				Simulation				Réal			
	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c
Moyenne (des valeurs absolues)	2,94	3,71	2,65	0,50	5,59	7,82	4,85	0,94	3,41	3,59	1,62	0,53	5,76	8,00	6,03	1,06
Ecart type (des valeurs absolues)	3,18	4,34	4,41	0,61	12,9	11,7	7,52	1,33	6,38	3,87	4,49	0,81	14,5	10,1	9,91	1,49

Taux de réussite	85%	65%	94%	68%
------------------	-----	-----	-----	-----

Le tableau ci-dessous présente l'apport des modifications réalisés sur la grille locale à savoir la superposition des cônes, la discrétisation des arcs par un modèle 8-connexe et la rotation analogique.

	Correction de la grille locale							
	Simulation				Réal			
	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c
Moyenne (des valeurs absolues)	2,1	2,6	1,2	0,26	7,6	6,1	5,6	1,2
Ecart type (des valeurs absolues)	2,1	3,4	2,4	0,50	19,2	9,1	8,1	1,94

Taux de réussite	94%	68%
------------------	-----	-----

ANNEXE 3 : Essais relatifs à la grille globale

Le tableau ci-dessous présente les moyennes des erreurs des résultats des 100 essais réalisés dans les trois environnements présentés à la figure III-20, avant puis après correction de la grille globale.

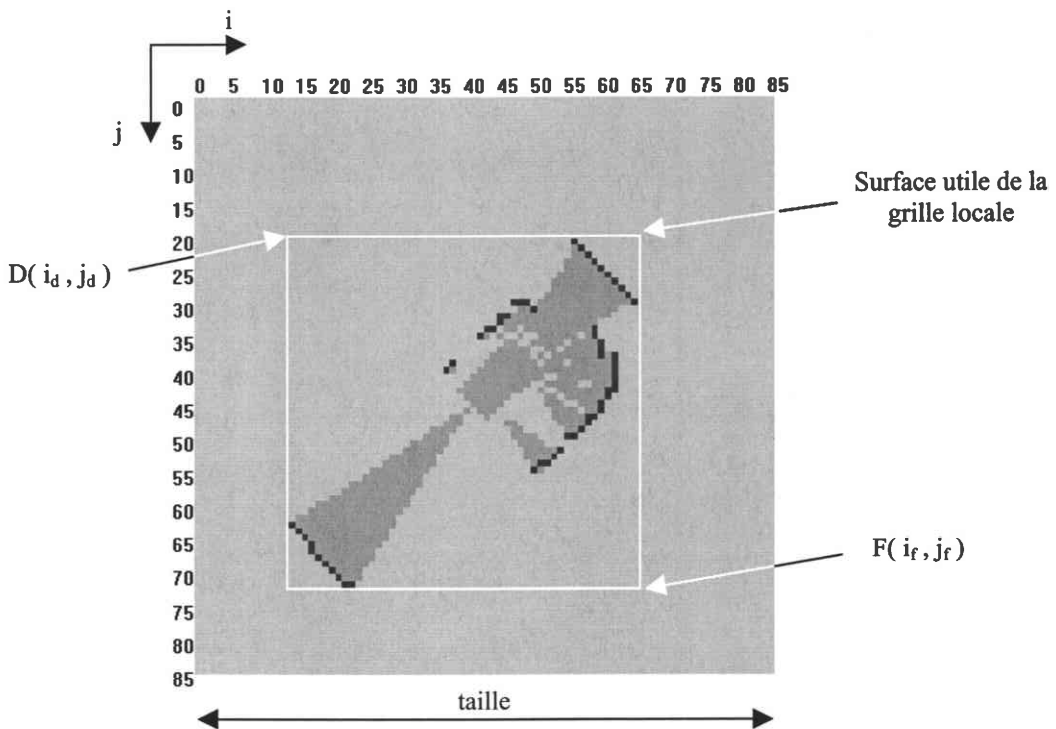
	Avant correction de la grille globale								Après correction de la grille globale							
	Simulation				Réel				Simulation				Réel			
	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c
Moyenne (des valeurs absolues)	4,7	6,4	2,3	0,9	13	12	8,9	1,9	2,9	2,9	0,65	0,39	10	11	7,9	1,5
Ecart type (des valeurs absolues)	5,9	9,4	3,3	1,0	20	19	12	2	1,9	2,0	1,9	0,18	19	19	11	2
Taux de réussite	75%				54%				99%				66%			

Le tableau suivant montre les moyennes des erreurs des résultats des 100 essais réalisés dans l'appartement simulé.

	Simulation				Réel			
	ϵ_x	ϵ_y	ϵ_θ	ϵ_c	ϵ_x	ϵ_y	ϵ_θ	ϵ_c
Moyenne (des valeurs absolues)	2,1	1,9	2,2	0,30	6,5	5,0	5,7	0,9
Ecart type (des valeurs absolues)	2,6	2,8	2,0	0,31	12	10	7,7	1,3
Taux de réussite	100%				80%			

ANNEXE 4 : Algorithme permettant d'obtenir la grille locale utile

En reprenant l'exemple de la figure IV-15, nous recherchons les coordonnées i_d , j_d , i_f , et j_f délimitant la surface utile.



Recherche de i_d :

```

Debut
   $i \leftarrow 0 ; j \leftarrow 0 ; \text{Test} \leftarrow \text{Faux}$ 
  Faire
    Faire
      Si Grille[i][j]  $\neq 0$  Alors
         $i_d \leftarrow i ; \text{Test} \leftarrow \text{Vrai}$ 
      Sinon  $j \leftarrow j+1$ 
    Tant que ( (  $j < \text{taille}$  ) Et (  $\text{Test} = \text{Faux}$  ) )
       $i \leftarrow i+1$ 
    Tant que ( (  $i < \text{taille}$  ) Et (  $\text{Test} = \text{Faux}$  ) )
Fin
  
```

Recherche de j_d :

De même que pour i_d en permutant les i et les j

Recherche de i_f :

```

Debut
   $i \leftarrow \text{taille} ; j \leftarrow \text{taille} ; \text{Test} \leftarrow \text{Faux}$ 
  Faire
    Faire
      Si Grille[i][j]  $\neq 0$  Alors
         $i_f \leftarrow i ; \text{Test} \leftarrow \text{Vrai}$ 
      Sinon  $j \leftarrow j - 1$ 
    Tant que ( (  $j \geq 0$  ) Et (  $\text{Test} = \text{Faux}$  ) )
       $i \leftarrow i - 1$ 
    Tant que ( (  $i \geq 0$  ) Et (  $\text{Test} = \text{Faux}$  ) )
Fin

```

Recherche de j_f :

De même que pour i_f en permutant les i et les j

Nouvelle Grille :

```

Debut
   $i \leftarrow 0 ; j \leftarrow 0$ 
  Nouvelle taille  $i \leftarrow i_f - i_d + 1$ 
  Nouvelle taille  $j \leftarrow j_f - j_d + 1$ 
  Faire
    Faire
      Nouvelle Grille[i][j] = Grille[i +  $i_d$ ][j +  $j_d$ ]
       $i \leftarrow i + 1$ 
    Tant que (  $i < \text{Nouvelle taille } i$  )
       $j \leftarrow j + 1$ 
    Tant que (  $j < \text{Nouvelle taille } j$  )
Fin

```

ANNEXE 5 : Extraction de la partie visible de la grille globale

globale

Le but est d'extraire de la grille globale la partie visible depuis la zone de recherche. Pour chacune des quatre extrémités de la zone de recherche nous réalisons un lancer de rayon dans toutes les directions (Figure A5-1). Tant qu'un rayon traverse des cellules libres, le contenu est gardé. La Figure A5-2 présente le résultat que nous devrions avoir à partir de l'exemple ci-dessous.

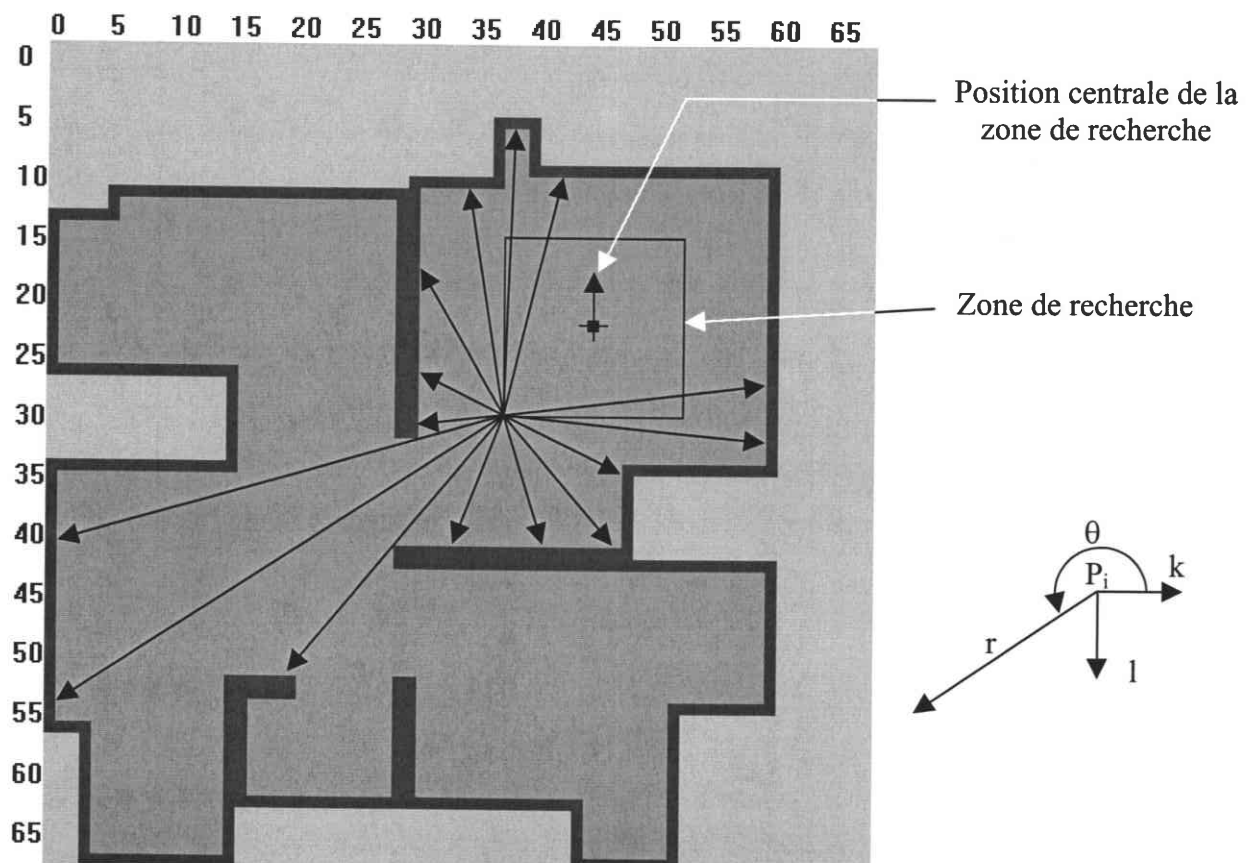


Figure A5-1 : Lancer de rayon pour une position

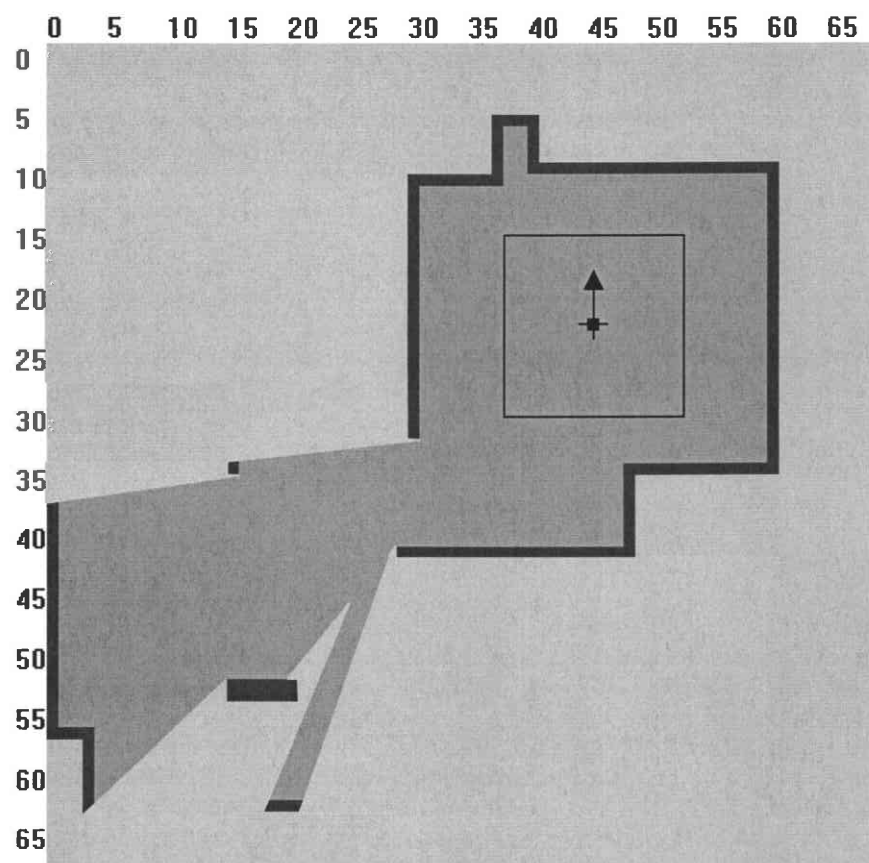


Figure A5-2 : Résultat devant être obtenu.

Algorithme :

Soient P_i les quatre points extrêmes de la zone de recherche.

Debut

$i \leftarrow 0$

Faire

$\theta \leftarrow 0$ //angle

Faire

$r \leftarrow 0$ //rayon

Test \leftarrow Faux

Faire

$(k, l) \leftarrow \text{Cellule}(P_i, r, \theta)$

Si ($\text{Grille}[k][l] = 1$) Alors Test = Vrai

Nouvelle $\text{Grille}[k][l] = \text{Grille}[k][l]$

$r \leftarrow r + 1$

Tant que (Test = Faux)

$\theta \leftarrow \theta + 1$

Tant que ($\theta < 360$)

$i \leftarrow i + 1$

Tant que ($i < 4$)

Fin

La fonction Cellule détermine la cellule d'indice (k, l) de la grille globale contenant le point de coordonnées polaire (r, θ) dans le repère de centre P_i .

REFERENCES
BIBLIOGRAPHIQUES

REFERENCES BIBLIOGRAPHIQUES

- [AKB96] **H. AKBARALLY, L. KLEEMAN** : « 3D Robot sensing from sonar and vision », Proceedings of the IEEE International Conference on Robotics and Automation, 1996, pp.686-691.
- [ARN93] **P. ARNOULD** : « Etude de la Localisation d'un Robot Mobile par Fusion de Données », Thèse de l'université de NANCY, INPL, septembre 1993.
- [ARR99] **K.O. ARRAS, N. TOMATIS** : « Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision », 3rd European Workshop on Advanced Mobile Robots (Eurobot'99), Zurich, Switzerland, september 6-8, 1999.
- [BEL94] **D.A. BELL, J. BORENSTEIN, S.P. LEVINE, Y. KOREN, L. JAROS** : « An Assistive Navigation System for Wheelchairs Based upon Mobile Robot Obstacle Avoidance », Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 2018-2022.
- [BEO95] **H.R. BEOM, H.S. CHO** : « Mobile robot localization using a single rotating sonar and two passive cylindrical beacons », Robotica, vol 13, 1995, pp. 243-252.
- [BOR91] **J. BORENSTEIN, Y. KOREN** « The Vector Field Histogram – Fast Obstacle-Avoidance for Mobile Robots », IEEE Transactions on Robotics and Automation, vol 7, n°3, June 1991, pp. 278-288.
- [BOR91a] **J. BORENSTEIN, Y. KOREN** « Histogramic in-motion mapping for mobile robot obstacle avoidance », IEEE Transactions on Robotics and Automation, vol 7, n°4, August 1991, pp. 535-539.

-
- [BOR96] **J. BORENSTEIN, L. FENG** : « Gyrodometry : a new method for combining data from gyros and odometry in mobile robots », Proceedings of the IEEE International Conference on Robotics and Automation, 1996, pp. 423-428.
- [BOR97] **J. BORENSTEIN, J. EVANS** : « The OmniMate Mobile Robot – Design, Implementation, and Experimental Results », Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 3505-3510.
- [BOU93] **G. BOURHIS, K. MOUMEN, P. PINO, S. ROHMER, A. PRUSKI** : « Assisted Navigation for a Powered Wheelchair », Proceedings of the IEEE International Conference on *Systems, Man, and Cybernetics*, 1993, pp. 553-558.
- [BRA95] **E. BRASSART** : « Localisation absolue d'un robot mobile autonome par des balises actives et un système de vision monoculaire », Thèse de l'université de Compiègne, janvier 1995.
- [BUL96] **H. BULATA** : « Modélisation d'un environnement structuré et localisation sur amers pour la navigation d'un robot mobile autonome », Thèse de l'université Paul Sabatier de Toulouse, mai 1996.
- [BYR92] **R.H. BYRNE, P.R. KLARER, J.B. PLETTA** : « Techniques for Autonomous Navigation » Sandia Report SAND92-0457, Sandia National Laboratories, Albuquerque, NM, March 1992.
- [CAS97] **J.A. CASTELLANOS, J.D. TARDOS, G. SCHMIDT** : « Building a Global Map of the Environment of a Mobile Robot : The Importance of Correlations », Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 1053-1059.
-

-
- [CHA91] **J-M. CHASSERY, A. MONTANVERT** : « Géométrie discrète en analyse d'images », *Traité des Nouvelles Technologies, Série Images*, HERMES, 1991.
- [CHO97a] **K.S. CHONG, L. KLEEMAN** : « Accurate Odometry and Error Modelling for a Mobile Robot », *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997, pp. 2783-2788.
- [COU94] **J.D. COURTNEY, A.K. JAIN** : « Mobile Robot Localization via Classification of Multisensor Maps », *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994, pp. 1672-1678.
- [COU98a] **A. COURCELLE, O. HORN, A. PRUSKI** : « Autonomous Mobile Robot Localisation Using Sonar », *IMACS Multiconference on Computational Engineering in Systems Applications*, Nabeul-Hammamet, Tunisie, Avril 1998, pp. 360-364.
- [COU98b] **A. COURCELLE, O. HORN** : « Ultrasonic data representation : application to mobile robots localisation », *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada, Octobre 1998, pp. 1559-1564.
- [COU99] **A. COURCELLE, O. HORN** : « Results on static localization for the V.A.H.M », *IMACS/IEEE International Multiconference on Circuits, Systems, Communications and Computers*, Athènes, Grèce, Juillet 1999.
- [COX 91] **I.J. COX** « Blanche-An experiment in Guidance and Navigation of an Autonomous Mobile Robot » *IEEE Transactions Robotics and Automation*, 7(3), 1991, pp. 193-204.
- [COZ97] **F. COZMAN, E. KROTKOV** : « Automatic Mountain Detection and Pose Estimation for Teleoperation of Lunar Rovers », *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997, pp. 2452-2457.
-

-
- [CRO89a] **J.L. CROWLEY** : « World modeling and position estimation for a mobile robot using ultrasonic ranging », Proceedings of the IEEE International Conference on Robotics and Automation, 1989.
- [DEL97] **L. DELAHOUCHE, C. PEGARD, B. MARHIC, P. VASSEUR** : « A Navigation System Based on an Omnidirectional Vision Sensor », Proceedings of the International Conference IROS'97, pp. 718-724.
- [DRU87] **M. DRUMHELLER** : « Mobile Robot Localization Using Sonar », IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 9, n°2, march 1987, pp. 325-332.
- [EIN97] **T. EINSELE** : « Real-Time Self-Localization in Unknown Indoor Environments using a Panorama Laser Range Finder », Proceedings of the International Conference IROS'97, pp. 697-702.
- [ELF87a] **A. ELFES** : « Sonar-Based Real-World Mapping and Navigation », IEEE Journal of Robotics and Automation, vol RA-3, n°3, June 1987, pp.233-249.
- [ELF87b] **A. ELFES, L. MATTHIES** : « Sensor Integration for Robot Navigation : Combining Sonar and Stereo Range Data in a Grid Based Representation », IEEE Conference on Decision and Control, Los Angeles CA, December 1987.
- [ELF89] **A. ELFES** : « Using Occupancy Grids for Mobile Robot Perception and Navigation », IEEE Computer, 1989, pp.46-57.
- [FER91] **A. FERRAND** : « Conception et mise en œuvre d'un système de capteurs proprioceptifs destiné à la localisation relative des robots mobiles », Thèse de l'université de Toulouse, LAAS, 1991.
- [FER97] **J. FERRUZ, A. OLLERO** : « Autonomous Mobile Robot Motion Control in Non-Structured Environments Based on Real-Time Video Processing », Proceedings of the International Conference IROS'97, pp. 725-731.
-

-
- [FUK96] **Y. FUKU, E. KROTKOV** : « Dead Reckoning for a lunar rover on uneven terrain », Proceedings of the IEEE International Conference on Robotics and Automation, 1996, pp. 411-416.
- [GON95] **J. GONZALEZ, A. STENTZ, A. OLLERO** : « A Mobile Robot Iconic Position Estimator Using a Radial Laser Scanner », Journal of Intelligent and Robotic Systems, vol 13, 1995, pp. 161-179.
- [GOU94] **C. GOURLEY, M. TRIVEDI** : « Sensor Based Obstacle Avoidance and Mapping for Fast Mobile Robots », Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 1306-1311.
- [HAN96] **U.D. HANEBECK, G. SCHMIDT** : « Set theoretic localization of fast mobile robots using an angle measurement technique », Proceedings of the IEEE International Conference on Robotics and Automation, 1996, pp.1387-1394.
- [HAG97] **G.D. HAGER, D. KRIEGMAN, E. YEH, C. RASMUSSEN** : « Image-based Prediction of Landmark Features for Mobile Robot Navigation », Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 1040-1046.
- [HAR87] **S.Y. HARMON** : « The ground surveillance Robot (GSR) : An Autonomous vehicle designed to Transit Unknown Terrain », IEEE Journal of Robotics and Automation, vol RA-3, June 1987, pp.266-279.
- [HAR98] **K.D. HARRIS, M. RECCE** : « Experimental modelling of time-of-flight sonar », Robotics and Autonomous Systems, vol 24, 1998, pp. 33-42.
- [HAS97] **M. HASHIMA, F. HASEGAWA, S. KANDA, T. MARUYAMA, T. UCHIYAMA** : « Localization and Obstacle Detection for a Robot for Carrying Food Trays », Proceedings of the International Conference IROS'97, pp. 345-351.
-

- [HON97a] **M.I. HONG, L. KLEEMAN** : « Ultrasonic classification and location of 3D room features using maximum likelihood estimation - Part I », *Robotica*, vol 15, 1997, pp. 483-491.
- [HON97b] **M.I. HONG, L. KLEEMAN** : « Ultrasonic classification and location of 3D room features using maximum likelihood estimation - Part II », *Robotica*, vol 15, 1997, pp. 645-652.
- [HOR94] **O. HORN, G. RIEMER** : « Dynamic Localisation Method for the V.A.H.M », SPRANN'94, IMACS-LILLE, France, avril 1994, pp. 512-517.
- [HOR97] **O. HORN, A. COURCELLE** : « Localisation statique d'un fauteuil électrique autonome », *RAIRO-APII-JESA*, Volume 31 n°5/1997, pp. 867-878.
- [HUA97] **J. HUANG, T. SUPAONGPRAPA, I. TERAURA, N. OHNISHI, N. SUGIE** : « Mobile Robot and Sound Localization », *Proceedings of the International Conference IROS'97*, pp. 683-689.
- [JOR95] **K-W. JORG** : « World modeling for an autonomous mobile robot using heterogenous sensor information », *Robotics and Autonomous Systems*, vol 14, 1995, pp. 159-170.
- [KAR97] **O. KARCH, H. NOLTEMEIER** : « Robot Localization – Theory and Practice », *Proceedings of the International Conference IROS'97*, pp. 850-856.
- [KIM94] **J.H. KIM, H.S. CHO** : « Experimental investigation for the estimation of a mobile robot's position by linear scanning of a landmark », *Robotics and Autonomous Systems*, vol 13, 1994, pp. 39-51.
- [KIM97] **K.KIMOTO, C. THORPE** : « Map Building with Radar and Motion Sensors for Automated Highway Vehicle Navigation », *Proceedings of the International Conference IROS'97*, pp. 1721-1728.

-
- [KO96] **J.H. KO, S.D. KIM, M.J. CHUNG** : « A method of indoor mobile robot navigation using acoustic landmarks », Proceedings of the IEEE International Conference on Robotics and Automation, 1996, pp.1726-1731.
- [KRE99] **M. KREUTNER** : « Localisation du VAHM : Estimation de l'orientation d'un fauteuil électrique autonome par vision monoculaire », Rapport de recherche de D.E.A d'automatique et traitement numérique du signal, septembre 1999.
- [KWO97] **Y.D. KWON, J.S. LEE** : « A Stochastic Environment Modelling Method for Mobile Robot by using 2-D Laser scanner », Proceedings of the International Conference IROS'97, pp. 1688-1693.
- [LAN92] **D. LANGER, C. THORPE** : « Sonar Based Outdoor Vehicle Navigation and Collision Avoidance », IROS'92.
- [LAW95] **G. LAWITZKY, W. FEITEN, M. MOLLER** : « Sonar sensing for low-cost indoor mobility », Robotics and Autonomous Systems, vol 14, 1995, pp. 149-157.
- [LEO92] **J. LEONARD, H. DURRANT-WHYTE, I.J. COX** : « Dynamic Map Building for an Autonomous Mobile Robot », The International Journal of Robotics Research, vol 11, n° 4, August 1992, pp. 286-298.
- [LI98] **S. LI, A. HAYASHI** : « Robot Navigation in Outdoor Environments by Using GPS Information and Panoramic Views », Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'98, Victoria, Canada, October 13-17, 1998, pp. 570-575.
- [MAC94] **P. MACKENZIE, G. DUDEK** : « Precise Positioning Using Model-Based Maps », Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 1615-1621.
-

-
- [MAD98] **C.B. MADSEN, C.S. ANDERSEN** : « Optimal landmark selection for triangulation of robot position », *Robotics and Autonomous Systems*, vol 23, 1998, pp. 277-292.
- [MAR99] **B. MARHIC, E.M. MOUADDIB, C. PEGARD, N. HUTIN** : « Localisation Based on Invariant-Models Recognition by SYCLOP », *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'99*, 1999, pp. 793-798.
- [MEN95] **P. MENEZES, J. DIAS, H. ARAUJO, A.D. ALMEIDA** : « Using Sonar Sensors to Perform Obstacle Detection », *Revue d'Automatique et de Productique appliquées*, 1995.
- [MIY97] **N. MIYAKE, T. AONO** : « Position Estimation and Path Control of an Autonomous Land Vehicle », *Proceedings of the International Conference IROS'97*, pp. 690-696.
- [MOR85] **H.P. MORAVEC, A. ELFES** : « High Resolution Maps from Wide Angle Sonar », *Proceedings of the IEEE Conference on Robotics and Automation*, Washington, D.C, 1985, pp. 116-121.
- [MOU89] **P. MOUTARLIER, R. CHATILA** : « An Experimental System for Incremental Environment Modelling by an Autonomous Mobile Robot », *Experimental Robotics I – Springer Verlag*, vol 139, 1989, pp. 327-346.
- [NEI96] **S. NEIRA, J. HORN, J.D. TARDOS, G. SCHMIDT** : « Multisensor mobile robot localization », *Proceedings of the IEEE Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996, pp. 673-679.
- [OHY94] **A. OHYA, Y. NAGASHIMA, S. YUTA** : « Exploring Unknown Environment and Map Construction Using Ultrasonic Sensing of Normal Direction of Walls », *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994, pp. 485-492.
-

-
- [OHY97] **A. OHYA, A. KOSAKA, A. KAK** : « Vision-Based Navigation of Mobile Robot with Obstacle Avoidance by Single Camera Vision and Ultrasonic Sensing », Proceedings of the International Conference IROS'97, pp. 704-711.
- [OUS97] **M. OUSSALAH, H. MAAREF, C. BARRET** : « Positioning of a Mobile Robot With Landmark-Based Method », Proceedings of the International Conference IROS'97, pp. 865-871.
- [PAG98] **D. PAGAC, E. NEBOT, H. DURRANT-WHYTE** : « An evidential approach to Map-Building for Autonomous Vehicles », IEEE Transactions on Robotics and Automation, vol 14, n°4, August 1998, pp. 623-629.
- [PAN98] **T. PANNEREC, M. OUSSALAH, H. MAAREF, C. BARRET** : « Absolute Localisation of a Miniature Mobile Robot Using Heterogeneous Sensors : Comparison Between Kalman Filter and Possibility Theory Method », IMACS Multiconference on Computational Engineering in Systems Applications, Nabeul-Hammamet, Tunisie, April 1-4, 1998, pp. 265-267.
- [PET97] **B. PETERSON, D. BRUCKNER, S. HEYE** : « Measuring GPS Signals Indoors », Proceedings of The 10TH International Technical Meeting of The Satellite Division of the Institute of Navigation, Kansas City, Missouri, September 16-19, 1997, pp. 615-624.
- [PIN93] **F.G. PIN, Y. WATANABE** : « Using Fuzzy Behaviors for the Outdoor Navigation of a Car with Low-Resolution Sensors », Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, Georgia, May 2-7, 1993, pp. 548-553.
- [POL91] **POLAROID 1991** : « Ultrasonic Ranging System », Product Literature, Polaroid Corporation, 784 Memorial Drive, Cambridge, MA02139, 617-386-3964.
-

-
- [SAB94] **A.M. SABATINI, O.D. BENEDETTO** : « Towards a Robust Methodology for Mobile Robot Localisation Using Sonar », Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 3142-3147.
- [SAT97] **M. SATO, J.K. AGGARWAL** : « Estimation of position and orientation from Image Sequence of a Circle », Proceedings of the IEEE International Conference on Robotics and Automation, New Mexico, 1997, pp. 2252-2257.
- [SHA76] **G. SHAFER** : « A Mathematical Theory of Evidence », Princeton, NJ : Princeton University Press, 1976.
- [SOI97] **M. SOIKA** : « Grid Based Fault Detection and Calibration of Sensors on Mobile Robots », Proceedings of the IEEE International Conference on Robotics and Automation, New Mexico, April 1997, pp.2589-2594.
- [TRA97] **P.E. TRAHANIAS, M.I.A. LOURAKIS, A.A. ARGYROS, S.C. ORPHANOUDAKIS** : « Navigational Support for Robotic Wheelchair Platforms : An Approach that Combines Vision and Range Sensors », Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 1265-1270.
- [UCH98] **S. UCHIDA, S. MAEYAMA, A. OHYA, S. YUTA** : « Position Correction Using Elevation Map for Mobile Robot on Rough Terrain », Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'98, Victoria, Canada, October 13-17, 1998, pp. 582-587.
- [WEC98] **P. WECKESSER, R. DILLMANN** : « Modeling unknown environments with a mobile robot », Robotics and Autonomous Systems, vol 23, 1998, pp. 293-300.
- [YAM96] **B. YAMAUCHI** : « Mobile Robot Localization in Dynamic Environments Using Dead Reckoning and Evidence Grids », Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April 1996, pp. 1401-1406.
-



UNIVERSITE DE METZ

Le Président de l'Université de METZ

VU l'arrêté du 30 mars 1992 relatif aux études de troisième cycle,

VU l'arrêté du Président de l'Université de METZ en date du 16 novembre 1999 nommant les rapporteurs chargés d'examiner les travaux de Monsieur COURCELLE Alain,

VU les rapports de Madame ROMBAUT et de Monsieur PRAJOUX, Rapporteurs,

ARRETE

ARTICLE 1 : Monsieur COURCELLE Alain est autorisé à soutenir une thèse de Doctorat de l'Université de METZ (UFR MIM) intitulée :

« Localisation d'un robot mobile : Application à l'aide à la mobilité des personnes handicapées moteur »

ARTICLE 2 : La composition du jury de soutenance de thèse de Monsieur COURCELLE Alain est fixée comme suit :

Madame HORN Odile,
Maître de Conférences à l'Université de METZ, LASC,

Monsieur MARTINET Philippe,
Habilité à diriger des Recherches, Maître de Conférences à l'Université de CLERMONT-FERRAND, LASMEA-CUST,

Monsieur PRAJOUX Roland, Rapporteur,
Directeur de Recherche CNRS, LAAS-CNRS de TOULOUSE,

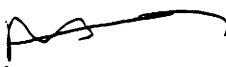
Monsieur PRUSKI Alain, Directeur de Thèse,
Professeur à l'Université de METZ, LASC,

Monsieur REDARCE Tanneguy,
Professeur à l'Institut National des Sciences Appliquées de LYON, LAI,

Madame ROMBAUT Michèle, Rapporteur,
Professeur à l'UTUT de TROYES, LM2S, Département GEII.

ARTICLE 3 : La soutenance aura lieu le **vendredi 14 janvier 2000 à 11 h** à l'Institut Supérieur de Génie Mécanique et Productique de METZ - Salle des Assemblées de l'Université -.

Fait à Metz, le 4 janvier 2000
Le Président de l'Université de METZ


Marie-Jeanne PHILIPPE



Localisation d'un robot mobile : application à l'aide à la mobilité des personnes handicapées moteur

Mots clés : Localisation, Capteurs à ultrasons, Robot mobile, Grilles d'occupation.

Résumé : Notre étude se situe dans l'aide à la mobilité des personnes handicapées moteur, dans le cadre du projet VAHM (Véhicule Autonome pour Handicapés Moteur). Ce projet se propose de développer un fauteuil roulant pour personne handicapée permettant d'en alléger la conduite. Comme pour toute application de robotique mobile, le fauteuil doit être capable de déterminer automatiquement sa position et son orientation dans son environnement. Cette information est notamment nécessaire avant d'exécuter un déplacement pour planifier la trajectoire à faire suivre au fauteuil. Notre contribution dans le projet consiste à établir le mode de localisation statique du fauteuil à l'aide des mesures d'une ceinture de 14 capteurs à ultrasons et d'une information imprécise sur la position du robot.

Pour ce faire, nous avons établi un modèle du capteur à ultrasons basé sur les grilles d'occupation permettant d'avoir pour chacun d'eux, une contribution équivalente pour la localisation quelle que soit la distance détectée. Ce modèle a ensuite été adapté pour l'utilisation d'une ceinture de 14 capteurs en tenant compte des superpositions de plusieurs mesures et des différentes orientations de chaque capteur. La grille ainsi obtenue représente l'espace perceptible par le robot depuis sa position. L'environnement d'évolution du robot quant à lui est représenté par une grille décrivant les limites de l'espace libre. Les deux grilles sont mises en correspondance pour déterminer la position et l'orientation du robot dans son environnement. La recherche de la meilleure correspondance entre les deux grilles est très coûteuse en temps de calcul. C'est pourquoi, une méthode de recherche hiérarchisée a été mise en œuvre pour rendre la localisation réalisable en un temps relativement court.

La méthode développée a fait l'objet d'une validation en conditions réelles dans un appartement simulé et les résultats obtenus sont conformes aux attentes du projet.

Mobile robot localization : application for a powered wheelchair

Keywords : Localization, Ultrasonic sensors, Mobile robot, Occupancy grids.

Abstract : Our study situates in the mobility aid for people with disabilities, as part of the VAHM project (French acronym for Autonomous Vehicle for People with Disabilities). This project proposes to develop a wheelchair for people with disabilities allowing autonomous displacements. As for mobile robotics applications, the wheelchair has to determine automatically its position and its orientation in its environment. This information is particularly necessary before executing a navigation in order to realize the path planning. Our contribution in the project consists in establishing the static localization mode of the wheelchair thanks to the measures of a belt of 14 ultrasonic sensors and an imprecise information on the robot's position.

To do this, we have set up an ultrasonic sensor model based on occupancy grids allowing to have for each of them, an equal contribution for the localization whatever the detected distance. Then, this model has been adapted for the use of a belt of 14 sensors by taking account the superposition of several measures and the different orientations of each sensor. Thus, the obtained grid represents the environment of the robot which is directly perceivable from its position. Moreover, the robot's evolution environment is represented in a grid describing limits of the free space. The two grids are compared to determine the robot's position and orientation in its environment. The research of the best matching between the two grids is very expensive in time of calculation. This is why, a graded research method has been achieved to give the localization's result in a relatively short time.

The developed method has been validated in real conditions in a simulated apartment and obtained results are in accordance with the project.