



HAL
open science

Localisation Et Guidage Du Robot Mobile Atrv2 Dans Un Environnement Naturel.

A. Oualid Djekoune

► **To cite this version:**

A. Oualid Djekoune. Localisation Et Guidage Du Robot Mobile Atrv2 Dans Un Environnement Naturel.. Robotique [cs.RO]. Université des Sciences et de la Technologie Houari Boumediene, 2010. Français. NNT : 06/2010-D/EL . tel-00673213

HAL Id: tel-00673213

<https://theses.hal.science/tel-00673213>

Submitted on 23 Feb 2012

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.

N° d'ordre : 06/2010-D/EL

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE

Faculté d' Electronique et d'Informatique



Thèse

Présentée pour l'obtention du grade de DOCTORAT

En: ELECTRONIQUE

Spécialité: Contrôle de Processus et Robotique

Par: A. Oualid DJEKOUNE

Sujet

Localisation Et Guidage Du Robot
Mobile Atrv2 Dans Un Environnement
Naturel.

Soutenue publiquement le 15 décembre 2010, devant le jury composé de:

Mr. ATTARI Mokhtar	Professeur à l'USTHB	Président
Mr. ACHOUR Karim	Directeur de Recherche au CDTA	Directeur de thèse
Mr. TOUMI Redouane	Professeur à l'USTHB	Codirecteur de thèse
Mr. HAMERLAIN Mustapha	Directeur de Recherche au CDTA	Examineur
Mr. BELHOCINE Mahmoud	Maître de Recherche A au CDTA	Examineur
Mme. ACHOUR Nora	Maître de conférence A à l' USTHB	Examinatrice

à mes parents,
à ma femme,
à mes fils Imad Eddine et Mohamed Wassim,
à mes filles Lyna et Farah,
à mes frères et sœurs.

Avant propos

Les travaux présentés dans cette thèse ont été effectués au sein de l'équipe de recherche "Vision Artificielle et ANalyse d'IMages" du Centre de Développement des Technologies Avancées (CDTA).

Honneur à ceux qui sont à l'origine de ce travail, je parle bien évidemment de mes directeurs de thèse Messieurs Achour Karim et Toumi Redouane. Sans eux, sans leur confiance qu'ils m'ont accordée, sans leur aide, sans leurs conseils, rien, absolument rien de cette extraordinaire aventure n'aurait existé.

J'exprime toute ma gratitude à Azouaoui Wahiba, Toumi Redouane, Belhocine Mahmoud, Hamerlain Mustapha et Bouzouia Brahim, qui n'ont compté ni leurs efforts ni leur temps pour lire ce rapport. Leurs critiques constructives et leurs questions pertinentes m'ont été d'un grand bénéfice.

J'exprime ma sincère reconnaissance à Monsieur Attari Mokhtar, professeur à l'*USTHB* pour l'honneur qu'il m'a fait en acceptant de juger ce travail et de présider le jury de soutenance.

Je tiens à remercier Madame Achour Nora, maître de conférences A à l'*USTHB* et messieurs Hamerlain Mustapha, directeur de recherche et directeur de la division "Productique et Robotique" et Belhocine Mahmoud, maître de recherche A, chef d'équipe "Vision Artificielle et ANalyse d'IMages", qui m'ont fait l'honneur de juger ce travail en participant en tant que membres de jury.

Mes remerciements vont également à tous ceux qui ont permis de loin ou de près, de contribuer à l'aboutissement de ce travail et à tous les membres de la division "Productique et Robotique" pour l'ambiance de travail.

Un remerciement particulier pour mes parents, mon épouse et mes enfants Imad Eddine, Lyna et Farah, ainsi qu'à toute ma famille pour m'avoir donné la force d'aller au bout de ce travail, pour leur patience et leur amour.

Résumé

Les travaux présentés dans cette thèse portent essentiellement sur la localisation et la navigation du système mobile *ATRV2* dans un environnement inconnu a priori. Ce système doit d'abord modéliser et construire sa carte de l'environnement à partir de données issues de ses capteurs embarqués, se localiser dans cette carte et ensuite générer un chemin sans collisions lui permettant d'atteindre son but.

Le modèle de l'environnement que nous avons choisi est un modèle métrique (grille de certitude) basé sur l'algorithme *HIMM* et est mis à jour en utilisant les données en provenance des capteurs embarqués.

La méthode de localisation utilisée est basée sur le filtre de *Kalman* étendu (EKF). Cette méthode fusionne les données odométriques (modèle du système) et les données du système de perception embarqué (mesure) pour corriger régulièrement l'estimation relative de la position du système mobile due à l'erreur cumulative engendrée par les capteurs odométriques. Pour déterminer la mesure, la méthode utilise la mise en correspondance des grilles tout en estimant sa zone de recherche et la position autour de laquelle elle effectue la recherche de la position estimée.

Pour la navigation, nous avons proposé une nouvelle méthode appelée *DVFF* combinant une méthode de planification de trajectoire globale basée sur l'algorithme D^* , et une méthode réactive non stop basée sur le principe des champs de forces virtuelles *VFF*. Cette méthode de navigation utilise les valeurs courantes des capteurs ultrasonores embarqués (sur une petite fenêtre temporelle), et les données provenant du modèle de l'environnement, pour décider de l'action à effectuer. Elle permet au système mobile *ATRV2* de se mouvoir sans crainte de collisions avec les obstacles de son environnement.

Abstract

The work presented in this thesis focus on the localization and the navigation of the ATRV2 mobile system in an unknown environment. This mobile system must first model and build his map using data from its onboard sensors, localize itself in this map, and then, generate a collision-free path allowing to it to reach his goal.

The model of the environment we have chosen is a metric model (certainty grid) based on the *HIMM* algorithm, and is updated using data from the onboard sensors.

The used localization method is based on the extended *Kalman* filter (EKF). This method merges the odometric data (system model) and the data from the onboard perception system (measure) to regularly correct the relative estimate of the mobile system position due to the cumulative error caused by the odometer sensors. To determine the measure, the method uses the map matching while estimating its search area and the position around which it searches the estimated position.

For the navigation, we have proposed a new method called *DVFF* combining path planning method based on the D^* algorithm, and reactive non-stop method based on the virtual force field *VFF*. This navigation method uses the current values of onboard ultrasonic sensors (on a small time window), and data from the environment model, to decide what action to take. It allows to the ATRV2 mobile system to move without collisions with obstacles of its environment.

Table des matières

Avant propos	iii
Résumé	iv
Abstract	v
Table des matières	vi
Table des figures	ix
Liste des tableaux	xii
Présentation générale.	13
I. Contexte général.	14
II. Problématique traitée.	15
III. Motivation et contributions.	15
IV. Plan de la thèse.	16
Partie I : Introduction aux systèmes de robotique mobile.	18
Chapitre I : Système de perception des systèmes de robotique mobile.	19
I.1. Capteurs utilisés dans les systèmes de robotique mobiles.	20
I.1.1. Capteurs proprioceptifs.	20
I.1.1.1. Capteurs de déplacement.	21
I.1.1.2. Capteurs d'attitude.	24
I.1.2. Capteurs extéroceptifs.	25
I.1.2.1. Les capteurs télémétriques.	25
I.1.2.2. Les capteurs à micro-onde.	28
I.1.2.3. Les capteurs tactiles.	28
I.1.2.4. Les GPS.	28
I.1.2.5. Les caméras.	29
a. La stéréovision.	30
b. Les techniques de lumières structurées.	41
I.1.3. Conclusion.	42
I.2. Fusion de données capteurs.	42
I.2.1. Méthodes classiques.	43
I.2.1.1. Raisonnement Bayésien.	43
I.2.1.2. Utilisation d'estimateurs.	44
I.2.2. Méthodes liées à un traitement non probabiliste de l'incertain.	45
I.2.2.1. Combinaison par la théorie des possibilités.	45
I.2.2.2. Utilisation de la théorie de l'évidence.	46
I.2.3. Méthodes dérivées de l'intelligence artificielle.	47
I.2.4. Méthodes d'optimisation.	48
I.2.5. Conclusion.	48
I.3. Représentation d'un environnement de navigation.	48
I.3.1. Représentation métrique.	49
I.3.1.1. Représentation géométrique.	49
I.3.1.2. Grille d'occupation.	50
I.3.2. Représentation topologique.	51
I.3.3. Conclusion.	53
I.4. Conclusion.	53
Chapitre II : Localisation et navigation des systèmes de robotique mobile.	55
II.1. Localisation des systèmes de robotique mobile.	56
II.1.1. La localisation relative.	56

II.1.1.1. L'odométrie.	57
II.1.1.2. La vélocimétrie.	57
II.1.1.3. Les capteurs inertiels.	57
II.1.1.4. Les capteurs d'images.	57
II.1.2. La localisation absolue.	58
II.1.2.1. Nature des balises.	58
II.1.2.2. Principes de la localisation.	61
II.1.3. Méthodes hybrides.	64
II.1.3.1. Le recalage dynamique.	65
II.1.3.2. La coopération multi-capteurs.	66
II.1.4. Localisation et modélisation simultanées de l'environnement (<i>SLAM</i>).	67
II.1.5. Conclusion.	68
II.2. Navigation des systèmes de robotique mobile.	68
II.2.1. Les difficultés de la navigation.	69
II.2.2. Les approches locales.	71
II.2.2.1. Les algorithmes Bug.	71
II.2.2.2. Les approches basées sur les champs de potentiel.	71
II.2.2.3. La méthode Vector Field Histogram (<i>VFH</i>).	73
II.2.2.4. Les approches d'optimisation de contraintes (Constrained optimization approaches).	74
II.2.2.5. La méthode Nearness Diagram ou <i>ND</i> .	76
II.2.2.6. La méthode Elastic Bande ou <i>EB</i> .	78
II.2.2.7. Autres méthodes.	78
II.2.3. Les approches globales.	79
II.2.3.1. Espace des configurations.	79
II.2.3.2. Discrétisation de l'environnement.	80
II.2.3.3. Planification d'un chemin (trajectoire) global.	81
II.2.3.4. Recherche d'un chemin optimal.	86
II.2.4. Approches Hybrides.	88
II.2.4.1. La méthode <i>Vector Field Histogram*</i> (<i>VFH*</i>).	88
II.2.4.2. La méthode <i>Global Dynamic Window Approach</i> (<i>GDWA</i>).	90
II.2.4.3. La méthode <i>Global Nearness Diagram</i> (<i>GND</i>).	91
II.2.4.4. Autres méthodes de planification.	91
II.2.5. Conclusion.	92
II.3. Conclusion.	92
Partie II : Localisation et navigation d'un système mobile : cas de l'ATRV2.	94
Chapitre III : Modèle de l'environnement de navigation utilisé.	95
III.1. Système de perception utilisé.	96
III.1.1. Capteurs à ultrasons.	96
III.1.2. Système de vision.	99
III.2. Représentation des mesures issues des capteurs embarqués.	100
III.2.1. Modèle de l'environnement.	100
III.2.1.1. La carte locale.	100
III.2.1.2. La carte globale.	100
III.2.2. Capteurs à ultrasons.	101
III.2.3. Système de vision.	105
III.2.3.1. Calcul des paramètres du modèle géométrique d'une Caméra : Méthode choisie.	105
III.2.3.2. Mise en œuvre du calibrage : Application au banc stéréoscopique du système mobile <i>ATRV2</i> .	108
III.2.4. Fusion des données capteurs.	108

III.2.4.1. Calibrage Capteurs à ultrasons-Banc stéréoscopique.	109
III.2.4.2. La fusion.	110
III.3. Expérimentations.	115
III.3.1. Construction basée sur les données capteurs ultrasonores.	115
III.3.2. Construction basée sur les données du banc stéréoscopique.	117
III. 4. Conclusion.	120
Chapitre IV : Localisation et navigation proposées pour le système mobile ATRV2.	122
IV.1. Le système mobile <i>ATRV2</i> .	123
IV.1.1. Dimensions.	125
IV.1.2. Cinématique.	125
IV.1.3. Holonomie et non Holonomie.	129
IV.2. Approche de localisation utilisée.	130
IV.2.1. Le modèle du système odométrique.	130
IV.2.2. Le modèle de mesure.	134
IV.2.3. Le modèle d'incertitude.	134
IV.2.4. La prédiction de la position du système mobile.	135
IV.2.5. La mesure.	135
IV.2.6. Estimation finale de la position du système mobile.	136
IV.3. Approche de navigation proposée.	137
IV.3.1. Algorithme de planification de chemin optimal D^* .	137
IV.3.1.1. Introduction.	137
IV.3.1.2. Algorithme développé.	137
IV.3.1.3. Résultats de simulation.	139
IV.3.1.4. Résultats expérimentaux.	141
IV.3.2. L'algorithme Champs de Forces Virtuelles ou <i>VFF</i> (Virtual Force Field).	145
IV.3.2.1. Introduction.	145
IV.3.2.2. Algorithme de navigation développé.	147
IV.3.2.4. Résultats expérimentaux.	148
IV.3.3. L'approche <i>DVFF</i> .	152
IV.3.3.1. Premier algorithme.	153
IV.3.3.2. Deuxième algorithme.	153
IV.3.3.3. Résultats expérimentaux.	154
IV.4. Contrôle de mouvement du système mobile <i>ATRV2</i> .	158
IV.5. Conclusion.	160
Conclusion générale et perspectives.	162
Bibliographie.	165
Annexes.	181

Table des figures

Figure I.1: Principe de l'odométrie.	22
Figure I.2: Principe du capteur à effet Doppler.	23
Figure I.3: Exemples de télémètres ultrasonores	26
Figure I.4: Exemple de ceinture de 12 capteurs à ultrasons.	26
Figure I.5: Exemple d'utilisation d'un capteur tri-aural.	26
Figure I.6: Télémètre infrarouge <i>Sharp</i> de la famille <i>GP2</i> .	27
Figure I.7: Télémètre laser <i>SICK</i> fournissant 180 mesures réparties sur 180 degrés.	27
Figure I.8: Géométrie du capteur stéréoscopique.	34
Figure I.9: Modèle sténopé d'une caméra.	36
Figure I.10: Faisceaux de lignes épipolaires: cas général non réctifié.	38
Figure I.11: Faisceaux de lignes épipolaires: cas rectifié.	38
Figure I.12: Triangulation par plusieurs plans Laser.	42
Figure I.13: Exemple d'opérateurs à comportement constant.	46
Figure I.14: Fusion en cas de conflit important.	46
Figure I.15: Création d'une RCD (Regions of Constant Depth).	50
Figure II.1: Estimation de la position d'un système mobile utilisant un repère artificiel.	60
Figure II.2: Localisation par mesure de distances de trois balises.	62
Figure II.3: Localisation par mesure de distances incertaines de trois balises.	62
Figure II.4: Localisation par triangulation.	62
Figure II.5: Un exemple de navigation utilisant un algorithme <i>Bug</i> .	71
Figure II.6: L'approche champ de potentiel.	72
Figure II.7: Grille d'occupation locale construite par la méthode <i>HIMM</i> .	73
Figure II.8: Calcul de la direction de déplacement basée sur l'histogramme des obstacles.	73
Figure II.9: Fenêtre Dynamique.	75
Figure II.10: Construction du diagramme <i>PND</i> selon un environnement donné.	76
Figure II.11: Construction du diagramme <i>RND</i> selon un environnement donné, un obstacle pénètre la zone de sécurité.	77
Figure II.12: Types d'actions menées dans les situations <i>LS1</i> et <i>LS2</i> .	77
Figure II.13: Différents types d'actions menées dans des situations dites de haute sécurité.	77
Figure II.14: Chapelet de bulles.	78
Figure II.15: Reconstruction d'un <i>C-Space</i> .	80
Figure II.16: Décompositions en cellules de l'espace libre dans les cartes métriques.	81
Figure II.17: Décompositions en chemins précalculés dans les cartes métriques.	81
Figure II.18: Diagramme de <i>Voronoi</i> .	82
Figure II.19: Exemple d'un graphe de visibilité.	83
Figure II.20: Méthode de <i>Brooks</i> .	83
Figure II.21: Exemple de la méthode des silhouettes.	84
Figure II.22: Exemples de décomposition exacte de C_{free} .	85
Figure II.23: Décomposition approchée de l'espace libre.	85
Figure II.24: approximation des trajectoires.	89
Figure II.25: chemins générés selon la profondeur de l'arbre.	90
Figure II.26: Arbres générés pour les chemins b), c) et d) de la figure II.25.	90
Figure II.27: Génération des distances au but par la fonction <i>NF1</i> .	91
Figure III.1: Deux obstacles renvoyant la même mesure.	97

Figure III.2: Répartition des énergies.	97
Figure III.3: Phénomène de spécularité.	98
Figure III.4: Exemple de réflexions multiples.	98
Figure III.5: Exemple de diaphonie.	98
Figure III.6: Le système mobile <i>ATRV2</i> et sa perception panoramique.	99
Figure III.7: Système de vision de l' <i>ATRV2</i> .	100
Figure III.8: Fonction de probabilité d'un capteur idéal.	101
Figure III.9: Grille pour un capteur idéal.	102
Figure III.10: Grille pour un capteur idéal prenant en compte l'espace libre.	102
Figure III.11: Exemple d'obstacle mal perçu.	102
Figure III.12: Grille prenant en compte l'ouverture angulaire du capteur.	102
Figure III.13: Profils de probabilité appliqués au capteur à ultrasons.	103
Figure III.14: Vue des profils de probabilité en 3D.	103
Figure III.15: Exemple de modélisation d'un capteur à ultrasons par une gaussienne.	104
Figure III.16: Modélisation du capteur à ultrasons.	104
Figure III.17: Mise en oeuvre du calibrage du banc stéréoscopique du système mobile <i>ATRV2</i> .	107
Figure III.18: Reconstruction 3D de la mire.	107
Figure III.19: Positions et orientations des capteurs US embarqués et le modèle géométrique du banc stéréoscopique.	110
Figure III.20: Le champ de vision commun entre les capteurs à ultrasons et le banc stéréoscopique.	111
Figure III.21: Exemple de fusion des données capteurs basée sur la combinaison adaptative.	112
Figure III.22: Le logiciel d'analyse et d'affichage développé.	115
Figure III.23: Situation réelle statique 1.	116
Figure III.24: Situation réelle statique 2.	116
Figure III.25: Situation réelle statique 3.	117
Figure III.26: Première situation de mouvement du système mobile.	117
Figure III.27: Deuxième situation de mouvement du système mobile.	117
Figure III.28: Troisième situation de mouvement du système mobile.	117
Figure III.29: La zone de recherche: cas général.	119
Figure III.30: Principe de la recherche directe.	119
Figure III.31: Résultats d'appariement et la reconstruction 3D en utilisant tous les points de la paire stéréo.	119
Figure III.32: Résultats d'appariement et la reconstruction 3D en utilisant les points contours de la paire stéréo.	120
Figure IV.1: Le système mobile <i>ATRV2</i> .	124
Figure IV.2: Vue de haut du système mobile <i>ATRV2</i> .	124
Figure IV.3: Vue arrière de l' <i>ATRV2</i> .	125
Figure IV.4: Vue de l'intérieur de l' <i>ATRV2</i> .	125
Figure IV.5: Géométrie de l' <i>ATRV2</i> .	125
Figure IV.6: Le repère de l' <i>ATRV2</i> dans le repère absolu R_u .	126
Figure IV.7: Le modèle cinématique utilisé pour l' <i>ATRV2</i> .	126
Figure IV.8: Espace de vitesses de l' <i>ATRV2</i> .	127
Figure IV.9: Quatre différents types de mouvement réalisables par l' <i>ATRV2</i> .	128
Figure IV.10: Déplacement de l' <i>ATRV2</i> dans le cas d'un virage à gauche.	129

Figure IV.11: Position de l'ATR _{V2} dans le repère univers R_u en fonction des données odométriques.	131
Figure IV.12: Déplacement circulaire entre deux instants d'échantillonnage.	132
Figure IV.13: L'algorithme du filtre de <i>Kalman</i> étendu utilisé.	136
Figure IV.14: L'attribut <i>wc</i> selon les dimensions du système mobile.	138
Figure IV.15: Calcul de la direction globale du chemin conformément à la position du <i>back pointer</i> .	138
Figure IV.16: Direction du chemin global à partir d'un point de départ et un point d'arrivée donnés.	138
Figure IV.17: Calcul de la direction du chemin global au point CP_1 .	138
Figure IV.18: L'interface du logiciel de recherche de chemin optimal développé basé sur l'algorithme D^* .	140
Figure IV.19: Chemin optimal.	141
Figure IV.20: Première expérimentation.	143
Figure IV.21: Deuxième expérimentation.	144
Figure IV.22: Troisième expérimentation.	145
Figure IV.23: Principe du Champ de Forces Virtuelles.	146
Figure IV.24: Calcul de la force répulsive frontale \vec{F}_F .	147
Figure IV.25: Calcul des forces latérales.	147
Figure IV.26: Décomposition des forces latérales.	148
Figure IV.27: Calcul du vecteur résultant \vec{R} .	148
Figure IV.28: Première expérience.	150
Figure IV.29: Deuxième expérience.	151
Figure IV.30: Troisième expérience.	152
Figure IV.31: Diagramme de l'approche de navigation $DVFF$.	153
Figure IV.32: Calcul du vecteur résultant \vec{F} .	153
Figure IV.33: Organigramme du deuxième algorithme.	154
Figure IV.34: Première expérience.	156
Figure IV.35: Deuxième expérience.	156
Figure IV.36: Troisième expérience.	157
Figure IV.36: Troisième expérience.	157
Figure IV.38: Cinquième expérience.	158
Figure IV.39: Modèle cinématique du système mobile ATR _{V2} et sa commande de mouvement.	160

Liste des tableaux

Tableau III.1 : Récapitulatif des états possibles des cellules appartenant au champ de vision commun pour la fusion.	112
Tableau III.2 : Résultats de la fusion des données capteurs basée sur la combinaison adaptative pour les grilles A et B.	114

Présentation Générale.

I. Contexte général.

La principale ligne actuelle de recherche pour la robotique mobile est de permettre à un système robotique mobile (système mobile ou robot mobile) de se déplacer de manière autonome dans son environnement pour accomplir un certain nombre de tâches. Ces tâches sont, par exemple, se déplacer vers une cible fixe ou mobile, éviter les obstacles, accoster, suivre une route, explorer et intervenir dans un milieu hostile. Pour cela, les problèmes à résoudre sont parfois assez complexes. L'un d'eux est la détermination d'une représentation interne de l'environnement (carte) du système mobile au moyen de la perception pour permettre la planification des actions et le contrôle de l'exécution. Le choix de la représentation est essentiel. Cette représentation doit être adaptée à la tâche du système mobile mais aussi à l'environnement dans lequel il évolue.

Le type d'environnement dans lequel devra se déplacer le système mobile détermine bien entendu le type de capteurs à utiliser pour la perception. De très nombreux capteurs sont utilisables : des caméras, des proximètres, et des capteurs tactiles, etc. Mais aucun d'entre eux ne peut à lui seul rendre compte de tout ce que le système mobile doit connaître sur le monde qui l'entoure. La perception de chacun d'eux est partielle et entachée d'erreurs. La solution consistera donc, à faire coopérer un grand nombre de capteurs. Cette coopération entre capteurs est très intéressante, voire indispensable pour la réalisation des objectifs de la perception. Elle peut se faire entre des capteurs de nature différente (vision et capteurs acoustiques, par exemple) permettant la perception de caractéristiques totalement différentes des objets, ou de même type, placés à différents endroits diminuant ainsi les problèmes de résolution, de champ de vue ou dus à l'occultation.

Un autre problème en relation avec la perception est celui de la localisation du système mobile dans la carte produite à partir des éléments perçus de l'environnement. Elle consiste à calculer et à maintenir à jour la connaissance de la position et de l'orientation du système mobile dans un repère absolu lié à son environnement. Ensuite, un itinéraire entre la position calculée du système mobile et son but, généré grâce à des outils de planification de chemins, peut enfin être exécuté, ce qui amène le système mobile à évoluer dans son environnement.

C'est dans ce contexte que se situent les travaux de notre thèse. Le but est de doter le système mobile *ATRV2*¹ d'une capacité de réaction et d'action dans un environnement naturel qui peut être inconnu a priori. En particulier, il sera nécessaire de modéliser et de construire la carte de l'environnement à partir de données issues des capteurs de son système de perception (une ceinture de douze capteurs à Ultrasons de type Polaroid et un banc stéréoscopique couleur) pour se localiser et ensuite calculer un itinéraire lui permettant d'atteindre son but.

¹Le système mobile, tout-terrain, *ATRV2* est une plate forme expérimentale construite par la société *iRobot*, équipé d'une ceinture de douze capteurs ultra sonores et d'un banc stéréoscopique avec deux caméras couleur.

II. Problématique traitée.

Pour un système mobile se déplaçant dans un environnement qui peut être inconnu a priori, l'autonomie de déplacement devient un problème encore plus difficile à résoudre que dans le cas où le système mobile évolue dans un environnement parfaitement connu. La navigation autonome doit pouvoir répondre aux trois mêmes questions que nous nous posons tous inconsciemment lorsque nous utilisons nos véhicules :

Où suis-je ?

Où dois-je aller ?

Comment m'y rendre ?

Les deux premières questions posent le problème de la localisation. Cette localisation nécessite le choix de capteurs afin que le système mobile puisse reconnaître son environnement. La reconnaissance implique la présence d'une connaissance initiale. Tout comme le conducteur humain qui utilise sa mémoire pour se localiser, un système mobile doit être pourvu d'une représentation interne de son environnement de navigation.

La troisième question impose une contrainte supplémentaire sur la représentation interne de l'environnement. Si la mémoire du système mobile doit lui permettre de se localiser, elle doit aussi lui donner la capacité de définir un plan de navigation pour atteindre son but. Cette aptitude passe par la mise en place d'une organisation adéquate de la mémoire du système mobile. Elle doit lui permettre de faire le lien entre sa position initiale et la position de son but.

Les réponses à ces trois questions suffisent aux conducteurs humains pour réaliser une tâche de navigation. Tout conducteur est censé posséder son permis de conduire ; le contrôle des mouvements du véhicule n'est donc pas, a priori, une difficulté. Cependant, donner aux systèmes mobiles la capacité de contrôler leurs mouvements reste un problème majeur de la recherche en robotique mobile. Une quatrième question doit donc être rajoutée pour pouvoir réaliser une tâche de navigation par un système mobile : "Comment réaliser ce déplacement ?".

De cette analyse, nous en déduisons qu'un système mobile autonome doit pour réaliser une tâche de navigation :

- posséder un (ou des) capteurs(s) proprioceptif(s) et extéroceptif(s) ;
- avoir une représentation de son environnement de navigation ;
- être capable de se localiser dans cet environnement ;
- posséder un outil de définition d'un plan de navigation ;
- être capable de contrôler ses mouvements pour réaliser ce plan.

Se sont de pré-requis que nous devons satisfaire durant notre travail pour définir une méthode de localisation et une méthode de navigation autonome robuste permettant au système mobile ATRV2 de se déplacer d'un point initial vers un point destination (ou but) tout en garantissant l'évitement des obstacles de son environnement.

III. Motivation et Contributions.

Les travaux présentés dans cette thèse portent essentiellement sur le développement d'une méthode de localisation et d'une méthode de navigation pour le système mobile ATRV2 dans un environnement naturel inconnu a priori.

Nous nous intéressons plus particulièrement au développement des méthodes de localisation et navigation beaucoup plus qu'à l'environnement où le système mobile *ATRV2* se déplace. L'environnement naturel nécessite un certain type de capteur (inclinomètre, GPS, etc.) que le système mobile *ATRV2* ne possède pas. Nous nous sommes restreints au déplacement de ce système mobile sur un sol plat et n'ayant aucune connaissance a priori sur son environnement.

Notre motivation première est le développement de techniques de perception, de localisation et de navigation appropriées rendant ce système mobile capable de percevoir et d'utiliser ses données perceptuelles pour augmenter son autonomie de déplacement, pouvoir agir et réagir face à son environnement et accomplir ses tâches tout en évitant les obstacles imprévus. Pour cela, le système mobile *ATRV2* doit d'abord modéliser et construire sa carte de l'environnement à partir de données issues de ses capteurs embarqués, se localiser dans cette carte et ensuite générer un chemin sans collisions lui permettant d'atteindre son but.

Nos contributions ont porté sur les points suivants:

- ⇒ Mise en œuvre d'une méthode de construction du modèle de l'environnement du système mobile *ATRV2*.
- ⇒ Réalisation d'une méthode de localisation du système mobile *ATRV2* dans son environnement combinant le filtre de *Kalman* étendu et la mise en correspondance de cartes
- ⇒ Etude et mise en œuvre d'une méthode de navigation locale basée sur le principe des champs de forces virtuelles *VFF*.
- ⇒ Etude et mise en œuvre d'une méthode de navigation globale basée sur l'algorithme *D** utilisant la contrainte géométrique du système mobile *ATRV2*.
- ⇒ Etude et mise en œuvre d'une nouvelle méthode de navigation hybride nommée *DVFF* combinant les méthodes de navigation *D** et *VFF*.
- ⇒ Mise en œuvre d'une méthode de contrôle du mouvement du système mobile *ATRV2*.

IV. Plan de la thèse.

Ce mémoire de thèse se divise en deux grandes parties contenant chacune deux chapitres. La première partie est un état de l'art concernant tous les points devant être abordés pour l'intégration d'un système de perception sur un système de robotique mobile. Les différents types de capteurs disponibles sur le marché et les approches traitant de la modélisation-fusion de leurs mesures sont présentés dans le premier chapitre ainsi que les différentes représentations d'environnements proposées dans la littérature. Le deuxième chapitre passe en revue les principaux travaux menés dans le cadre des différents systèmes de localisation et de navigation et les méthodes développées pour leur résolution.

La deuxième partie présente notre contribution concernant la localisation et la navigation du système mobile *ATRV2* dans un environnement qui peut être inconnu a priori, où nous commençons d'abord dans le troisième chapitre par présenter les capteurs embarqués formant le système de perception de l'*ATRV2*, nous donnons les principes physiques qui régissent leur fonctionnement et nous exposons les différentes sources d'erreurs inhérentes à leur utilisation. Ensuite, nous donnons le modèle choisi de l'environnement de navigation qui va être

utilisé tout au long de ce travail. A la fin de ce chapitre, nous proposons une approche permettant la fusion multisensorielle de données visuelles et télémétriques. Au début du quatrième chapitre, nous présentons les caractéristiques spécifiques du système mobile *ATRV2*, ses architectures matérielle et logicielle, ainsi que sa géométrie. La cinématique et le caractère holonome et non holonome du système sont des caractéristiques extrêmement importantes pour le développement de notre travail. Ils vont être utilisés dans le développement d'une nouvelle méthode de navigation nommée *DVFF*. Cette méthode combine une méthode de planification de trajectoire globale basée sur l'algorithme D^* , et une méthode réactive non stop basée sur le principe des champs de forces virtuelles *VFF*. Elle permet au système mobile *ATRV2* de s'orienter vers sa position destination tout en évitant les collisions avec les obstacles de son environnement.

Nous concluons ce mémoire de thèse en faisant le bilan de nos différentes contributions tout au long des chapitres, et l'analyse qualitative de l'ensemble des résultats obtenus. Nous évoquons ensuite quelques perspectives que nous trouvons les plus intéressantes pour la suite de ces travaux.

Partie I

Introduction aux systèmes de
robotique mobile.

Chapitre I

Systeme de perception des
systemes de robotique mobile.

Dans ce chapitre, nous présentons les différents types de capteurs utilisés dans les systèmes de robotique mobile disponibles sur le marché et les approches traitant de la modélisation-fusion de leurs mesures ainsi que les différentes représentations d'environnement proposées dans la littérature.

I.1. Capteurs utilisés dans les systèmes de robotique mobile.

Pour réaliser un système de robotique mobile intelligent, il est nécessaire d'utiliser des capteurs qui fourniront la perception requise de l'environnement pour une prise de décision intelligente.

A ce jour, il n'existe pas de capteur qui se démarque des autres par une efficacité vraiment supérieure. Ils ont tous des avantages et des inconvénients avec lesquels il faut pouvoir composer. Par contre, leur importance peut être passablement influencée par l'environnement dans lequel le système est amené à évoluer. Le défi est donc de réaliser des systèmes qui utilisent le mieux possible les capteurs ou une combinaison de capteurs en tenant compte de leurs conditions d'utilisation, et ceci requiert une bonne connaissance de leurs caractéristiques.

Nous pouvons définir deux catégories de capteurs couramment utilisées en robotique mobile : ceux qui délivrent une information caractérisant l'environnement et ceux qui interviennent dans le fonctionnement du système robotique. Les premiers sont appelés *capteurs extéroceptifs* qui fournissent des informations sur le monde extérieur au système mobile, et les seconds sont les *capteurs proprioceptifs* qui fournissent des informations propres au comportement interne du système mobile, c'est-à-dire sur son état à un instant donné.

I.1.1. Capteurs proprioceptifs.

Les capteurs proprioceptifs fournissent par intégration des informations élémentaires sur les paramètres cinématiques du système mobile. Les informations sensorielles gérées dans ce cadre sont généralement des vitesses, des accélérations, des angles de giration, des angles d'altitude. Cependant, ils ne peuvent pas procurer de renseignements lors de l'arrêt du système mobile.

On peut regrouper les capteurs proprioceptifs en deux familles [Fra90] :

- ◆ Les capteurs de déplacement qui comprennent les odomètres, les accéléromètres et les radars Doppler. Cette catégorie permet de mesurer des déplacements élémentaires, des variations de vitesse ou d'accélération sur des trajectoires rectilignes ou curvilignes.

- ◆ Les capteurs d'attitude, qui mesurent deux types de données : les angles de cap et les angles de roulis et de tangage. Ils sont principalement constitués par les gyroscopes, les gyromètres, les gyrocompas, les capteurs inertiels composites, les inclinomètres et les magnétomètres. Ces capteurs sont en majorité de type inertiels.

I.1.1.1. Capteurs de déplacement [Cou00].

a. Les odomètres.

Les systèmes odométriques fournissent la position du système mobile 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, etc.).

Le principe de l'odométrie est le suivant :

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

$$\delta\theta = \frac{R_d \delta\theta_d - R_g \delta\theta_g}{L} \quad (\text{I.1})$$

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 système mobile est :

$$\delta d = \frac{R_d \delta\theta_d + R_g \delta\theta_g}{2} \quad (\text{I.2})$$

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 système en fonction de la position $(x_{t-1}, y_{t-1}, \theta_{t-1})$ précédente par :

$$\begin{cases} x_t = x_{t-1} + \delta d_{t-1} \cos\left(\theta_{t-1} + \frac{\delta\theta_{t-1}}{2}\right) \\ y_t = y_{t-1} + \delta d_{t-1} \sin\left(\theta_{t-1} + \frac{\delta\theta_{t-1}}{2}\right) \\ \theta_t = \theta_{t-1} + \delta\theta_{t-1} \end{cases} \quad (\text{I.3})$$

Si l'on note $U_t = (\delta d_t, \delta\theta_t)^t$, on peut écrire les relations du système ci-dessus sous la forme :

$$X_{t+1} = f(X_t, U_t) \quad (\text{I.4})$$

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 les erreurs sur la position croissent au fur et à mesure du déplacement du fait de l'intégration des mesures nécessaire pour obtenir la position du système mobile.

Pour mesurer ou réduire ces erreurs, il est important de faire la différence entre l'erreur systématique et non systématique de ce capteur. Les erreurs systématiques sont celles qui sont une partie inhérente à la cinématique du système mobile ou à ces paramètres de contrôle indépendamment de

l'environnement. Les erreurs non systématiques sont celles qui dépendent de l'environnement du système mobile et diffèrent d'un environnement à un autre.

La distinction entre ces deux groupes est importante car chaque groupe influe différemment sur la plateforme mobile, leurs remèdes sont différents et plus important tous les deux nécessitent différentes techniques de mesure.

Borenstein et al. dans [Bor94,Bor95,Bof95,Bor96a,Bor98] ont étudié les différentes sources des deux types d'erreurs, et ont catégorisé ces erreurs comme suit :

- ◆ Les erreurs systématiques :
 - Diamètres des roues différents.
 - Roues mal positionnées.
 - Une résolution limitée des encodeurs.
 - Vitesse d'échantillonnage des encodeurs limitée.
- ◆ Les erreurs non systématiques :
 - Se déplacer sur un sol raboteux.
 - Se déplacer à travers des objets imprévus sur le sol.
 - Glissement des roues (sol lisse, excès d'accélération, patiner dans un tournant rapide, etc.).
 - Forces externes (interaction avec des corps externes).
 - Forces internes (roue folle).
 - Pas de contact avec le sol.

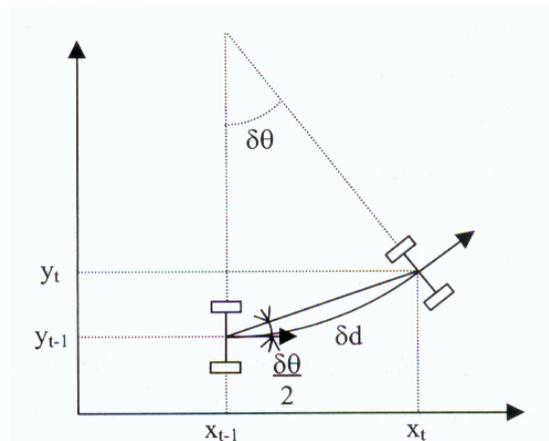


Figure I.1 : Principe de l'odométrie.

Dans la littérature, certains travaux ont réduit les problèmes liés à ce type de capteur. Dans [Cox91] les auteurs ont utilisé une référence indépendante périodiquement, qui par la suite a été appliquée par exemple dans la méthode de localisation dynamique du projet V.A.H.M (Véhicule Autonome pour Handicapés Moteurs)[Hor94], ou encore pour la correction de la position 3D d'un système mobile 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 système mobile au cours du mouvement [Mor85], ou à la fin du mouvement en mémorisant le trajet effectué [Cho97]. 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 système mobile d'une architecture particulière, comme le système mobile OmniMate de l'université de Michigan [Bor97b]. Ce système 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 système mobile.

b. Les accéléromètres.

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

c. Le radar Doppler.

Les capteurs radar Doppler sont employés couramment pour des applications aéronautiques et maritimes pour mesurer la vitesse linéaire du véhicule par rapport à la terre en se basant sur l'effet *Doppler-Fizeau*. Le principe est de diriger un faisceau électromagnétique de fréquence F vers le sol. Le faisceau reçu après rediffusion sur le sol présente un décalage de fréquence ΔF proportionnel à la vitesse V du déplacement relatif du véhicule par rapport au sol (Figure I.2). L'intégration de la mesure de vitesse fournira une estimation du déplacement du mobile.

Ces capteurs sont le plus souvent utilisés en milieu extérieur. Ils présentent le gros avantage d'être insensible au patinage et aux glissements des roues, raison pour laquelle ils sont employés dans certaines applications pour réduire les effets des erreurs odométriques. En revanche, leur précision se dégrade rapidement avec la diminution de la vitesse. De plus ils sont généralement lourds à mettre en oeuvre, aussi bien du point de vue électronique que du point de vue du traitement des informations reçues.

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].

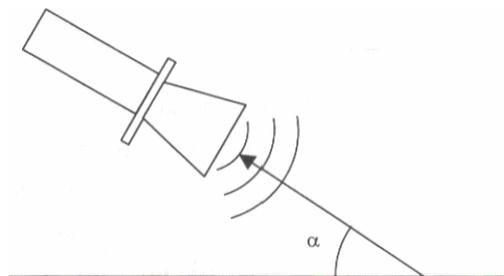


Figure I.2 : Principe du capteur à effet Doppler.

L'expression de la vitesse est :

$$v = \frac{c \Delta F}{2F \cos \alpha} \quad (I.5)$$

avec :

v : la vitesse de déplacement de l'émetteur,
 c : la vitesse de propagation de l'onde,
 ΔF : la différence de fréquence,
 F : la fréquence d'émission,
 α : l'angle d'inclinaison entre le capteur et le sol.

I.1.1.2. Capteurs d'attitude.

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

a. Les gyroscopes, gyromètre et gyrocompas.

Les gyroscopes permettent la mesure de l'angle de rotation de la trajectoire du système mobile. Ces mesures sont intégrées pour fournir la position du système mobile. Ces capteurs sont particulièrement répandus en positionnement de systèmes mobiles car ils aident à compenser pour la plupart, l'imprécision sur l'orientation fournie par l'odomètre [Fer91,Bor96b,Fuk96]. Une erreur d'orientation odométrique peut entraîner une erreur de position cumulative qui peut être diminuée voire compensée par l'utilisation conjointe de gyroscopes [Bor96b]. Les gyroscopes très précis sont trop onéreux pour être utilisés en robotique mobile. Cependant, les gyroscopes à fibre optique, connus pour leur grande précision, ont vu leur prix chuter et sont donc devenus une solution attractive pour la navigation en robotique mobile.

Le gyromètre est un capteur qui permet de mesurer une vitesse angulaire. Il existe plusieurs types de gyromètres : les premiers à avoir fait leur apparition furent mécaniques, aujourd'hui, on utilise surtout des gyromètres laser ou des gyromètres optiques [Vag93].

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

b. Les magnétomètres.

Les capteurs magnétiques sont des instruments qui mesurent la direction du champ magnétique terrestre. Ils sont utilisés pour déduire l'orientation du système mobile. Ces derniers étant perturbés par les masses métalliques, l'utilisation du compas magnétique n'est possible que pour des systèmes mobiles 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 système mobile [Arn93,Kar97].

I.1.2. Capteurs extéroceptifs.

Les capteurs extéroceptifs sont employés en robotique mobile pour collecter des informations sur l'environnement d'évolution du système mobile. Ils sont le complément indispensable aux capteurs proprioceptifs présentés précédemment. Des méthodes de fusion de données sont alors utilisées pour conditionner et traiter les informations sensorielles de natures différentes.

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 capteurs utilisés en robotique mobile sont : les capteurs télémétriques (les ultrasons, les lasers et les infrarouges), le GPS et les caméras.

I.1.2.1. Les capteurs télémétriques.

On appelle télémétrie toute technique de mesure de distance par des procédés acoustiques, optiques ou radioélectriques. L'appareil permettant de mesurer les distances est appelé *télémètre*. Il existe différentes technologies pour réaliser un télémètre.

Nous présentons dans ce qui suit les télémètres les plus utilisés dans les systèmes de robotique mobile, en donnant une idée de leur gamme de mesure et d'application.

a. Les ultrasons [Cou00,Lar00].

Les capteurs à ultrasons (ou télémètres ultrasonores) sont aujourd'hui les capteurs les plus communément employés dans les systèmes de robotique mobile autonome en raison de leurs bas prix, de leur faible consommation, de leur simplicité et de leur compacité (Figure I.3). 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.

Le capteur à ultrasons mesure le temps de vol d'une onde ultrasonore (de 20 Khz à 200 Khz) 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 [Dje05]. Il existe néanmoins des erreurs potentielles 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 système mobile afin de percevoir une grande partie de l'environnement. Par ce dispositif, des régions à trous correspondant à l'espace entre les cônes de deux capteurs voisins apparaissent (Figure I.4).

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.5) [Ohy94a,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, etc.). 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). La valeur de l'angle est donnée par l'expression suivante :

$$\theta = \frac{\Delta l}{2d} \tag{I.6}$$

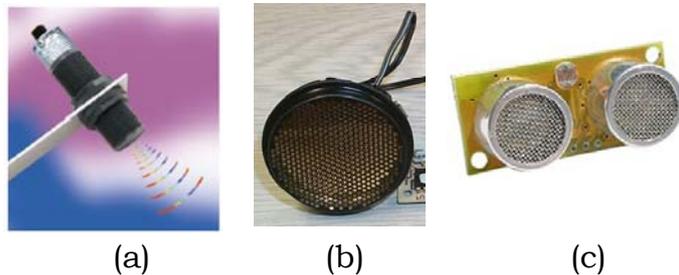


Figure I.3 : Exemples de télémètres ultrasonores, (a) Télémètre ultrasonore Polaroid Migatron RPS 409 IS, (b) Télémètre ultrasonore Polaroid série 600, (c) Télémètre ultrasonore MSU08.

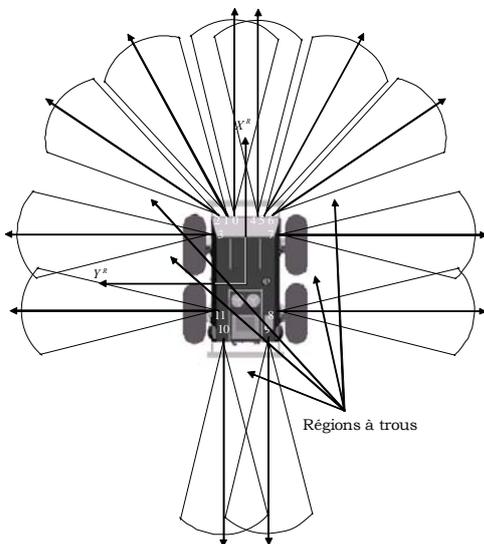


Figure I.4 : Exemple de ceinture de 12 capteurs à ultrasons.

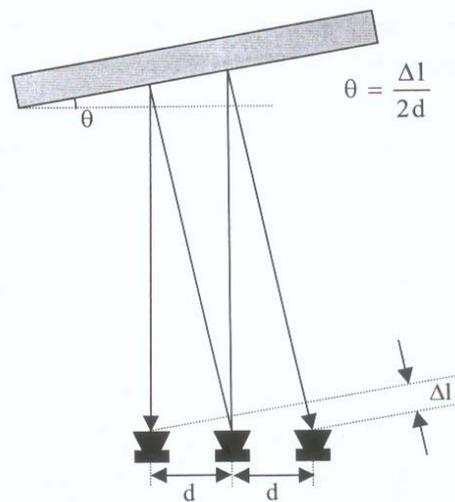


Figure I.5 : Exemple d'utilisation d'un capteur tri-aural.

b. Les capteurs à infrarouges.

Les capteurs à infrarouges fonctionnent suivant le même principe que les capteurs à ultrasons. Ils se composent d'un couple émetteur-récepteur (Figure I.6). 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 [Cou00].

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].

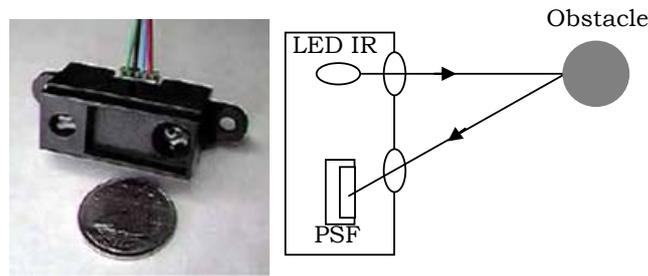


Figure I.6 : Télémètre infrarouge Sharp de la famille GP2.



Figure I.7 : Télémètre laser SICK fournissant 180 mesures réparties sur 180 degrés.

c. Les 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 (Figure I.7). 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, etc.) [Cou00]. De plus, ils nécessitent un entretien très fréquent car les mesures sont affectées si le système n'est pas propre et bien ajusté. Ils sont aussi grandement influencés par les conditions environnementales telles que la température, la neige, la pluie et la brume. Dans ces conditions, la qualité du signal est atténuée.

Il faut aussi considérer l'effet thermique du rayonnement laser qui peut représenter un certain danger. Il a été calculé qu'à une distance inférieure à 70 cm, le signal laser utilisé avec de tels systèmes est considéré comme étant dangereux, particulièrement pour l'œil humain [Dub96]. Il est donc nécessaire de tenir compte de cette particularité lors de la conception d'un système de détection, surtout s'il opère dans un milieu où des personnes peuvent se trouver.

I.1.2.2. Les capteurs à micro-onde.

Le capteur à micro-onde, ou radar, est basé sur le rayonnement d'un signal électromagnétique de l'ordre du gigahertz. Il peut fonctionner en trois modes différents [Lar00]. La télémétrie peut être utilisée pour fournir une information de distance, comme dans le cas des capteurs télémétriques précédents [Dub96, Eve95]. On peut aussi, en utilisant un radar à émission continue, exploiter le principe de l'effet Doppler afin d'obtenir une indication de mouvement d'une cible. Toutefois, l'effet Doppler ne donne aucune indication de distance [Dub96, Eve95]. Le troisième mode permet d'obtenir la distance en plus de déterminer si l'obstacle est mobile ou non. Il s'agit d'utiliser un radar à émission continue en lui ajoutant une modulation de la fréquence d'émission. Plusieurs formes d'ondes sont possibles pour effectuer la modulation [Dub96, Eve95].

Les capteurs à micro-onde ont comme avantages d'avoir une très longue portée (mètres à kilomètres) et une très grande précision. De plus, ils sont un des seuls types de capteurs à être pratiquement insensibles aux conditions environnementales comme la pluie et les écarts de température [Eve95]. Mais ils ont aussi leurs inconvénients, dont le principal est leur coût très élevé. Une autre caractéristique qui les rend inadéquats pour certaines applications est la dimension assez importante de l'émetteur.

L'effet des micro-ondes sur la santé des individus exposés aux rayonnements est un autre point à considérer [Dub96].

I.1.2.3. Les capteurs tactiles.

Les capteurs tactiles sont des détecteurs sensibles au contact. Il existe plusieurs variétés de capteurs tactiles qui peuvent avoir de nombreuses formes et différents niveaux de sensibilité. La plupart de ces capteurs ne sont pas coûteux. Dans un système de détection à distance, ce type de capteur n'est pas le centre d'intérêt mais peut quand même être utilisé pour compléter le système, en assurant par ailleurs une sécurité supplémentaire [Lar00].

I.1.2.4. Les GPS.

Un des systèmes les plus connus de ces dernières années est le Système de Positionnement Global (GPS) du département de la défense des Etats-Unis qui exploite une constellation de vingt-quatre satellites en orbite à haute altitude. Chaque satellite détecte les récepteurs terrestres fixes ou mobiles qui se trouvent dans sa zone de visibilité et transmet à chacun, à période fixe, un signal contenant des informations telles que son identification, l'heure d'émission du signal et les paramètres définissant son orbite donc sa position. Théoriquement, un récepteur doit être visible d'au moins trois satellites, pour pouvoir calculer sa position dans un repère centré sur la terre, en calculant l'intersection de trois

sphère (trois cercles pour un système mobile au sol) ; de fait, pour tenir compte des décalages temporels entre les mesures fournies par les satellites, quatre mesures de distance entre le satellite et le récepteur sont requises pour disposer d'une précision satisfaisante [Lau01].

Ce système est cependant loin de résoudre tous les problèmes de localisation des systèmes mobiles. Il fonctionne en effet difficilement dans des environnements urbains, et n'est pas utilisable à l'intérieur des bâtiments car les ondes employées sont fortement atténuées en traversant les murs. Sa précision est de plus souvent trop faible pour qu'un système mobile terrestre puisse utiliser ces informations seules. En pratique, il est souvent couplé à un système inertiel qui permet de pallier aux pertes du signal GPS et il ne remplace de toute façon pas les capteurs du système mobile qui lui permettent de percevoir son environnement immédiat [Fil04].

Mis à part son utilisation pour la localisation, 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, ainsi que dans la surveillance de camionnettes de livraison et de véhicules d'urgence, en vue de choisir au mieux leurs itinéraires.

I.1.2.5. Les caméras.

Depuis les années 70, les caméras CCD (Charge Coupled Device) ont été déterminantes dans l'évolution de la vision: la rapidité d'acquisition, la robustesse et la miniaturisation sont autant d'avantages qui ont facilité leur intégration. Elles sont très performantes en termes de portée, précision et quantité d'informations exploitables.

Elles sont de plus les seules capables de restituer une image sensorielle de l'environnement la plus proche de celle perçue par l'être humain. 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 et des couleurs.

En revanche, l'inconvénient majeur de tels capteurs se situe d'abord au niveau de la gestion du flux important de données exploitables (traiter une image demeure une opération délicate et surtout coûteuse en temps de calcul), ensuite à leur sensibilité aux conditions d'éclairage.

Pour augmenter le champ de vision de ces caméras, on peut les associer soit avec d'autres caméras ou soit à des systèmes de réflexion de type miroir : il s'agit des techniques de vision omnidirectionnelle. Cette association permet d'obtenir une vue sur 360 degrés de l'environnement [Dro02].

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

a. La stéréovision [Lem05].

La stéréovision consiste à observer une même scène : soit avec deux caméras éloignées l'une de l'autre et dont on connaît la distance qui les séparent, soit avec deux images successives prises à partir de deux points de vue différents. Connaissant la géométrie exacte du système stéréoscopique, il est possible de reconstruire en 3D la scène observée ou d'estimer la distance des objets et d'avoir ainsi une image de profondeur, qui peut être utilisée pour l'évitement d'obstacles ou la cartographie.

Afin de reconstruire en 3D la scène observée à partir des images acquises, le processus de stéréovision comporte une chaîne de traitement qui se présente de la façon suivante :

1. Le calibrage (ou calibration).
2. La rectification (étape optionnelle).
3. La mise en correspondance (ou appariement).
4. La reconstruction 3D.

Dans ce qui suit, nous présentons chacune des étapes décrites dans la chaîne de traitement.

a.1. Le calibrage.

La phase de calibration de caméra est considérée comme une étape importante pour de nombreux algorithmes et applications de la vision par ordinateur, en particulier les systèmes qui s'intéressent à récupérer l'information 3D perdue lors de l'acquisition des images. La calibration revient à déterminer analytiquement la fonction projective ou la transformation spatiale qui associe à chaque point 3D dans le repère scène sa projection 2D sur le plan image.

a.1.1. Calibrage d'une seule caméra [Cro97].

De nombreuses applications ne nécessitent pas un calibrage complet de caméras, mais, dès que nous souhaitons effectuer une reconstruction tridimensionnelle euclidienne métrique, il s'avère nécessaire d'effectuer ce calibrage. Le calibrage consiste à obtenir un modèle approché de la caméra.

Une caméra est un capteur sensible au rayonnement lumineux émis par l'environnement. Elle est définie comme étant un moyen pour obtenir une image 2D à partir d'un objet 3D.

Un modèle de caméra est souvent défini par des paramètres intrinsèques et extrinsèques. Les paramètres intrinsèques d'une caméra modélisent ses caractéristiques internes et ne dépendent donc pas de sa position ni de son orientation dans l'espace. Les paramètres extrinsèques, quant à eux, relient le système de coordonnées de la caméra au système de coordonnées du monde par une série de transformations 3D (rotation et translation). Ils représentent la position et l'orientation de la caméra par rapport à un repère fixe (repère du monde).

Le but du calibrage consiste à établir une relation mathématique entre les coordonnées 3D du monde réel et leurs coordonnées 2D correspondantes sur une image issue de la caméra. Cette relation est généralement obtenue sous forme d'une matrice, appelée matrice de transformation 3D-2D qui contient les

paramètres intrinsèques et extrinsèques. Une fois cette matrice calculée, l'information 3D peut être inférée à partir des coordonnées 2D et vice-versa.

Différentes techniques de calibrage ont été proposées dans la littérature. Elles peuvent toutes être classées en deux grandes familles selon qu'elles utilisent une mire de calibrage ou pas.

Une mire de calibrage est un objet du monde réel dont on connaît parfaitement la géométrie et à partir duquel seront extraits les points qui seront utilisés dans la phase de calibrage. Elle est généralement composée de motifs répétitifs (cercles ou rectangles) afin de choisir des points d'intérêt dont les coordonnées image (2D) peuvent être mesurées avec une plus grande précision.

Les techniques de calibrage qui utilisent une mire sont basées sur l'observation de cette dernière. Les coordonnées 3D de quelques points d'intérêt sont alors déterminées et leurs projections sont mesurées dans le repère image et on obtient ainsi les coordonnées 2D de ces points (en pixels).

Une fois les coordonnées 2D et 3D de ces points connues, il est possible d'estimer la matrice de transformation 3D-2D. La précision de la calibration dépend bien sûr de la précision de la grille et de celle des points d'intérêts choisis.

Les techniques de calibrage qui n'utilisent pas de mire (ou connues aussi sous le nom de calibrage automatique ou semi-automatique) supposent, elles, que seuls les paramètres intrinsèques de la caméra sont connus. Le calibrage automatique ne nécessite donc aucun appareillage de calibration et aucune connaissance a priori sur le système d'imagerie ou la scène, si ce n'est une liste de primitives 2D (points d'intérêt) mises en correspondance entre différentes vues.

La littérature fait souvent aussi état de calibrage *faible* ou *fort* ; la distinction se situe au niveau d'une estimation globale de la matrice de projection (cas faible) ou d'une estimation de chaque paramètre qui compose cette matrice (cas fort).

Dans ce qui suit, nous présentons un panorama non exhaustif des grandes tendances de la calibration de caméra [Ben02].

1. Méthode des multiplans.

C'est une méthode qui utilise plusieurs vues d'un même plan [Bas98b, Bas98a, Bea92, Bou99] dans un volume donné. La distance entre la caméra et le plan de calibration est mesurée avec une extrême précision. Le principe consiste à modéliser le volume généré par les positions quelconques du plan dans l'espace. Initié par la méthode des deux plans où chaque point contenu dans les plans est une ligne de visée, cette méthode ne nécessite aucune formulation géométrique. L'ensemble caméra et plan de calibration se comporte comme une boîte noire, où les paramètres à identifier sont obtenus par interpolation. En général, une dizaine de vues du plan contenant au minimum une vingtaine de points de repère sont nécessaires pour rendre la méthode robuste.

2. Méthode de Tsai.

Roger Tsai [Tsa86, Tsa89] présente une méthode basée également sur le principe du déplacement d'un plan mais avec la particularité de modéliser la calibration d'une caméra par une approche géométrique simplifiée. Cette

technique de calibration présente la possibilité d'estimer le coefficient qui exprime la distorsion radiale du système optique.

Une série d'hypothèses simplificatrices sont nécessaires pour utiliser cette méthode. Notamment la caméra est positionnée de sorte que son axe optique fasse au minimum 30 degrés avec la normale du plan de calibration. Le plan de calibration doit se déplacer de façon parallèle à sa position précédente afin d'obtenir une contrainte de parallélisme et annihiler les distorsions axiales. D'autres hypothèses supposent l'invariance de la distorsion avec le déplacement du plan de calibration. Une cascade de résolutions de systèmes linéaires et non linéaires est nécessaire pour aboutir à la détermination des paramètres intrinsèques et extrinsèques. Il faut plus de 200 points de mesure pour calibrer une caméra.

3. Méthode des points de fuite.

Cette méthode dont la communauté des chercheurs en vision attribue la primeur à Capril et Torre [Cap90], utilise les propriétés liées aux trois points de fuite [Cip99] d'un cube vu en perspective.

En effet, les points de fuite de trois directions de droites orthogonales deux à deux forment un triangle dont l'orthocentre est le point de fuite principal.

La méthode nécessite l'acquisition d'au moins deux images d'un cube de dimensions connues vu sous deux angles différents pour y étudier chaque face. Sur chaque face du cube figure des lignes parallèles et l'orientation des lignes de chaque face est différente de sorte à obtenir trois points de fuite distincts. L'ensemble des points de fuite répartis sur les deux images permet de définir un orthocentre.

C'est ainsi qu'est calculée la distance focale, en supposant que le centre optique est au centre de l'image, par un calcul linéaire pour ensuite être affinée par une méthode non linéaire intégrant les distorsions radiales. La matrice de rotation est estimée à partir des appariements des deux images (stéréovision) [Hor93].

4. Méthode de Faugeras et Toscani.

C'est l'une des méthodes de calibration de caméra qui est très souvent utilisée dans l'industrie. Les ingénieurs spécialistes du traitement d'images trouvent dans la thèse de doctorat de Gorgio Toscani [Tos87] un véritable document exploitable.

Toscani a abordé, dans sa thèse, la problématique de la calibration, ainsi que la localisation en stéréovision, par un langage mathématique couramment usité et d'une implémentation immédiate. Toscani s'approche de la calibration par un formalisme et par une méthodologie rigoureuse.

À l'aide d'une mire de calibration et à partir des équations de la projection perspective linéaire du modèle sténopé, cette méthode extrait les paramètres intrinsèques et extrinsèques par la résolution d'un système de $2n$ équations, où n est le nombre de points non coplanaires supérieur à 6 de la mire de calibration par deux approches.

La première méthode linéaire, issue de la photogrammétrie aérienne, permet de résoudre le système d'équations en proposant d'admettre que la distance entre le

repère de la caméra et le repère de l'objet de calibration soit ramenée à la valeur 1. La seconde méthode, très originale, a une approche non linéaire et utilise, pour résoudre le système d'équations, l'une des seules contraintes d'orthogonalité de la matrice de rotation qui ne subit par l'influence des paramètres intrinsèques et qui se nomme la contrainte Faugeras-Toscani [Fau86].

5. Méthode à base d'ellipses.

La méthode à base d'ellipses met en évidence un problème, souvent négligé, que l'on relègue aux opérations de traitement de l'image. Or cette nouvelle approche à la calibration est plus que nécessaire pour effectuer de bonnes mesures dans l'image.

A juste titre, Jean Philippe Tarel [Tar96] admet que la précision des données est souvent liée à la qualité des mesures et pour ce faire, l'auteur explicite la relation qui existe entre un objet dans l'espace et son image par une approche lambertienne. En effet, la lumière exercée sur une surface de la mire de calibration rayonne par une réflexion diffuse dans une demi-sphère gaussienne. D'après le théorème de Jean Henry Lambert : "*la quantité d'énergie émise à partir d'un élément de surface dans une direction déterminée est proportionnelle au cosinus de l'angle que fait cette direction avec la normale à la surface*". La réponse proportionnelle du capteur CCD à l'intensité reçue permet de mesurer, avec une grande précision, des caractéristiques géométriques, comme l'aire, le barycentre et les moments d'inertie.

6. Méthode par autocalibration.

Cette méthode est qualifiée par ses auteurs comme un procédé permettant d'extraire les paramètres intrinsèques d'une caméra sans intervention manuelle et sans mire de calibration.

L'intérêt majeur de cette technique réside dans l'utilisation de systèmes dits à vision active où sans cesse les paramètres optiques sont modifiés par le zoom ou la mise au point.

L'autocalibration est particulièrement adaptée lorsque la luminosité est fortement inégale ou lorsque la caméra travaille à des distances fortement variables. Thomas Skordas [Sko95] a réalisé un comparatif détaillé sur les trois catégories d'autocalibration. La première méthode est basée sur la géométrie épipolaire entre deux caméras et repose sur le calcul de la matrice fondamentale qui intervient entre les deux caméras. La seconde méthode recherche à estimer un grand nombre de paramètres issus d'un système d'équations quadratiques. La dernière méthode se base sur la connaissance a priori du mouvement de la caméra, d'où la nécessité de faire une série de mouvements (de la caméra) ainsi que l'acquisition de nombreuses images. La connaissance de points caractéristiques permet d'estimer les paramètres intrinsèques de la caméra.

a.1.2. Calibrage d'un banc stéréoscopique [Cro97].

Pour calibrer un capteur stéréoscopique composé dans notre cas de deux caméras², nous commençons tout d'abord par calibrer chaque caméra en utilisant les mêmes points de référence 3D et dans la même situation.

Ces deux calibrages vont donner deux matrices de projection perspective M_l et M_r à partir desquelles nous pouvons construire les différentes transformations (repère absolu/repère caméra, repère caméra/repère rétinien et repère rétinien/repère image) pour les deux caméras. Nous calculons ensuite la matrice $A_{l \rightarrow r}$ qui décrit la transformation rigide permettant de passer du repère caméra gauche au repère caméra droite en utilisant la relation suivante :

$$A_{l \rightarrow r} = A_r A_l^{-1} \quad (I.7)$$

où A_l et A_r sont respectivement la transformation repère absolu/repère caméra gauche et droite.

La figure suivante (Figure I.8) décrit la géométrie du capteur stéréoscopique.

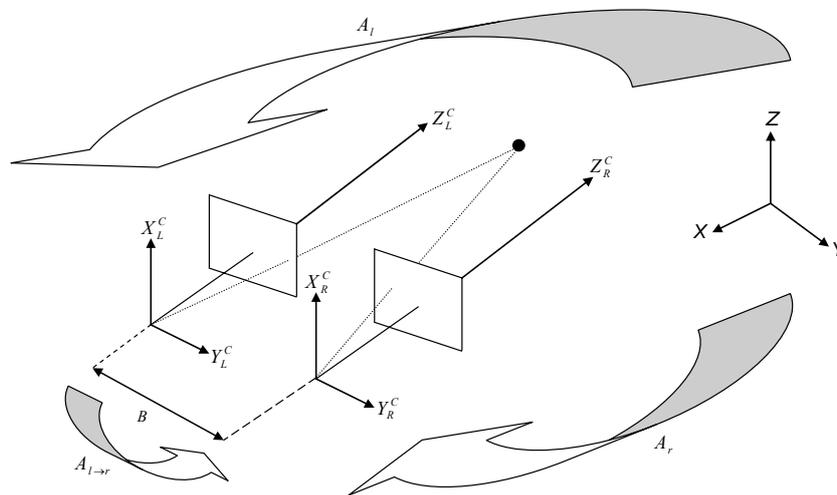


Figure I.8 : Géométrie du capteur stéréoscopique.

a.1.3. Modèle géométrique d'une caméra [Cro97].

Une caméra réalise une transformation ponctuelle qui fait passer d'un point physique de l'espace réel 3D à un point 2D sur le plan image de celui-ci. Ce qui revient à une transformation mathématique de R^3 vers R^2 .

On suppose généralement que la transformation réalisée est une projection centrale par rapport au centre de la caméra : c'est le modèle *Pinhole* ou modèle *Sténopé*.

Le modèle sténopé se caractérise par un plan de projection (le plan image) et un centre de projection (le centre optique F). La Figure I.9 montre qu'un point B de la scène se projette sur le plan image en un point b qui est l'intersection de la droite (FB) avec le plan image.

Avec ce modèle, le processus d'obtention d'une image peut être décrit de manière synthétique par la matrice de projection perspective (ou encore matrice de

² Le banc stéréoscopique utilisé est composé de deux caméras CCD couleur montées sur le système mobile ATRV2.

transformation 3D-2D). Il s'agit en fait de la matrice qui décrit la transformation qui permet de passer des coordonnées de B exprimées, par exemple, en millimètres dans un repère absolu tridimensionnel, à sa projection b sur le plan image décrite par ses coordonnées exprimées en pixels dans un repère bidimensionnel lié au plan image.

Si les coordonnées de B dans le repère absolu sont (x, y, z) alors les coordonnées (u, v) de sa projection b dans le repère image sont obtenues par une transformation linéaire en coordonnées homogènes qui s'écrit sous la forme matricielle suivante :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (\text{I.8})$$

$$\text{avec } M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \quad (\text{I.9})$$

M est la matrice de projection perspective. Elle permet de passer des coordonnées d'un point dans le repère absolu aux coordonnées de sa projection dans le repère image. Elle peut être décomposée en d'autres transformations à savoir :

- Une transformation qui permet de passer des coordonnées d'un point dans le repère absolu aux coordonnées de ce même point dans un repère lié à la caméra.
- Une transformation qui permet de passer des coordonnées d'un point de la scène dans le repère caméra aux coordonnées de sa projection dans le repère image.

1. Transformation repère absolu/repère caméra.

Soit (x, y, z) les coordonnées d'un point dans le repère absolu et (x_c, y_c, z_c) ses coordonnées dans le repère caméra. Le déplacement entre ces deux repères est constitué d'une rotation R et d'une translation T qui forment les paramètres extrinsèques de la caméra.

La transformation entre ces deux repères peut s'écrire sous la forme suivante :

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (\text{I.10})$$

où en coordonnées homogènes :

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (\text{I.11})$$

2. Transformation repère caméra/repère image.

Dans le repère caméra, la projection centrale d'un point de l'espace sur le plan rétinien est:

$$\begin{cases} x_r = f & x_c / z_c \\ y_r = f & y_c / z_c \end{cases} \quad (\text{I.12})$$

Ce système d'équations peut être réécrit sous forme matricielle en utilisant les coordonnées homogènes :

$$\begin{pmatrix} s x_r \\ s y_r \\ s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \quad (\text{I.13})$$

Les points images sont mesurés en pixels dans un repère bidimensionnel associé à l'image. Afin de pouvoir écrire la matrice de transformation du repère caméra au repère image, nous devons introduire les paramètres suivants : u_0 et v_0 , coordonnées de l'intersection de l'axe optique avec le plan image (mesurées en pixels), ainsi que k_v et k_u qui sont respectivement le facteur d'échelle vertical et horizontal (pixels/mm) car les pixels d'une caméra sont rarement carrés.

La relation entre les coordonnées d'un point de l'image dans le repère rétinien et celles du même point dans le repère lié à l'image est :

$$\begin{cases} u = -k_u x_r + u_0 \\ v = k_v y_r + v_0 \end{cases} \quad (\text{I.14})$$

ou encore en coordonnées homogènes :

$$\begin{pmatrix} s u \\ s v \\ s \end{pmatrix} = \begin{pmatrix} -k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s x_r \\ s y_r \\ s \end{pmatrix} \quad (\text{I.15})$$

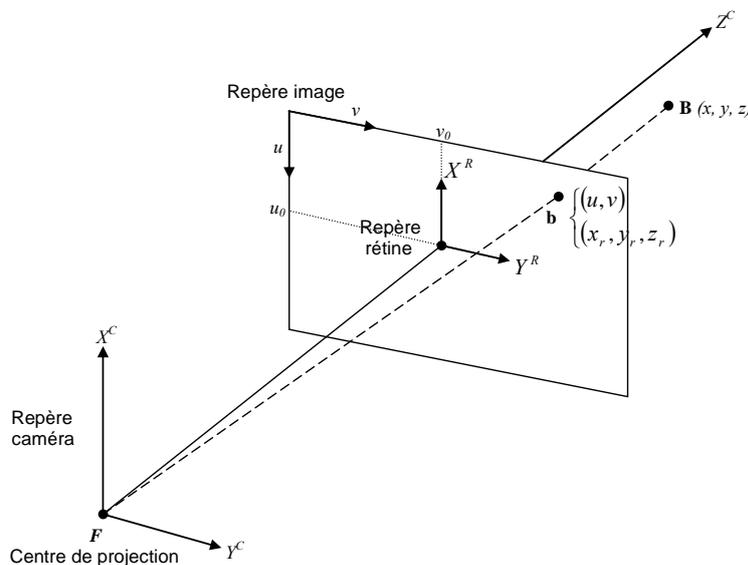


Figure I.9 : Modèle sténopé d'une caméra.

3. Transformation repère absolu/repère image.

La matrice de projection perspective permet de passer du repère absolu au repère image et vice-versa. En d'autres termes, elle permet le passage des coordonnées 3D (x, y, z) aux coordonnées 2D (u, v) .

Ce passage repère absolu/image se décompose en trois étapes :

- ❖ Une transformation repère absolu/repère caméra.
- ❖ Une transformation repère caméra/repère rétinien.
- ❖ Et enfin, une transformation repère rétinien/repère image.

Ainsi à partir des équations I.11, I.13 et I.15, la matrice de projection M s'écrit comme suit :

$$\begin{aligned}
 M &= \begin{pmatrix} -k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} -f k_u r_{11} + u_0 r_{31} & -f k_u r_{12} + u_0 r_{32} & -f k_u r_{13} + u_0 r_{33} & -f k_u t_x + u_0 t_z \\ f k_v r_{21} + v_0 r_{31} & f k_v r_{22} + v_0 r_{32} & f k_v r_{23} + v_0 r_{33} & f k_v t_y + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \quad (I.16) \\
 &= \begin{pmatrix} \alpha_u r_{11} + u_0 r_{31} & \alpha_u r_{12} + u_0 r_{32} & \alpha_u r_{13} + u_0 r_{33} & \alpha_u t_x + u_0 t_z \\ \alpha_v r_{21} + v_0 r_{31} & \alpha_v r_{22} + v_0 r_{32} & \alpha_v r_{23} + v_0 r_{33} & \alpha_v t_y + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}
 \end{aligned}$$

En identifiant l'équation (I.16) avec (I.9), et en tenant compte des propriétés d'orthogonalité de la matrice de rotation R , nous obtenons un ensemble d'équations qui permettent de calculer les paramètres intrinsèques et extrinsèques en fonction des coefficients de la matrice M . Nous aurons donc :

$$\left\{ \begin{array}{l} r_3 = m_3 \\ u_0 = m_{1 \cdot} (m_3)^t \\ v_0 = m_{2 \cdot} (m_3)^t \\ \alpha_u = -\|m_1 \wedge m_3\| \\ \alpha_v = \|m_2 \wedge m_3\| \\ r_1 = \frac{(m_1 - u_0 m_3)}{\alpha_u} \\ r_2 = \frac{(m_2 - v_0 m_3)}{\alpha_v} \\ t_x = \frac{(m_{14} - u_0 m_{34})}{\alpha_u} \\ t_y = \frac{(m_{24} - v_0 m_{34})}{\alpha_v} \\ t_z = m_{34} \end{array} \right. \quad (I.17)$$

Pour trouver les paramètres intrinsèques et extrinsèques, nous devons donc :

1. Calculer la matrice de projection M .

2. Extraire les paramètres de la caméra à partir de la matrice M grâce aux formules fournies par le système d'équations (I.17).

a.2. Rectification.

La rectification est une étape qui simplifie le processus de la vision stéréoscopique dans certaines applications. En effet, elle permet de se ramener à une géométrie épipolaire simple, vu que nous obtenons des lignes épipolaires parallèles aux lignes des images (Figures I.10 et I.11). Donc les points homologues des deux images ont la même ordonnée (ils se trouvent sur la même ligne épipolaire), ce qui réduit la recherche des correspondants à une recherche monodimensionnelle. Ceci rend cette configuration beaucoup plus intéressante du point de vue algorithmique.

La rectification est donc une transformation des images qui dépend uniquement de la géométrie du système d'acquisition (paramètres des caméras et leur position relative).

Il existe plusieurs méthodes de rectification. Les méthodes de rectification traditionnelles [Aya88] consistent en une transformation des plans images afin d'obtenir des plans espaces qui coïncident. Il existe plusieurs variantes de ces méthodes telles que dans [Cro97, Fau93b, Pap96] qui ont démontré leurs limites lorsque les épipoles sont localisés dans les images. Roy [Roy97] a proposé une méthode pour résoudre ce problème, mais elle reste complexe et rencontre certains problèmes. Pollefeys [Pol99] a proposé une méthode qui consiste à sélectionner le plan parallèle à la ligne de base, puis de projeter les deux images sur ce plan. Zhang [Loo99] et Hartley [Har00] quant à eux ont proposé une méthode qui repose sur une décomposition homographique. Cette méthode peut être appliquée dans le cas où les images sont calibrées ou non, ce qui rend la méthode plus intéressante.

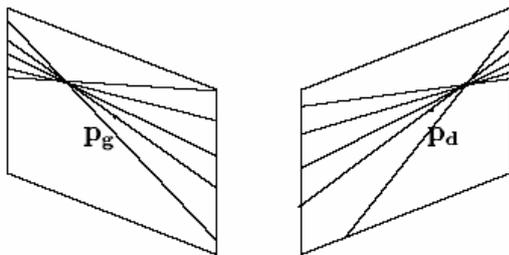


Figure I.10: Faisceaux de lignes épipolaires: cas général non réctifié.

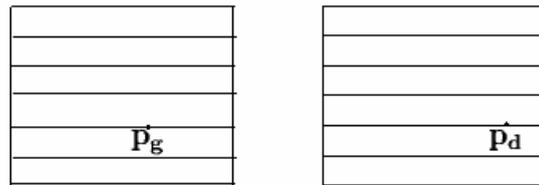


Figure I.11: Faisceaux de lignes épipolaires: cas rectifié.

a.3. Mise en correspondance.

La mise en correspondance est communément considérée comme une étape clé dans le processus de stéréovision. C'est aussi l'une des étapes les plus difficiles du fait de son aspect combinatoire. En effet, sans aucune autre information pour retrouver les correspondances d'une primitive donnée, il faut la comparer à toutes les autres primitives.

Cette comparaison est elle-même délicate du fait des pertes occasionnées lors de l'acquisition des images. Vu son importance majeure, la mise en correspondance stéréoscopique a suscité de nombreuses recherches. Dans la littérature les

algorithmes de mise en correspondance stéréoscopique existants sont classifiés en deux catégories:

a.3.1. Méthodes à base de primitives.

Dans cette catégorie, les algorithmes extraient les primitives saillantes [Aya88,Asc92,Hri88,Mor79] telles que : les segments, les contours, les coins, etc., puis les appariant dans deux ou plusieurs images. Ces méthodes sont beaucoup plus rapides grâce à la réduction de l'information contenue dans les images. Elles permettent d'obtenir une précision égale à celle des primitives extraites.

Cependant, ces méthodes sont fortement limitées par le nombre des primitives présentes dans les images. Les résultats obtenus sont des données 3D éparses.

Dans la littérature, nous trouvons plusieurs approches de mise en correspondance à base de primitives. Néanmoins, nous citons les plus souvent rencontrées :

- Algorithme de relaxation.
- Approches hiérarchiques.
- Utilisation des graphes.
- Approches énergétiques.
- Réseaux de neurones.
- Algorithme de prédiction et vérification d'hypothèses.

a.3.2. Méthodes surfaciques.

Ces méthodes consistent à fixer d'abord un point d'intérêt dans une image, puis à l'aide d'une mesure de similarité (distance euclidienne par exemple), nous cherchons un point dans l'autre image dont le voisinage est semblable à celui du point de référence. Ces méthodes proposent généralement de fournir une solution pour chaque point d'intérêt.

Elles ont l'avantage, par rapport aux méthodes à base de primitives, de produire une représentation dense de la scène observée. Parmi ces méthodes, nous citons les plus souvent rencontrées dans la littérature:

- Algorithmes corrélatifs (ou recherche directe)
- Programmation dynamique.
- Approche multi résolution.
- Flot maximum.
- Recuit Simulé.
- Champs de Markov.

a.3.3. Contraintes de mise en correspondance.

A cause de facteurs tels que le bruit, les variations de la lumière, les occlusions et la distorsion perspective, l'apparence des primitives correspondantes diffère dans les différentes vues, ce qui engendre un ensemble de candidats et rend le choix du bon correspondant délicat et crucial. Afin d'éliminer l'ambiguïté et de valider les bons appariements, nous imposons les contraintes suivantes [Fau93a] pour mieux localiser les correspondants potentiels et éliminer les faux appariements.

1. Contrainte épipolaire.

La contrainte épipolaire permet, une fois une primitive particulière sélectionnée dans la première image, de réduire l'espace de recherche de l'homologue à une dimension dans la seconde image. La recherche ne se fait plus dans toute l'image, mais sur la droite épipolaire.

2. Contrainte d'unicité.

La contrainte d'unicité se traduit par le fait qu'une primitive dans la première image a au plus un homologue dans la seconde image. Une primitive est la projection d'une seule entité 3D.

3. Contrainte d'ordre.

La contrainte d'ordre permet de vérifier la compatibilité de deux appariements en considérant que l'ordre dans lequel sont disposées les primitives appartenant à la première image est celui dans lequel leurs homologues sont disposés.

4. Contrainte de lissage.

La contrainte de lissage s'appuie sur le fait que le monde physique est constitué de surfaces continues. Elle impose que les primitives voisines ont la même disparité.

5. Contrainte lambertienne.

La luminance ou l'intensité des primitives homologues est supposée la même dans les différentes vues. On dit dans ce cas que les surfaces sont lambertiennes. Ce qui n'est pas toujours le cas. Pour cela, on suppose que les surfaces sont quasi-lambertiennes. En d'autres termes, elles ont presque la même intensité.

D'autres contraintes peuvent être imposées. Nous venons de citer les plus utilisées. Dans ce qui suit, nous allons énumérer les problèmes rencontrés lors de la phase de mise en correspondance.

a.3.4. Les difficultés liées à la mise en correspondance.

Lors de l'acquisition des images, une masse importante d'informations est perdue. Pour récupérer l'information la plus importante qui est la profondeur (ou la disparité), nous utilisons les techniques de mise en correspondance entre les images prises selon différents points de vue. C'est grâce au changement du point de vue que nous pourrions retrouver l'information 3D.

Mais malheureusement cette étape possède des difficultés causées par plusieurs facteurs, les plus importants sont:

1. Occultation.

Ce sont des zones visibles de la scène dans une image et invisibles dans les autres, ceci est dû principalement au changement du point de vue. Ces zones n'auront donc pas de correspondant ce qui représente un problème crucial. C'est pour ces raisons que leur détection est nécessaire afin qu'elle n'influe pas sur

l'appariement. Plusieurs algorithmes de mise en correspondance essaient de traiter ce problème afin d'obtenir de bonnes cartes de disparité.

2. Différence de luminosité.

La présence des sources lumineuses dans une scène et le changement du point de vue, induit une variation de luminosité entre les images appariées. Cette variation influe négativement sur l'appariement puisque la mesure de ressemblance (ou de dissemblance) se base essentiellement sur la luminosité ainsi que sur la contrainte lambertienne.

3. Bruit.

L'acquisition des images se produit avec un certain bruit qui est dû en particulier aux imperfections des dispositifs qui génèrent les images, c'est-à-dire les capteurs (caméra ou détecteur électronique). Un bon capteur est celui qui produit un faible bruit de lecture. Les bruits ne pouvant pas être éliminés une fois introduits dans l'image brute, il convient de les minimiser car ce sont eux qui limitent la précision de mesure dans les traitements des images en particulier la mise en correspondance.

a.4. Reconstruction 3D.

La reconstruction tridimensionnelle consiste à calculer la géométrie de la scène observée par le système d'acquisition. Elle est obtenue à partir des résultats de la phase de mise en correspondance en utilisant les paramètres de calibrage du système de caméras utilisées. Selon le type de calibrage utilisé, nous pouvons procéder à trois types de reconstruction [Fau92].

a.4.1. Reconstruction projective.

La reconstruction projective [Har94] de la surface est utilisée dans le cas où le système est faiblement calibré. Ce type de reconstruction est le moins riche vu qu'il ne requiert aucune connaissance a priori sur la scène ou sur les paramètres des caméras.

a.4.2. Reconstruction affine.

La reconstruction affine [Zel96] requiert une connaissance du plan à l'infini qui permet de déterminer les caractéristiques affines de la scène. Pour cela, il suffit d'identifier trois paires de droites parallèles ou bien connaître les paramètres de calibrage.

a.4.3. Reconstruction euclidienne.

À partir des paramètres obtenus par auto-calibrage ou par calibrage fort, nous pouvons procéder à une reconstruction euclidienne [Fau95]. La reconstruction affine peut être aussi utilisée en imposant d'autres contraintes euclidiennes.

b. Les techniques de lumières structurées.

L'obtention de l'information 3D est gérée comme pour la stéréovision, mais on remplace une caméra par un laser qui sera chargé de projeter un motif de

lumière structurée. Le principe est d'éclairer l'environnement avec une lumière cohérente et de l'observer avec une caméra sous une certaine parallaxe. Connaissant la position de la source lumineuse et celle de la caméra, l'information peut être obtenue par triangulation.

Ces capteurs sont utilisés pour obtenir des images de profondeur, mais également pour l'évitement d'obstacles [Ohy94b, Fof97], pour la numérisation d'objet 3D [Pap97].

Plusieurs patrons de lumière structurée peuvent être utilisés :

- *Capteur à simple triangulation* : les capteurs à simple triangulation acquièrent la distance d'un seul point lumineux à la fois. Une application célèbre est celle du véhicule Pathfinder destiné à l'exploration de la planète Mars [Mat95].
- *Capteur par plan de lumière* : la source de lumière projetée n'est plus un point, mais un plan de lumière. Un système de localisation basé sur ce principe a été développé dans [Jou01].
- *Capteur à plusieurs plans de lumière* : il s'agit dans ce cas de projeter un motif tel que des lignes ou de petits disques sur la scène observée (Figure I.12).

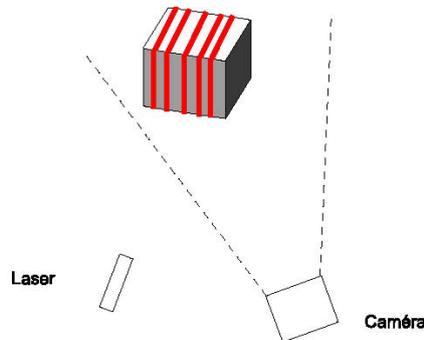


Figure I.12 : Triangulation par plusieurs plans Laser.

Ces techniques sont plus robustes que la stéréovision car elles permettent de faire l'économie de l'étape de mise en correspondance. En revanche, la précision n'est pas uniforme sur l'ensemble de l'image : plus un objet est éloigné, plus l'erreur de mesure est importante [Fof97].

I.1.3. Conclusion.

Nous avons montré dans ce paragraphe, qu'un système de perception d'un système de robotique mobile est généralement composé de deux catégories de capteurs. Ces catégories sont plus complémentaires que concurrentes, c'est pour cette raison qu'un système de navigation sera généralement basé sur l'exploitation des informations sensorielles émanant de ces deux catégories de capteurs. Nous pouvons étendre cette remarque à la problématique spécifique qu'est la perception du milieu d'évolution d'un système mobile : l'emploi d'un seul capteur pour une tâche donnée sera généralement insuffisant. Ainsi, l'association de plusieurs capteurs, qu'ils soient complémentaires ou redondants, permettra d'obtenir un modèle sensoriel robuste et hautement descriptif.

I.2. Fusion de données capteurs.

Les systèmes de robotique mobile utilisent de plus en plus plusieurs types de capteurs. Ils comportent de nombreux capteurs qui acquièrent de l'information

complémentaire ou redondante afin d'obtenir une perception plus complète ou plus fiable de leur environnement.

Mais la façon de gérer l'information provenant de tous ces capteurs demeure une préoccupation. Il est difficile de combiner des données de capteurs de nature différente (vision et acoustique, par exemple) permettant la perception de caractéristiques totalement différentes des objets, ou de capteurs de même type placés à différents endroits diminuant ainsi les problèmes dus à l'occultation, la résolution ou au champ de vue. Il y a donc un besoin croissant pour des méthodes efficaces de fusion de données, et beaucoup de recherches sont actuellement consacrées à ce sujet.

Dans ce contexte, [Vis99] a fait la distinction entre trois types de fusion:

- Fusion *complémentaire* : Chaque capteur ne fournit qu'une information partielle de l'état du système. Très souvent, un capteur proprioceptif (comme ceux utilisés dans l'odométrie) est combiné à un capteur extéroceptif (balises) [Bor97a]. Ce type de fusion permet de fournir une information ne pouvant être obtenue séparément avec chacun des capteurs.
- Fusion *compétitive* : les capteurs de type différent mesurent le même paramètre (exemple : un sonar et une caméra mesurant la distance d'un objet au système mobile). Ce type de fusion accroît la précision et le degré de confiance d'une donnée obtenue avec plusieurs capteurs distincts.
- Fusion *coopérative* : l'un des capteurs ne réalise une mesure qu'après avoir reçu le résultat de mesure d'un autre capteur (exemple : deux caméras d'un montage stéréoscopique sont nécessaires pour estimer la dimension d'un objet situé dans leur champ commun).

Dans ce qui suit, nous proposons un panorama des principales méthodes de fusion numérique, à savoir les approches classiques, les méthodes liées à un traitement non probabiliste de l'incertain, les méthodes dérivées de l'intelligence artificielle et les méthodes d'optimisation [Tec,Bre04].

I.2.1. Méthodes classiques.

I.2.1.1. Raisonnement Bayésien.

L'utilisation de raisonnement de type Bayésien basé sur la théorie des probabilités fait partie des méthodes de prise de décision les plus anciennes. L'inférence utilisant le théorème de Bayes permet de mettre à jour la *vraisemblance d'une hypothèse* à partir d'une estimation précédente de cette vraisemblance et d'observations nouvelles.

Cette technique peut utiliser des probabilités classiques (le type d'incertitude représenté est lié à la fréquence d'apparition des événements) ou des probabilités subjectives. Supposons que H_1, \dots, H_n représentent des hypothèses exhaustives et mutuellement exclusives pouvant expliquer une observation E , alors :

$$p(H_i/E) = \frac{P(E/H_i)P(H_i)}{\sum_i P(E/H_i)P(H_i)} \quad (\text{I.18})$$

avec :

$$\sum_i P(H_i) = 1,$$

$P(H_i/E)$ la probabilité *a posteriori* que l'hypothèse H_i devienne vraie étant donnée l'évidence E ,

$P(H_i)$ la probabilité *a priori* que l'hypothèse H_i devienne vraie,

$P(E/H_i)$ la probabilité d'observer l'évidence E sachant que H_i est vraie.

Si l'on considère k capteurs observant une entité dont l'identité est inconnue, si cette entité peut avoir m identités différentes, chaque capteur C peut fournir n_c déclarations différentes sur l'identité de l'objet, on connaît les probabilités *a priori* que le capteur déclare que l'objet est du type i lorsqu'il est du type j : $P_c(D_i/O_j)$.

Si le capteur était seul, on obtiendrait la probabilité *a posteriori* que l'objet soit de type j par :

$$P_c(O_j/D_c) = \frac{P_c(D_c/O_j)P(O_j)}{\sum_j P_c(D_c/O_j)P(O_j)} \quad (I.19)$$

Pour combiner l'ensemble des déclarations des capteurs, on peut également employer la formule de Bayes :

$$P(O_j/D_1 \wedge D_2 \wedge \dots \wedge D_k) = \frac{P(O_j)P_1(D_1/O_j) \dots P_k(D_k/O_j)}{\sum_i P(O_i)P_1(D_1/O_i) \dots P_k(D_k/O_i)} \quad (I.20)$$

On peut alors prendre une décision concernant la nature de l'objet par exemple en choisissant l'objet O_j pour lequel la probabilité conjointe $P(O_j/D_1 \wedge D_2 \wedge \dots \wedge D_k)$ est maximale.

On peut utiliser ce type de méthode à différents niveaux, que ce soit à partir d'observations multicapteurs ou pour combiner des évidences de plus haut niveau.

I.2.1.2. Utilisation d'estimateurs.

L'utilisation d'estimateurs s'applique aux données de bas niveau : ils permettent de convertir des données observées en estimations de l'état d'une cible par exemple. Il s'agit donc plus spécifiquement de méthodes de reconstruction de l'environnement. Les problèmes liés à la localisation statique sont résolus par des estimateurs qui, à partir de données multiples prises aux mêmes instants, fournissent une estimation de position, par contre, dans les cas de suivi dynamique, ces estimateurs s'appliquent à des séquences temporelles de données. L'estimateur statistique le plus utilisé est le filtre de Kalman qui fournit une solution récursive aux problèmes d'estimation à variance minimale.

L'équation d'état reliant l'état X_k à l'instant k à l'état à l'instant précédent X_{k+1} est:

$$X_{k+1} = \Phi_k X_k + W_k \quad (I.21)$$

où Φ_k est la matrice de transition d'état, W_k est le bruit du processus supposé blanc et à moyenne nulle, de covariance Q_k .

Le modèle de mesure relie l'observation Y_k à l'état :

$$Y_k = H_k X_k + V_k \quad (I.22)$$

avec H_k matrice de mesure, V_k bruit de mesure, blanc, de moyenne nulle, de covariance R_k .

Ce filtre nécessite la connaissance des statistiques de second ordre des bruits du processus et de la mesure pour fournir la solution qui minimise l'écart quadratique moyen entre la valeur de l'état et son estimation.

La matrice de gain de *Kalman* est :

$$K_k = P_{k(-)} H_k \left[H_k P_{k(-)} H_k^T + R_k \right]^{-1} \quad (I.23)$$

I.2.2. Méthodes liées à un traitement non probabiliste de l'incertain.

Les probabilités sont l'une des mesures de confiance possibles. On peut utiliser pour la fusion d'autres mesures non additives, qui peuvent donner lieu à des combinaisons et des agrégations de types plus variés et supposent moins de connaissance *a priori* sur les données.

I.2.2.1. Combinaison par la théorie des possibilités.

Les mesures de possibilité et de nécessité constituent la première famille utilisable. On peut associer, à une source d'information X , une distribution de possibilités notée π_x , prenant ses valeurs sur l'intervalle $[0,1]$; $\pi_x(x)$ représente le degré de possibilité pour que $X=x$.

Les possibilités permettent de manipuler conjointement les notions d'imprécision et d'incertitude, ce qui permet à la source d'information d'être caractérisée par cette distribution de possibilité. Les manières de fusionner des distributions de possibilité correspondent à trois types d'opérateurs [Blo94] :

- les opérateurs qui ont un comportement constant et indépendant du contexte;
- les opérateurs qui ont un comportement variable et indépendant du contexte;
- les opérateurs qui ont un comportement dépendant du contexte.

Le premier type comprend l'intersection des distributions de possibilité des sources qui donne donc une idée pessimiste de la combinaison (T-normes), l'union de ces distributions qui au contraire donne une vision trop optimiste (T-conormes) et les moyennes de différents types peu adaptées à la fusion si les informations sont contradictoires. La Figure I.13 donne un exemple avec la T-norme min et la T-conorme max.

Le second type correspond aux sommes symétriques qui, selon les valeurs respectives des sources, se comportent comme des opérateurs de conjonction, de disjonction ou de moyenne.

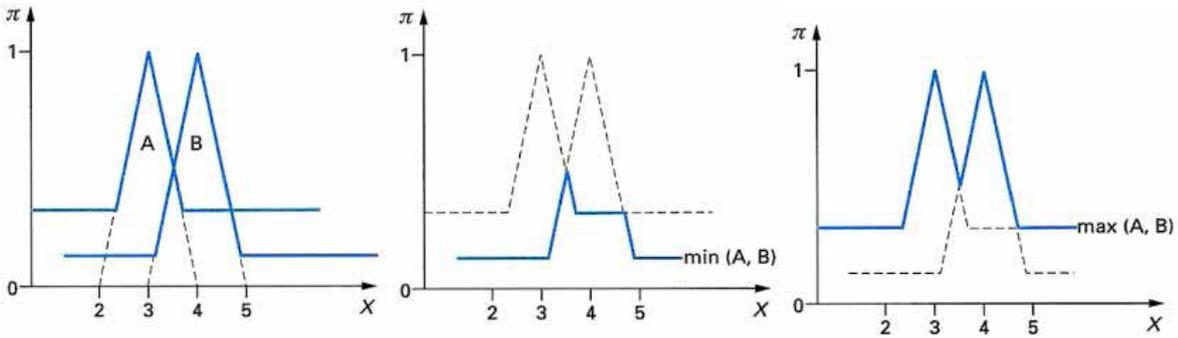


Figure I.13 : Exemple d’opérateurs à comportement constant.

Les opérateurs dépendant du contexte sont quant à eux beaucoup plus puissants car, selon le conflit existant entre les sources ou la fiabilité de celles-ci, on module leur comportement : par exemple, si les sources présentent un consensus, on réalisera une conjonction, s’il y a conflit, on réalisera une disjonction et si le conflit est faible, on réalisera un compromis.

L’une des règles heuristiques repose sur une mesure de conflit $1-h$ entre deux sources A et B, dont les informations sont caractérisées par des distributions de possibilité π_A et π_B .

$$1-h = 1 - \sup_x \min(\pi_A(x), \pi_B(x)) \tag{I.24}$$

La combinaison de A et B par cet opérateur adaptatif donne une distribution de possibilité π_{adap} calculée de la façon suivante :

$$\pi_{adap}(x) = \sup \left(\frac{\min(\pi_A(x), \pi_B(x))}{h}, \min(\max(\pi_A(x), \pi_B(x)), 1-h) \right) \tag{I.25}$$

Plus le conflit entre deux informations est important (Figure I.14), plus l’incertitude sur le résultat augmente.

L’utilisation des possibilités est donc intéressante en fusion quelque soit le niveau de celle-ci.

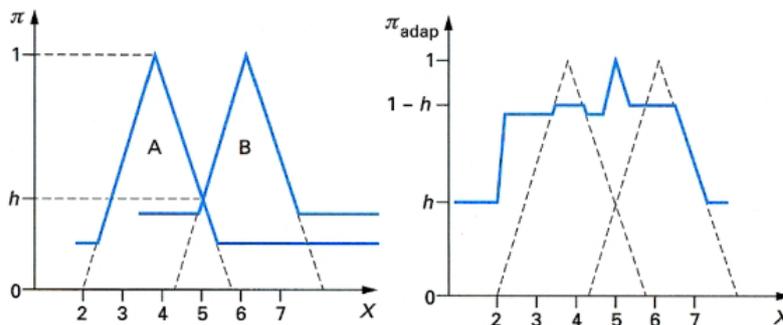


Figure I.14 : Fusion en cas de conflit important.

I.2.2.2. Utilisation de la théorie de l’évidence.

L’autre mesure de confiance très utilisée concerne la théorie de l’évidence appelée aussi théorie des fonctions de croyances ou de *Dempster Shafer*. Cette théorie permet de représenter l’ignorance de façon explicite en se dispensant de la clause d’équirépartition qui existe avec une mesure additive [App91].

La méthode de *Bayes* donne les mêmes résultats que la théorie de l'évidence lorsque les hypothèses sont mutuellement exclusives et que l'ensemble des hypothèses est exhaustif, ce qui est rarement le cas en pratique [Tec].

Dans de nombreuses applications, cette modélisation de l'ignorance est en effet nécessaire, que ce soit au niveau des capteurs délivrant des informations incomplètes ou à un niveau sémantique plus élevé, pour déterminer la vraisemblance d'hypothèses basées sur des évidences multiples.

L'ensemble des propositions élémentaires est appelé *cadre de discernement*. On assigne à chaque ensemble connu de faits (correspondant soit à une proposition élémentaire soit à une combinaison de propositions élémentaires) une masse m appelée *masse d'assignement probabiliste* qui permet de calculer la crédibilité et la plausibilité de ces combinaisons.

La crédibilité Cr (quelquefois notée Bel , pour *belief*) s'obtient à partir des masses élémentaires, par exemple pour trois propositions élémentaires :

$$Cr(A_1 \vee A_2 \vee A_3) = m(A_1) + m(A_2) + m(A_3) + m(A_1 \vee A_2) + m(A_1 \vee A_3) + m(A_2 \vee A_3) + m(A_1 \vee A_2 \vee A_3) \quad (I.26)$$

De la même façon, la plausibilité Pl d'une proposition A_i est définie par :

$$Pl(A_i) = 1 - Cr(\bar{A}_i) \quad (I.27)$$

La sortie d'un processus de *Dempster Shafer* est un ensemble d'intervalles d'évidence (Cr, Pl) qui encadrent la probabilité.

Il s'avère que, particulièrement dans les applications de type militaire, où les observations peuvent être troublées par des contre-mesures, la possibilité donnée dans cette théorie à la modélisation de l'ignorance est particulièrement appréciable [Tec].

I.2.3. Méthodes dérivées de l'intelligence artificielle.

Dans de nombreuses applications de fusion de données [Zui92], on rencontre la nécessité de réaliser des déductions de haut niveau qui peuvent éventuellement compléter une fusion bas niveau numérique ayant permis d'obtenir une estimation de paramètres ou une identification. Ces déductions peuvent concerner l'interprétation de contexte, la prédiction d'intentions, la découverte de relations entre des faits ou des objets, le diagnostic. On trouvera donc l'utilisation de systèmes à base de connaissances tant dans des applications d'interprétation d'imagerie (médicale ou satellitaire) que dans des applications robotiques ou militaires.

Différents types de raisonnement peuvent être nécessaires : la mise en correspondance, le raisonnement temporel ou spatial, l'induction ou la déduction, l'apprentissage, le traitement du langage naturel.

De nombreuses applications ont été réalisées en utilisant des systèmes experts bâtis à partir d'un des nombreux générateurs disponibles commercialement.

Un système expert comprend une *base de connaissances* qui contient l'expertise (sous forme de règles de production, de réseau sémantique, de frame ou de script), une *base de données* qui enregistre la dynamique du système et un

moteur d'inférence qui permet de tirer des conclusions. Le processus d'inférence fonctionne en chaînage avant ou arrière. La sélection de la règle à activer peut être effectuée par une technique de réfraction, d'actuation, de classement de règles, de datage, de spécificité ou par un choix aléatoire. Le système expert permet d'associer diverses techniques de représentation de l'incertitude : probabilité, raisonnement basé sur l'une des théories de l'évidence, facteurs de confiance, ensembles flous. Le choix d'une telle technique est délicat et doit, le plus souvent être basé sur l'expérimentation afin de déterminer la méthode la plus intéressante vis à vis de l'utilisation visée.

I.2.4. Méthodes d'optimisation.

On peut formuler le problème de la fusion de données comme un problème d'optimisation combinatoire. Les réseaux de neurones permettent, après une phase d'apprentissage, de fusionner des informations de natures différentes.

Dans la phase d'association, quand le nombre de cibles est important, des techniques de traitement parallèle comme les machines de *Boltzman* permettent de résoudre les problèmes en temps réel. C'est une application particulière des réseaux utilisés en classification.

Le réseau de neurones est alors un composant d'un système de fusion qui permet une précision et des performances temporelles meilleures que les approches traditionnelles de ce type de problème.

L'algorithme génétique partage avec le réseau de neurones ces avantages de parallélisme et d'indépendance aux paramètres d'entrée [Dec92].

On trouve des applications dans le domaine du suivi de cibles, de surveillance de zones sensibles.

I.2.5. Conclusion.

Nous avons montré dans ce paragraphe, que l'utilisation de plusieurs types de capteurs permet d'avoir une précision accrue, une plus grande portée de détection, une fiabilité supérieure, et une acquisition plus rapide des données. Mais la façon de gérer ces données demeure une préoccupation.

Plusieurs méthodes sont proposées dans la littérature. Ces dernières sont adaptées et adaptables à de nombreux problèmes liés à la fusion de données en provenance de plusieurs sources différentes.

I.3. Représentation d'un environnement de navigation.

La problématique de la représentation d'un environnement de navigation a été abordée par de très nombreux travaux depuis le début des années 70. A cette époque, les grands espoirs posés sur les puissances de calcul sans cesse grandissantes des machines ont amené les chercheurs à considérer que le système mobile peut "facilement" construire une carte de son environnement de navigation. Si l'on connaît le modèle de la scène, il est possible de projeter dans le référentiel associé au système les obstacles qu'il doit éviter durant son déplacement (des murs, ou des objets statiques). Les recherches portaient alors principalement sur la planification de la trajectoire du système mobile dans cet espace particulier. Des capteurs tels que le sonar, le laser, ou un système

stéréoscopique, peuvent ensuite être utilisés pour contrôler le bon suivi de la trajectoire précalculée.

Malheureusement, la reconstruction d'un environnement n'est pas une tâche aisée. En effet, des erreurs de reconstruction peuvent être engendrées par divers facteurs (les mesures des capteurs sont bruitées, la modélisation du système mobile est rarement parfaite, le système mobile peut glisser sur le sol, etc.) [Dro02].

C'est pourquoi, depuis le début des années 90, de nombreuses approches proposent de s'affranchir de la reconstruction de la scène. Puisqu'il existe une relation directe entre la position du système mobile et les observations de ses capteurs extéroceptifs, un environnement peut directement être défini dans l'espace des observations des capteurs. Le modèle de la scène n'est alors plus explicitement nécessaire.

Ainsi, deux grandes familles de représentation d'un espace de navigation se distinguent :

- *les représentations métriques* : Elles décrivent explicitement la position "géométrique" des éléments de l'environnement [Nei99]
- *les représentations topologiques* : Elles sont basées sur des graphes représentant des informations de plus haut niveau comme certaines places caractéristiques de l'environnement (coins, croisement de deux couloirs, jonctions en T, etc.).

I.3.1. Représentation métrique.

Dans cette représentation, l'environnement est représenté par un ensemble d'objets auxquels sont associées des positions dans un espace métrique, généralement en deux dimensions. Elle peut être subdivisée en deux sous familles:

- Représentation purement géométrique qui utilise explicitement les positions "cartésiennes" des primitives cartographiques.
- Représentation par grille d'occupation, qui décrit les propriétés métriques par discrétisation de l'environnement en y ajoutant des informations d'incertitude.

I.3.1.1. Représentation géométrique.

C'est une représentation cartésienne de l'environnement où les coordonnées x , y voire z des amers (caractéristiques de l'environnement) sont décrites explicitement. Ces caractéristiques sont des points, des segments de droite, des rectangles, des polygones, etc., obtenues à partir des mesures et constituant une représentation des différents éléments de l'environnement.

Ces représentations sont obtenues soit à partir des mesures de capteurs télémétriques qui donnent la limite des obstacles, soit à partir des modèles établis à partir d'une image d'un capteur de vision qui donne une autre perception de l'environnement.

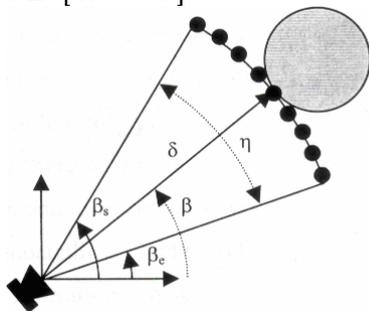
a. Par télémétrie.

Il existe différents filtrages et regroupements de points de mesure. Par exemple la méthode proposée dans [Cro89] consiste à prendre une par une les mesures d'un capteur à ultrasons tournant, 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 (*Régions 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.15).

b. Par vision.

Pour obtenir une représentation géométrique de l'environnement à partir d'un capteur de vision, il faut réaliser un prétraitement de l'image. Ce traitement consiste soit à extraire les contours des objets présents dans l'image grâce à un filtre de *Sobel* [Del97,Arr99] ou encore un filtre de *Deriche* [Mar99a], soit à établir une liste de segments à partir de ces points contours. En utilisant un capteur de vision stéréoscopique, il est possible d'obtenir, une carte composée de segments 3D [Wec98].



$$\beta = (\beta_e + \beta_s) / 2 \tag{I.28}$$

$$\eta = \beta_e - \beta_s \tag{I.29}$$

Figure I.15 : Création d'une RCD (Regions of Constant Depth).

I.3.1.2. Grille d'occupation.

Dans une grille d'occupation, les mesures sont représentées 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 [Mor85,Elf87b] ont été parmi les premiers à utiliser le principe des grilles d'incertitude (ou grille d'occupation). L'objectif de leurs travaux est de construire de manière autonome la carte de l'environnement d'un système mobile.

Pour cela, ce dernier 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 système.

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. Ces travaux ont été repris et étendus dans de nombreuses autres études telles que [Bor91c,Sch94].

L'avantage principal de la représentation métrique est de permettre de représenter l'ensemble de l'environnement. Cette représentation complète permet ainsi d'estimer avec précision et de manière continue la position d'un système mobile sur l'ensemble de son environnement. De plus, cette représentation complète ne se limite pas aux positions physiquement explorées, mais s'étend à toutes les zones que le système mobile a pu percevoir depuis les lieux qu'il a visités. Cette propriété permet la construction d'une carte plus exhaustive de l'environnement en un temps plus court. Un autre avantage de cette représentation est lié au fait que la position du système mobile est définie de manière non ambiguë par ses coordonnées au sein de l'espace dans lequel est représentée la carte.

L'inconvénient majeur de cette représentation est d'être relativement lourde d'un point de vue mémoire. La précision est évidemment limitée à la résolution de la grille. Précisons que cette résolution conditionne aussi le coût en temps de calcul: plus cette résolution sera forte, plus les temps de calcul seront importants. Un autre inconvénient de cette représentation est l'incertitude sur les paramètres des objets géométriques extraits dans la carte, par exemple sur leur position dans l'environnement. Ce type d'incertitude provient des erreurs de localisation du système mobile lors de la perception d'un objet donné, ou d'un bruit au niveau du capteur.

Globalement nous avons pu constater que ce type de représentation intègre un nombre limité de caractéristiques, comme par exemple la position par rapport à un objet géométrique, ou la notion d'état d'une cellule pour les grilles. Pour pouvoir gérer des caractéristiques de plus haut niveau, telles que la notion de coins, de jonctions, il est nécessaire d'utiliser un type de représentation plus descriptif : la représentation topologique.

I.3.2. Représentation topologique.

La représentation topologique permet de monter en niveau sémantique par rapport à la représentation précédente. Le principe est ici de décrire l'environnement sous forme de graphes qui représentent la connectivité entre des

lieux caractéristiques et prépondérants de l'environnement au sens de la navigation.

Ces lieux doivent correspondre à des places fortes permettant de localiser le système mobile de façon sûre, robuste et précise. Ces places sont généralement significatives d'une densité forte d'amers naturels observables. Ces configurations privilégiées de l'environnement pourront par exemple décrire des coins, des croisements de couloirs, des jonctions en T.

Les espaces libres sont utilisés comme lieu distinctif "*distinctive places*" dans [Kui91], les auteurs décrivent l'espace de manière hiérarchique en prenant des niveaux topologiques qui représentent la connectivité entre lieux distinctifs "*distinctive places*". Les arcs du graphe correspondent à la stratégie utilisée par le système mobile pour aller du noeud de départ au noeud d'arrivée de l'arc.

Les systèmes de vision sont des systèmes privilégiés pour construire des graphes de zones significatives au sens de la navigation. Des méthodes propres au traitement d'images permettant d'établir des corrélations fortes entre des images et une base de données pourront dans ce cadre être utilisées pour déterminer les places significatives. Ainsi l'analyse en composante principale est utilisée par Dudek dans [Dud00b] pour sélectionner des points d'intérêt dans des séquences d'images.

La vision omnidirectionnelle, particulièrement bien adaptée dans ce cas de figure, est aussi utilisée dans [Ulr00a] où un système de localisation reposant sur une représentation topologique est proposé. Là encore, la méthode repose sur la comparaison de l'image perçue par une caméra panoramique avec des images appartenant à une base de données représentatives des différents lieux de l'environnement.

Un avantage important de cette représentation est qu'elle ne requière pas de modèle métrique des capteurs pour fusionner les données proprioceptives et les perceptions au sein d'une représentation unifiée de l'environnement. Cela est avantageux pour deux raisons. D'une part, ces modèles métriques peuvent, comme nous l'avons vu, être difficiles à obtenir ou s'avérer peu fiables. D'autre part, le fait de ne pas fusionner les deux sources d'informations permet de séparer les influences des erreurs correspondantes.

En effet, l'estimation de la position d'objets, lorsque l'on utilise un modèle métrique, dépend à la fois des valeurs mesurées par les capteurs et de la position du système mobile. Une erreur sur la position d'un objet peut donc provenir des deux sources. Déterminer la contribution de chacune des sources peut être difficile. Dans la représentation topologique, au contraire, le bruit sur les mesures des capteurs influe principalement sur la reconnaissance des lieux, tandis que le bruit sur les données proprioceptives influe principalement sur la position associée à chaque lieu [Fil04].

Comme nous l'avons déjà mentionné, l'utilisation directe des perceptions sans modèle métrique empêche d'estimer ces données pour des positions non visitées. En conséquence, la représentation topologique nécessite en général une exploration très complète de l'environnement pour le représenter avec précision. En particulier, tous les lieux intéressants que l'on souhaite trouver dans la carte devront être visités au moins une fois au cours de la construction de la carte,

parce qu'ils ne peuvent pas être perçus à distance. Dans le cas où les lieux représentés sont des structures d'assez haut niveau (comme des couloirs ou des pièces), cela n'est pas gênant car ces lieux sont peu nombreux et une exploration exhaustive est donc relativement rapide. En revanche, dans les cartes topologiques représentant des lieux avec une assez grande densité spatiale, cela peut être un inconvénient, car l'exploration complète de l'environnement demandera un temps important [Fil04].

I.3.3. Conclusion.

Les deux types de représentation d'un environnement de navigation que nous avons présentés ont des avantages et des inconvénients. La représentation métrique dite probabiliste, est robuste et relativement efficace. En outre elle permet de répondre au problème de divergence lié à l'utilisation du filtrage de Kalman Étendu dans la représentation dite géométrique. En revanche elle possède le gros désavantage d'être lourde en temps de calcul et limitée en précision. L'autre problème de cette représentation est d'avoir des niveaux sémantiques faibles qui globalement se limitent à une certitude sur l'existence d'une cellule [Rem04].

La deuxième représentation quant à elle permet justement d'intégrer une description de plus haut niveau que dans la représentation métrique, ceci en intégrant la notion de places significatives au sens de la navigation. Ces lieux caractéristiques, ainsi que les liens entre ces lieux sont généralement gérés dans un graphe, d'où une combinatoire moindre et une optimisation du paramètre "mémoire" pour le stockage des données. En outre la quantification précise des contraintes géométriques entre ces places significatives, par exemple pour des petits déplacements du système mobile, s'avère être un problème [Rem04].

Nous pouvons finalement constater que ces deux représentations sont complémentaires. Il est donc intéressant de pouvoir associer ces deux types de représentation dans un module de génération de carte. Cette approche complémentaire, que l'on peut qualifier d'approche hybride, est de plus en plus intégrée dans des travaux tels que [Gas99, Van00]. Les résultats obtenus montrent clairement l'efficacité d'une telle approche.

I.4. Conclusion.

Nous avons présenté dans ce chapitre une vue d'ensemble des systèmes de perception utilisés dans le domaine de la robotique rendant les systèmes mobiles autonomes dans diverses applications où ils auront à se déplacer dans des environnements d'intérieur ou d'extérieur.

Deux familles de capteurs sont utilisées : proprioceptives et extéroceptives. Nous avons pu constater que celles-ci sont plus complémentaires que concurrentes. C'est la raison pour laquelle ces systèmes mobiles sont généralement munis de plusieurs types de capteurs.

L'association et la fusion de données en provenance de ces capteurs, permettront aux systèmes mobiles d'obtenir un modèle sensoriel robuste et hautement descriptif de leur environnement de navigation pour se localiser et établir un plan de navigation pour atteindre leur objectif.

Il existe différents modèles sensoriels (représentations, carte) de l'environnement. Nous avons montré qu'il existe principalement deux grandes familles de représentation d'un espace de navigation, les représentations métriques et topologiques.

Dans le chapitre suivant, nous présentons un état de l'art des méthodes de localisation et de navigation qui permettront à un système mobile de se localiser dans son environnement, ensuite, générer un chemin sans collision lui permettant d'atteindre son but.

Chapitre II

Localisation et navigation des systèmes de robotique mobile.

Le point fondamental pour un système mobile est sa capacité à réaliser de manière autonome un déplacement d'un point à un autre dans un environnement non parfaitement connu a priori. Ce problème, bien que largement étudié par les chercheurs depuis quelques décennies, ne peut encore être considéré comme résolu.

En effet, connaissant la carte locale de l'environnement proche du système mobile au moyen de la perception, comment ce dernier se localise dans la carte globale de l'environnement, et ensuite, se doter d'une capacité de définir un plan de navigation pour atteindre son objectif.

Dans ce chapitre, nous passons en revue les principaux travaux menés dans la localisation et la navigation des systèmes mobiles ainsi que les méthodes développées pour leur résolution.

II.1. Localisation des systèmes de robotique mobile.

La localisation est la détermination de la situation, position et orientation, du système mobile par rapport à un repère de référence. La localisation est absolue si le repère est fixé, lié à son environnement. La localisation est relative lorsque le repère est lié à une position précédente du système mobile. Nous pouvons aussi parler de localisation statique lorsque le calcul de position s'effectue à l'arrêt ou de localisation dynamique lorsque celle-ci est évaluée durant le mouvement.

Nous pouvons considérer trois grands systèmes de localisation [Dro02] :

- La localisation relative ou à l'estime (*Dead Reckoning* en anglais), basée sur l'utilisation des capteurs proprioceptifs,
- La localisation absolue, basée sur l'utilisation des capteurs extéroceptifs,
- La localisation dite *hybride*, basée sur l'utilisation conjointe des capteurs proprioceptifs et extéroceptifs.

II.1.1. La localisation relative.

Elle consiste à déterminer la position, l'orientation, et éventuellement la vitesse du système mobile par intégration des informations fournies par divers capteurs proprioceptifs depuis un point de départ. Ces données peuvent être des informations de déplacement (odomètre), de vitesse (vélocimètre) ou d'accélération (capteurs inertiels) [Ric04].

Ce type de localisation présente l'immense avantage d'être indépendant de l'environnement. Les seules erreurs qu'il peut générer sont celles dues à son mode de fonctionnement interne. Par contre, l'inconvénient majeur est l'accumulation d'erreurs due aux différentes intégrations.

II.1.1.1. L'odométrie.

L'odométrie permet de déterminer la position et l'orientation d'un système mobile par intégration de ses déplacements élémentaires, et ce, par rapport à un repère lié à sa configuration initiale.

La technique de localisation consiste à fixer sur les roues des codeurs délivrant une impulsion toutes les fractions de tour de roue. L'intégration de ces valeurs permet de déduire la position et l'orientation du système mobile par rapport à l'initialisation des compteurs d'impulsions.

Les avantages de cette technique résident dans sa simplicité de mise en œuvre et dans son faible coût. Ces caractéristiques en font la technique de localisation la plus couramment utilisée en robotique mobile. En outre, l'utilisation d'odomètres permet d'obtenir une estimation de la position et l'orientation à une cadence relativement élevée avec une précision qui reste bonne sur de faibles distances. Cependant, dès que les distances à parcourir augmentent et que la trajectoire comporte de nombreux changements de direction, l'imprécision sur la position et l'orientation du système mobile augmente considérablement, imprécision due à l'erreur cumulative générée lors de l'intégration des déplacements élémentaires [Ric04].

II.1.1.2. La vélocimétrie.

Cette méthode consiste à mesurer directement la vitesse du système mobile et à l'intégrer pour obtenir le déplacement. La vitesse de rotation instantanée, dans la majorité des cas, est obtenue avec des gyromètres. La vitesse linéaire est quant à elle généralement obtenue par l'utilisation d'un radar *Doppler* dirigé vers le sol [Ric04].

II.1.1.3. Les capteurs inertiels.

La localisation d'un système mobile en utilisant ce type de capteur (accéléromètres, gyroscopes, compas magnétiques) est déterminée à partir d'informations inertielles acquises au cours de son mouvement.

Cette localisation plus coûteuse constitue le haut de gamme des systèmes de localisation à l'estime. Le calcul de la position est effectué par double intégration de ces informations et entraîne une inévitable accumulation d'erreurs qui constitue une dérive d'estimation dans le temps. Un recalage périodique est alors indispensable.

II.1.1.4. Les capteurs d'images [Lau01].

Les capteurs optiques à corrélation exploitent des images du sol acquises généralement par une caméra CCD durant le déplacement d'un système mobile. En utilisant une technique de corrélation entre deux images consécutives de la séquence, il est possible de mesurer le mouvement apparent dans l'image, résultant de la projection du mouvement réel. Connaissant la position de la caméra par rapport au sol, il est alors possible de reconstruire une estimée du mouvement réel.

Pour souligner qu'il s'agit bien de mesurer des déplacements, cette méthode est souvent appelée *odométrie optique*; elle souffre également des dérives inhérentes à toute méthode qui intègre des déplacements élémentaires.

Ces capteurs ont d'excellentes performances dans les gammes de vitesse autour de zéro, ce qui les rend très utiles dans des problèmes de stabilisation en point fixe par rapport au sol (robotique sous-marine, engins aériens de surveillance).

Le principal inconvénient de cette méthode réside dans la nécessité d'avoir des images du sol suffisamment texturées pour permettre un bon fonctionnement de l'algorithme de corrélation; elle est surtout adaptée pour la localisation d'un système robotique mobile évoluant sur terrain naturel.

II.1.2. La localisation absolue.

La localisation absolue est une technique qui permet à un système mobile de se repérer directement dans son milieu d'évolution, que ce soit en environnement extérieur (mer, espace, terre), ou en environnement intérieur (ateliers, immeubles, centrales nucléaires) [Dro02]. Elle est basée sur l'utilisation de capteurs extéroceptifs et nécessite toujours une représentation de l'environnement. Le système mobile possède donc *une banque de données* regroupant les éléments caractéristiques de son milieu d'évolution (balises par exemple), et pour sa localisation absolue, il doit déduire de la perception de ces éléments caractéristiques, sa position dans son environnement.

II.1.2.1. Nature des balises [Dro02,Ric04].

a. Les balises actives.

Les balises actives (ou repères) artificielles sont ajoutées au milieu d'évolution du système mobile et dont les positions sont connues. Elles sont généralement interactives avec le système de perception du système mobile puisque ce sont elles qui émettent l'onde captée par l'équipement de mesure. Les deux principaux types d'organes de transmission utilisés sont les émetteurs de sources lumineuses (infrarouge par exemple) et les antennes émettrices hyperfréquences. Ces ondes peuvent être émises en continu ou être déclenchées par ordre provenant de l'équipement embarqué sur le mobile.

Leur inconvénient réside essentiellement dans leur manque de souplesse et dans leur lourdeur d'utilisation. En effet, un domaine d'évolution vaste nécessitera un investissement lourd en équipement. En outre tout changement de configuration de l'environnement impliquera une remise en cause du réseau de balises. En revanche, l'utilisation de ces balises a le gros avantage d'être précise, robuste et surtout de satisfaire la contrainte temps réel.

Un des systèmes les plus connus en utilisant ce type de balise, est le système GPS (§I.1.2.4). C'est un système de balises placées sur des satellites en orbite terrestre, utilisant les ondes radio émises de façon synchronisées dans le temps et qui est par conséquent accessible de quasiment partout à la surface du globe.

Par recoupement des instants d'arrivée des ondes ou signaux et de la position des satellites émetteurs, les récepteurs peuvent calculer leur position. Le principe de calcul de la position est basé sur une *triangulation*, à l'aide de quatre signaux

reçus simultanément (le quatrième signal assure la robustesse de la mesure). La position est calculée dans un repère global couvrant la terre avec une précision variant de quelques dizaines de mètres à quelques centimètres suivant les équipements [Fil04].

Un exemple d'utilisation de ce système est donné dans [Mar99a] 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 [Koo96]. 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 système mobile en comparant l'image actuelle de la caméra avec celles mémorisées lors d'une phase d'exploration.

L'inconvénient du GPS se situe au niveau des signaux satellites qui peuvent ne pas parvenir au récepteur à cause du relief du terrain et de l'environnement (hauts bâtiments et ponts en villes, relief montagneux, etc.). En solution à cette contrainte, on peut citer les travaux présentés dans [Cui01] pour localiser un véhicule dans un environnement urbain.

Notons qu'une source sonore peut aussi être une balise active [Gon95]. 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 système mobile peuvent être déduites.

b. Les balises passives.

On appelle balises passives des repères artificiels n'échangeant pas de signaux avec le système de perception. Comme précédemment ce type de méthode est basé sur le repérage de balises artificielles placées à des positions connues dans le milieu d'évolution du système mobile. La différence se situe au niveau de l'identification de la balise qui doit être faite par le système de perception. Ainsi le problème de la mise en correspondance est généralement facilité par un codage au niveau du marquage de la balise. Il est donc 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 système mobile [Cou00]. Ces systèmes de balises ont l'avantage d'être moins lourds à installer que les systèmes précédents.

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 [Kim97]. Une caméra CCD vise le panneau et l'algorithme proposé détermine la position du système mobile en analysant les déformations des bandes vues dans l'image (Figure II.1).

Une autre application, destinée à l'atterrissage automatique d'hélicoptère, utilise comme balise un cercle blanc peint sur le sol [Pol91]. 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.

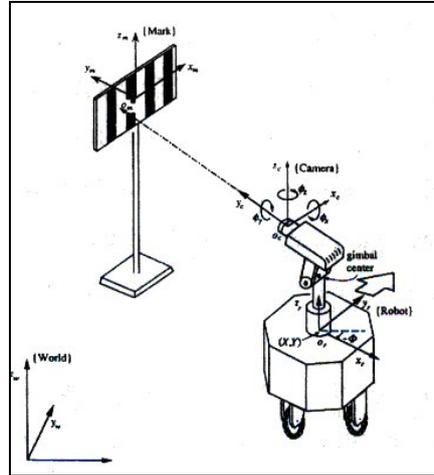


Figure II.1 : Estimation de la position d'un système mobile utilisant un repère artificiel (d'après [Kim97]).

En environnement intérieur, les repères peuvent être des réflecteurs placés sur le mur. La localisation du système mobile est alors réalisée par une triangulation pour laquelle les angles séparant le système des balises sont mesurés. Ces angles peuvent être mesurés à partir d'un capteur à ultrasons tournant et des réflecteurs acoustiques [Kwo97], ou d'un goniomètre laser et des réflecteurs de lumières [Gou94]. 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 système mobile. 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 système mobile en utilisant deux balises passives et un capteur à ultrasons tournant [Beo95].

Les balises peuvent aussi servir à corriger la position du système mobile lors de la navigation. Ce principe a été utilisé pour une application destinée à un système mobile sur autoroute [Kar97]. Dans cette application, une carte est créée au cours d'une phase d'exploration dans laquelle une personne conduit le système mobile. 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 système mobile.

c. Les repères naturels.

Cette méthode consiste à utiliser les éléments caractéristiques de l'environnement pour estimer la position du système mobile. On parle dans ce cas

d'amers. L'intérêt de ces méthodes est donc leur souplesse d'utilisation puisqu'elles ne nécessitent pas d'aménager le milieu d'évolution du système mobile. La fiabilité de ces méthodes n'est pas aussi grande que celle des balises passives puisqu'il faut réussir à qualifier les repères rencontrés pour les différencier les uns des autres [Cou00].

Parmi les travaux dans ce contexte, nous citons la méthode proposée dans [Mar99a]. Elle consiste à localiser un système 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. 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 système dans l'image étant représentée par un segment dirigé vers le haut, l'angle sous lequel le système perçoit un élément vertical de l'environnement est directement disponible. La position du système mobile et son orientation sont alors calculées par triangulation.

D'autres repères naturels ont été utilisés dans une application de positionnement dans un couloir qui permet à un système mobile de transporter les plateaux repas dans un milieu hospitalier [Hon97]. 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, etc.).

II.1.2.2. Principes de la Localisation [Ric04,Cou00].

a. Méthodes utilisant des balises.

Une fois les balises identifiées, la position du système mobile est généralement calculée par les deux méthodes suivantes :

- ◆ Par la mesure de distances ou trilatération.
- ◆ Par la mesure d'angles ou triangulation.

➤ Par la mesure de distances ou trilatération :

Si les distances qui séparent le système mobile de plusieurs balises actives sont connues, la position de ce dernier est déduite 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 système mobile (Figure II.2).

Soient $M(x_M, y_M)$ la position du système mobile et d_i la distance mesurée entre le système mobile 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 :

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

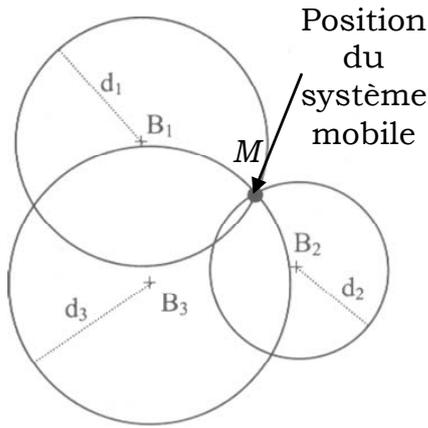


Figure II.2: Localisation par mesure de distances de trois balises.

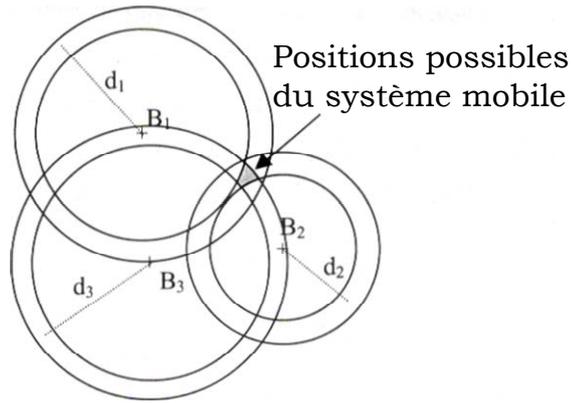


Figure II.3 : Localisation par mesure de distances incertaines de trois balises.

Si les erreurs de mesures sont prises en compte, il faut alors considérer des anneaux à la place des cercles. Et on obtient ainsi une zone à la place d'un point (Figure II.3).

➤ Par la mesure d'angles ou triangulation :

Au lieu d'employer les distances, cette méthode utilise les angles séparant l'axe du système mobile des directions sous lesquelles il perçoit les balises. Par triangulation on en déduit la position du système mobile (Figure II.4).

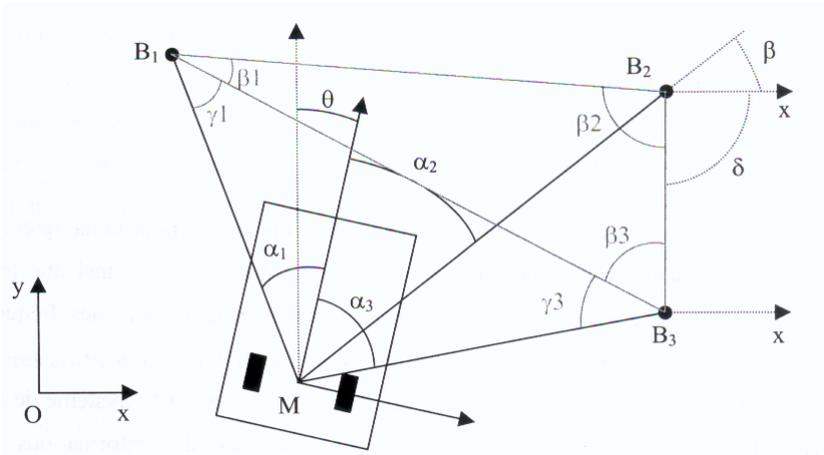


Figure II.4 : Localisation par triangulation.

Soit $M(x_M, y_M)$ la position du système mobile et θ son orientation. Soit α_i l'angle sous lequel le système 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 :

$$\frac{MB_2}{\sin(\beta_3 + \gamma_3)} = \frac{B_2 B_3}{\sin(B_3 M B_2)} \quad \text{et} \quad \frac{MB_2}{\sin(\beta_1 + \gamma_1)} = \frac{B_2 B_1}{\sin(B_1 M B_2)} \quad (\text{II.2})$$

en isolant MB_2 on obtient une relation où les seules inconnues sont γ_1 et γ_3 . En utilisant la relation :

$$B_3 MB_2 + B_1 MB_2 + \gamma_1 + \gamma_3 = \pi, \quad (\text{II.3})$$

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 :

$$\beta = B_3 MB_2 + \gamma_3 + \beta_3 - \delta \quad (\text{II.4})$$

et on obtient les coordonnées x_M et y_M du système mobile et son orientation θ par :

$$\begin{cases} x_M = x_{B2} - MB_2 \cos \beta \\ y_M = y_{B2} - MB_2 \sin \beta \\ \theta = -\frac{\pi}{2} + \beta + \alpha_2 \end{cases} \quad (\text{II.5})$$

b. Méthodes utilisant la mise en correspondance de modèles [Dro02,Ric04].

Ces méthodes de localisation consistent 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 système mobile depuis sa position, est appelé carte locale. Si une telle mise en correspondance (ou appariement) est trouvée, alors la position et l'orientation du système mobile peuvent être calculées.

Le modèle (ou carte globale) peut être pré-établi et stocké en mémoire dans certains cas. Dans d'autres cas il peut être inexistant au départ. C'est le système mobile qui le construit grâce à ses capteurs par exploration (*SLAM* ou localisation et modélisation simultanée). Nous reviendrons dans ce qui suit sur ce type de localisation.

Parmi les méthodes utilisant la mise en correspondance de modèles en robotique mobile, nous citons :

➤ Les méthodes de prédiction et vérification :

Ces méthodes consistent dans un premier temps à générer des hypothèses d'appariement par rapport à un historique sur les observations ou à des contraintes relatives à l'environnement. Dans un deuxième temps, ces hypothèses sont infirmées ou confirmées par une vérification des hypothèses par rapport aux observations. Ceci constitue le fondement des systèmes de localisation basés sur le Filtrage de Kalman Etendu (*EKF*) qui sont largement utilisés en robotique mobile. Dans ce cas l'odométrie permet de fournir une prédiction à partir de laquelle il est possible de générer un vecteur prédictif qui sera ensuite vérifié à l'aide des observations [Del98,Vic00].

➤ Les méthodes basées sur des arbres de recherche :

Le principe de la recherche dans un arbre d'interprétation nécessite dans un premier temps la construction d'un arbre représentant l'ensemble des appariements possibles entre les primitives du modèle et celles détectées par le système. La seconde étape consiste alors à trouver le chemin optimal en

profondeur dans cet arbre en déterminant à chaque niveau un appariement entre primitives en utilisant des contraintes ou des fonctions de coût.

L'utilisation de ces facteurs discriminants est justifiée afin de limiter le caractère combinatoire de cette technique. Un des précurseurs ayant utilisé cette approche pour la localisation est *Drumheller* [Dru87]. Des tests, prenant en compte les contraintes physiques des capteurs ultrasoniques ("*The sonar barrier test*"), permettent de générer un arbre de décision. *Simsarian* dans [Sma96], utilise un arbre de recherche pour effectuer la mise en correspondance. Il réduit le nombre de combinaison de la méthode en prenant avantage des *view-invariant regions*. Durant une phase de prétraitement, la carte est décomposée en régions ayant la propriété suivante : tous les points d'une même région voient le même ensemble d'amers.

➤ Les méthodes utilisant les invariants géométriques :

Ces méthodes caractérisent certaines configurations géométriques de primitives par une quantité qui reste inchangée aux points de vue et aux projections. Les valeurs de ces quantités, les invariants, sont directement utilisées pour effectuer la mise en correspondance. Des travaux ont été réalisés par rapport à cet aspect de la mise en correspondance [Mar99b]. Ce type de méthode de mise en correspondance est utilisé dans de nombreux travaux sur la localisation, comme par exemple dans [Dea00,Low01].

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

Le suivi multi-pistes consiste à effectuer une mise en correspondance temporelle entre deux observations consécutives. Elle peut alors être vue comme la possibilité de propager au cours des acquisitions une mise en correspondance initiale. La variation de configuration entre deux acquisitions consécutives étant généralement faible, l'appariement réalisé entre deux acquisitions consécutives permet d'obtenir une robustesse importante sur la mise en correspondance observations - primitives cartographiques. Le suivi multi-pistes est fortement utilisé dans le domaine militaire par exemple pour le suivi de missiles [Dro02].

c. Méthodes utilisant le principe d'indexation d'images.

Lorsque le système robotique mobile est équipé d'une seule caméra, des approches fondées sur la connaissance de l'environnement sous la forme d'une base d'images peuvent être mises en œuvre. La procédure de localisation consiste à rechercher l'image de la base la plus proche de l'image acquise depuis la position courante du système mobile; de ce fait, le résultat n'est pas une localisation dans un espace continu, mais plutôt dans l'espace discret des positions pour lesquelles une image a été apprise. Ces méthodes, de type force brute (corrélation, analyse en composantes principales sur des vecteurs de grande dimension, etc.), offrent une alternative à l'extraction de primitives-traitement peu robuste et dépendant du contexte [Lau01].

II.1.3. Méthodes hybrides [Dro02].

Les deux approches de localisation présentées précédemment (localisation à l'estime et localisation absolue) s'appuient soit sur des mesures de déplacement

relatif, soit sur des mesures de position absolue. Elles possèdent chacune des caractéristiques complémentaires.

Les avantages principaux de la localisation relative sont l'utilisation de capteurs proprioceptifs propres au système mobile, ne nécessitant aucune instrumentation de l'environnement. Les algorithmes utilisés sont simples (du point de vue de leur mise en œuvre) et fournissent, à une fréquence élevée, une estimée de la position du système mobile par rapport à un instant initial. Leur limitation réside dans leur principe même, à savoir l'intégration de mesures relatives, qui entraîne une dérive des estimées.

Les méthodes de localisation absolue s'appuient sur des données extéroceptives et nécessitent une instrumentation (cas des amers artificiels) ou une connaissance a priori (cas des amers naturels) de l'environnement. Elles sont souvent complexes au niveau algorithmique, mais fournissent des localisations absolues exemptes de dérive par rapport à un repère fixe de la scène. En raison de cette complexité, elles ne pourront fournir des estimées de la position du système mobile qu'à des cadences relativement faibles et nécessiteront souvent l'arrêt du système mobile durant l'acquisition des mesures [Lau01]

En développant des méthodes de localisation qui fusionnent les données proprioceptives et extéroceptives, il est possible d'allier les avantages des deux approches, tout en limitant leurs inconvénients. Le principe de ces méthodes consiste à corriger les positions estimées à cadence rapide au moyen des capteurs proprioceptifs par des mesures fournies à cadence beaucoup plus lente par les capteurs extéroceptifs. On parle alors de recalage dynamique, c'est-à-dire, les positions estimées par les capteurs proprioceptifs seront utilisées pour faciliter l'appariement des mesures fournies par les capteurs extéroceptifs.

Dans cette optique, nous proposons une synthèse non exhaustive des méthodes basées sur le recalage et celles basées sur l'association de plusieurs capteurs. C'est dans ce cadre général que des problématiques de fusion de données sont à traiter.

II.1.3.1. Le recalage dynamique.

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

Parmi les méthodes de fusion existantes, la plus communément utilisée est celle basée sur le filtre de *Kalman* étendu (EKF). Leonard et Durrant-Whyte dans [Leo92] ont été, après Crowley [Cro89], les premiers à utiliser le filtre de *Kalman* étendu pour localiser un système mobile dans son environnement par recalage dynamique. Une prédiction de la configuration du système fournie par l'odométrie est utilisée d'une part pour simplifier la mise en correspondance avec les

observations fournies par des capteurs à ultrasons et d'autre part pour linéariser l'équation de mesure. Les écarts entre les mesures et les prédictions sont utilisés pour estimer la position finale. Cette approche générique a été étendue par Durrant-Whyte à d'autres systèmes de perception extéroceptifs tels qu'un radar à ondes millimétriques dans [Dur95] ou à un télémètre laser embarqué sur un véhicule routier [Gui00a,Gui00b]. Cette dernière étude a porté sur une extension concernant une démarche de localisation et de modélisation simultanée. Dans [Vic00], un algorithme basé sur la fusion d'observations télémétriques issues d'un télémètre laser avec des données odométriques a été proposé. Une représentation sous forme de segments auxquels est associée une incertitude est obtenue à partir du relevé laser. Le processus de fusion se fait grâce à un filtre de *Kalman* étendu. *Virgil*, le système mobile destiné à faire une visite guidée de l'université de *Rice* aux Etats-Unis, fusionne les données issues d'un récepteur *GPS* avec les informations odométriques à l'aide d'un filtre de *Kalman* étendu [Thr01]. Le problème de l'obstruction des signaux *GPS* est notamment traité. La précision de la localisation atteint 40 cm. *Chung* utilise dans [Chu01] un filtrage de *Kalman* pour fusionner les informations issues d'un gyroscope à fibre optique et d'un odomètre. Une calibration soignée du gyroscope a été réalisée, notamment au niveau de l'erreur due à la dérive en température du capteur. Au final, il s'avère que la précision de localisation, qui est très bonne, est due d'une part à la calibration méticuleuse du gyroscope et d'autre part à son couplage avec un odomètre.

Le filtre de *Kalman* étendu est l'outil le plus couramment utilisé dans les processus de recalage dynamique pour sa relative simplicité d'utilisation et pour son efficacité. En revanche son gros inconvénient se situe au niveau de l'étape de linéarisation, qui, si l'estimation odométrique est éloignée de la position réelle du système mobile, fera diverger le filtre [Dro02].

II.1.3.2. La coopération multi-capteurs.

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

De nombreux travaux utilisent la coopération de capteurs pour contribuer aux problématiques telle que la localisation. Dans [Nei99], l'auteur ajoute à l'information odométrique classique, un système associant une image de profondeur et une image d'intensité, toutes deux issues d'un capteur laser. Cependant, il fait une utilisation limitée des deux images puisqu'il ne se sert de l'image d'intensité que pour affiner la position des coins détectés avec l'image de profondeur. Un filtre de *Kalman* achève ensuite la localisation du système mobile. La méthode s'avère précise puisque, à la suite des résultats expérimentaux, l'auteur aboutit à une erreur d'environ 10 cm en position et 0.15 degrés en orientation.

Dans [Arr99], les auteurs utilisent la coopération entre un télémètre laser et un système de vision monoculaire pour localiser leur système mobile *Pygmalion*. Une segmentation du relevé laser lui permet d'obtenir un ensemble de primitives de type segment. L'application de la transformée de *Hough* sur l'image acquise fournit les angles des amers verticaux devant le système. La carte théorique de l'environnement est composée d'une liste de segments et d'une liste d'amers verticaux. L'appariement des deux types d'objets est réalisé grâce au filtre de *Kalman* étendu.

Dans [Lal98], le système mobile utilise conjointement un télémètre laser et une caméra monoculaire CCD orientable pour la détection de coin dans la scène. Ensuite, un appariement de la carte théorique de l'environnement avec le modèle sensoriel ainsi obtenu est réalisé en utilisant la notion d'invariant.

II.1.4. Localisation et modélisation simultanées de l'environnement (SLAM) [Dro02,Ric04].

Lorsque la carte de l'environnement du système mobile n'est pas connue a priori, un module de génération de carte incrémentale doit obligatoirement être intégré au système de navigation. Il s'agit là d'une étape supplémentaire qui consiste à mettre à jour au fur et à mesure des acquisitions, une représentation du milieu d'évolution du système mobile. Ce processus se décline comme une fusion successive des différents modèles sensoriels générés lors du déplacement du système.

Il paraît clair à ce niveau que la fusion incrémentale de modèles sensoriels nécessite obligatoirement leurs recalages systématiques avec les primitives déjà insérées dans la carte. Cette étape de recalage est ni plus ni moins qu'une localisation du système mobile par rapport à une connaissance acquise au cours du déplacement [Dro02]. L'étape de localisation devient alors indissociable de celle de modélisation et on parle alors d'un système de *localisation et modélisation simultanée* (en anglais, *Simultaneous Localization and Mapping* ou *SLAM*), voire plus rarement *CML* (*Concurrent Mapping and Localization*).

L'implémentation d'un algorithme *SLAM* repose sur les étapes suivantes :

- ◆ **Perception** : les algorithmes de perception fournissent à partir des données issues de capteurs extéroceptifs un ensemble d'observations des éléments de l'environnement (amers), dont la position relative au système mobile peut être déterminée. Ce processus de perception dépend bien entendu du type d'environnement dans lequel le système mobile évolue et des capteurs utilisés.

- ◆ **Association de données** : les amers ne sont utiles pour estimer la position du système mobile que s'ils sont observés de différentes positions. Le système mobile doit décider si l'amer observé est déjà présent dans la carte, et de quel amer il s'agit, ou bien alors si c'est un nouvel amer à ajouter à la carte.

- ◆ **Proprioception** : Les données proprioceptives sont des mesures internes au système, et renseignent sur l'évolution de son état. Pour le *SLAM*, on s'intéresse à la position du système mobile dont l'évolution est typiquement fournie par des capteurs odométriques ou une centrale inertielle.

♦ **Estimation** : L'algorithme d'estimation doit intégrer les données issues de la perception ainsi que de la proprioception afin de fournir une estimée de la position du système mobile et des positions des amers, ainsi que les incertitudes associées à ces estimées.

♦ **Gestion de la carte** : Il s'agit d'intégrer ou bien fusionner toutes les données issues de la perception afin de fournir une carte globale cohérente de l'environnement.

Autrement dit, lorsqu'un système mobile se déplace dans un environnement, les amers détectés sont positionnés avec une imprécision due aux capteurs embarqués. Le système mobile mesure son déplacement par une technique de localisation à l'estime avec une certaine erreur. Les amers qui sont à nouveau observés, après une mise en correspondance, sont associés aux amers préalablement perçus. Une technique de fusion de données permet alors de réduire les erreurs sur la position courante du système mobile et les positions des amers. Ce processus est itéré à chaque nouvelle perception de l'environnement effectuée par le système mobile.

II.1.5. Conclusion.

Nous avons vu au cours de ce paragraphe qu'il existe une grande variété de méthodes de localisation qui diffèrent par les capteurs utilisés. Les choix sont établis en fonction des applications auxquelles elles sont dédiées.

La localisation à l'estime est utilisable en environnement extérieur ou intérieur mais la distance de déplacement doit être réduite pour limiter les accumulations d'erreurs. La technique par balisage est employée indifféremment à l'intérieur et à l'extérieur mais contraint le système mobile à rester dans une zone prédéfinie. La mise en correspondance de modèles nécessite la connaissance exacte de la topologie de l'environnement. La localisation sans connaissance a priori nécessite d'intégrer un module de construction incrémentale de carte.

Nous avons pu remarquer que l'étape de localisation est indissociable de celle de mise à jour. En d'autres termes, toute erreur sur l'estimation de la configuration du système mobile aura une influence sur celle commise lors de la mise à jour. Cette forte interaction est d'ailleurs à la base du constat, souvent fait dans les travaux portant sur cette problématique, qui souligne les phénomènes d'erreurs cumulatives sur des longs trajets. Une robustesse importante du paradigme de localisation et de modélisation passera donc par une gestion la plus rigoureuse possible de ces imprécisions.

L'idée maîtresse qui guide la robotique autonome consiste à dire que le système mobile construit sa carte ou modèle de son environnement au fur et à mesure de l'exploitation lui permettant d'évoluer à terme dans des lieux inconnus à l'avance.

II.2. Navigation des systèmes de robotique mobile.

La navigation des systèmes de robotique mobile consiste, de la manière la plus générale, à donner à ces systèmes la capacité d'aller d'une position initiale à une position finale de manière autonome. Historiquement, la navigation de ces systèmes mobiles a tout d'abord été considérée comme un problème de planification de trajectoires. Le modèle de la scène est supposé connu et des

stratégies globales ou locales permettent de définir la séquence de configurations que doit prendre le système mobile pour atteindre son objectif.

Depuis le début des années 80, les avancées technologiques et algorithmiques permettent de contrôler directement les mouvements d'un système robotique mobile à partir des informations perçues par ses capteurs. Ces informations sont utilisées pour définir une loi de commande du système mobile sans pour autant nécessiter le modèle de la scène.

Plus récemment, les recherches en robotique mobile ont considéré des environnements de navigation très vastes, à tel point que les capteurs embarqués (caméra ou même laser) ne peuvent les décrire que localement à un instant donné. Une représentation interne de l'environnement de navigation devient alors nécessaire. Elle permet dans ce cas d'étendre le champ d'action des capteurs du système robotique, mais aussi de définir un ensemble de stratégies de navigation adaptées à l'environnement traversé.

II.2.1. Les difficultés de la navigation [Bon03].

a. Le mouvement.

En robotique mobile, il existe deux grandes catégories de systèmes mobiles à roues :

➤ **Les systèmes non holonomes :**

Les systèmes mobiles dit non holonomes sont ceux que l'on rencontre le plus dans la vie courante (voiture particulière, bus, camion, etc.). Ces systèmes ont une structure mécanique relativement simple (des roues motrices, des roues directrices et des roues porteuses). Une roue peut avoir une, deux ou les trois fonctions. Mais tous ces systèmes ont une caractéristique commune: la direction de la vitesse d'entraînement (ou vitesse linéaire) est imposée par la direction des roues directrices.

Pour fixer les idées prenons un exemple: pour qu'un système de type voiture aille de sa position initiale à un mètre sur sa droite il est obligé de faire une manoeuvre : une marche avant puis une marche arrière.

➤ **Les systèmes holonomes :**

Ces systèmes sont beaucoup plus rares dans notre vie quotidienne. Ils ont une structure mécanique complexe qui leur permet de se déplacer dans toutes les directions sans manoeuvre.

Il existe, néanmoins, un exemple commun (et bien pratique): le chariot de magasin. Vous pouvez prendre un chariot et le tirer sur la droite, les roues s'orientent alors dans la bonne direction. Ce chariot n'a pas une structure mécanique compliquée mais ses articulations sont passives. La société *Nomadic*, disparue en l'an 2000, a conçu un système mobile holonome: le *XR4000*. Il dispose de 4 roues motrices et directrices montées comme des roues de chariot. La synchronisation des 8 axes (2 par roue, rotation et orientation) est assurée par une carte dédiée basée sur le microcontrôleur *Motorola 68332* et des circuits *FPGAs* et la structure mécanique est composée d'engrenages coniques [Hol00].

➤ **Robot holonome ou non holonome : les différences.**

La relative simplicité mécanique des systèmes non holonomes face aux systèmes holonomes a un prix. Les algorithmes d'exécution de trajectoire, de commandes et de planification de mouvements sont plus complexes. Mais leur facilité de construction en fait des systèmes très répandus dont l'automatisation présente un intérêt socio-économique évident.

b. La localisation.

Quel que soit le type de système mobile, holonome ou pas, la connaissance de sa position dans son environnement (son espace de travail) est importante pour s'y déplacer.

La localisation d'un système mobile dans son espace de travail est un thème de recherche très actif. La difficulté de connaître la position exacte d'un système mobile vient de l'impossibilité de mesurer précisément ses déplacements.

Différentes techniques existent (§II.2), parmi elles celles qui tentent de pallier les défauts de telle ou telle méthode en fusionnant les données issues de plusieurs capteurs. Ces dernières sont difficiles à mettre oeuvre car il faut prendre en compte les incertitudes de chaque mesure.

Quoi qu'il en soit, les algorithmes de navigation doivent, par une méthode ou une autre, prendre en considération le fait que le système mobile ne connaît jamais sa position avec exactitude [Bon03].

c. l'exécution du mouvement.

Pour naviguer dans son environnement, un système mobile doit pouvoir le représenter. Le système mobile peut utiliser une carte globale de son environnement pour planifier une trajectoire. Cette carte peut-être complète ou partielle, construite par le système mobile ou par l'homme. Le système pourra la mettre à jour pendant ses missions.

Dans la grande majorité des cas, la trajectoire créée dans un environnement modélisé ne peut être exécutée, ainsi, sans collision. Il existe toujours des artefacts de mesure ou des imprécisions que le système mobile doit être capable de prendre en compte pendant l'exécution du mouvement.

En conclusion, le système mobile doit, pendant son déplacement, prendre en compte son environnement. Il doit le faire pour plusieurs raisons :

- compenser les erreurs de localisation,
- compenser les imprécisions de modélisation de l'environnement initial,
- éviter les collisions avec les obstacles inconnus ou mobiles.

De nombreuses méthodes ont été étudiées pour permettre à un système mobile (holonome ou non holonome) de se déplacer sans collision dans un environnement partiellement connu ou inconnu. Ces méthodes se divisent en deux classes: les approches locales et les approches globales. Ces dernières sont basées sur une connaissance complète de l'environnement, tandis que les premières se basent sur une évaluation de l'espace immédiatement proche du système mobile.

II.2.2. Les approches locales.

Par définition, les approches locales ou réactives n'utilisent que les valeurs courantes des capteurs (ou des valeurs sur une petite fenêtre temporelle), et non des données provenant d'un modèle interne, pour décider de l'action à effectuer. Elles sont généralement exécutées très rapidement et elles permettent de réaliser des tâches de bas niveau, comme l'évitement d'obstacles imprévus, essentiel à la sécurité d'un système mobile.

Dans ce type d'approche, la perception est indispensable car on considère que le système mobile évolue dans un environnement qui lui est partiellement connu.

Parmi les approches locales, nous citons :

II.2.2.1. Les algorithmes *Bug*.

En se basant uniquement sur les informations sensorielles, les algorithmes *Bug* ont été développés pour résoudre le problème des labyrinthes sans aucune connaissance a priori sur la distribution ou la géométrie des obstacles.

Dans ces algorithmes, le système mobile se déplace vers l'objectif jusqu'à ce qu'il trouve un obstacle. Il contourne ce dernier jusqu'à ce qu'il soit en mesure de se déplacer à nouveau directement vers l'objectif (Figure II.5). En plus, si ces algorithmes maintiennent l'historique des lieux visités, le temps d'exécution diminuera considérablement.

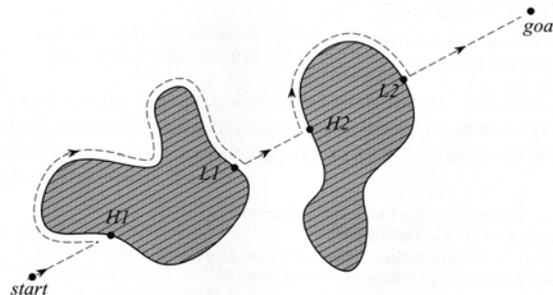


Figure II.5: Un exemple de navigation utilisant un algorithme *Bug* [Lum90].

Ces algorithmes ne prennent pas en compte la cinématique du système mobile et la dynamique des contraintes, de plus ils s'appuient uniquement sur la dernière lecture sensorielle, cela a un impact négatif sur les applications de ce type d'algorithme dans le monde réel [Sie04].

En général, ces algorithmes trouvent leur intérêt dans le contrôle des systèmes mobiles simples, cependant, ils ne sont pas assez puissants pour assurer une optimisation de la trajectoire.

II.2.2.2. Les approches basées sur les champs de potentiel.

Les approches basées sur les champs de potentiel ont été proposées pour produire des trajectoires douces à la fois pour les systèmes mobiles et les manipulateurs [Kha86]. Elles consistent à assimiler le système mobile à une particule contrainte à se déplacer dans un champ de potentiel fictif. Les obstacles exercent des forces répulsives sur le système mobile, tandis que le but à atteindre exerce sur lui une force attractive. Ensuite, la force résultante (la somme de toutes les forces) détermine la direction et la vitesse du mouvement (Figure II.6).

Ces approches peuvent être utilisées pour l'évitement des obstacles car elles n'agissent sur le système mobile que quand il en est très proche et a tendance à y pénétrer, et sont simples à mettre en œuvre et rapides, c'est les principales causes de leur grande popularité.

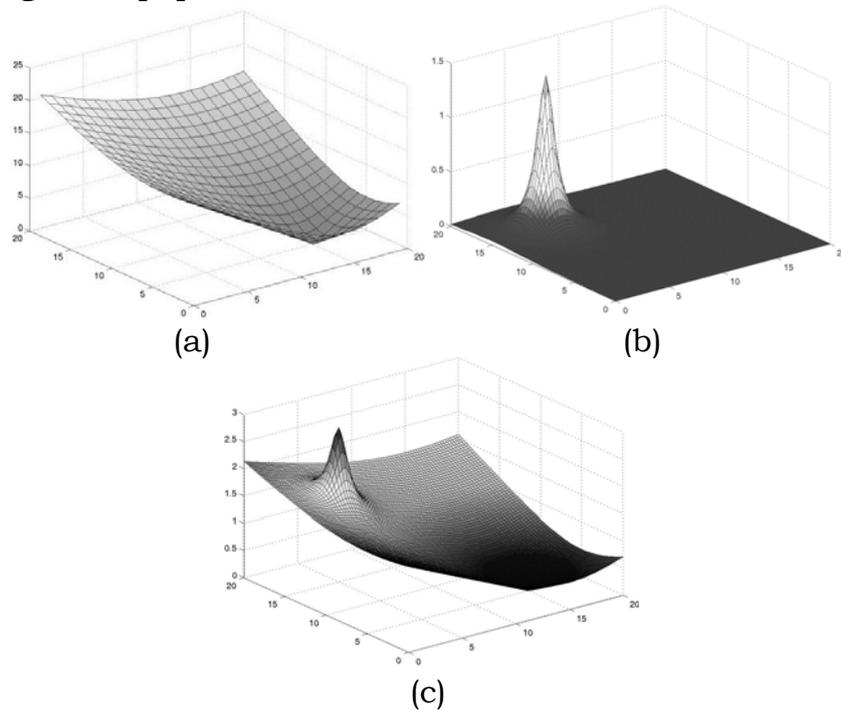


Figure II.6: (a) Le champ de potentiel du but, (b) Les champs de potentiel pour les obstacles, (c) Combinaison des deux champs précédents.

Toutefois, ces approches ont leurs propres inconvénients; *Koren et Borenstein* [Kor90] ont identifié certains problèmes qui sont inhérents à l'application des champs de potentiel, et qui sont:

- ◆ *Position piège en raison des minima locaux*: le système mobile peut être dans une position où le champ de potentiel est nul, ce qui signifie que le système mobile est piégé dans un minimum local. Ce problème peut être traité de différentes façons. Il est par exemple possible de déclencher un comportement particulier lorsque l'on rencontre un tel minimum : déplacement aléatoire, suivi de murs, etc., ou par une remonte (marche arrière) globale, c'est-à-dire le système mobile doit reconnaître la situation et remonter en arrière en empruntant le chemin parcouru jusqu'à ce qu'il trouve un nouveau chemin.
- ◆ *Pas de passage entre deux obstacles étroits*: Ce problème est similaire au précédent, si deux obstacles sont placés à proximité, comme une porte, les forces de répulsion de chaque obstacle sont combinées en une seule force de répulsion qui pointe hors de l'ouverture entre les obstacles. Le système mobile déviara alors de l'ouverture, même si le but est de l'autre côté.
- ◆ *Oscillations en présence d'obstacles et dans des passages étroits*: Un système mobile peut parfois osciller en présence d'obstacles, ceci pourrait par exemple se manifester lorsqu'il se déplace dans un couloir étroit.

II.2.2.3. La méthode Vector Field Histogram (VFH) [Bor91c].

La méthode *Vector Field Histogram* (ou *VFH*) est une version améliorée de la méthode *Virtual Force Field* (ou *VFF*) présentée dans [Kor88]. Elle utilise une grille d'occupation locale construite à partir des données issues des capteurs à ultrasons. Cette grille est construite de manière très rapide par la méthode dite *Histogrammic In Motion Mapping* (ou *HIMM*) où chaque cellule contient le nombre de fois qu'un obstacle est perçu à cet endroit (Figure II.7).

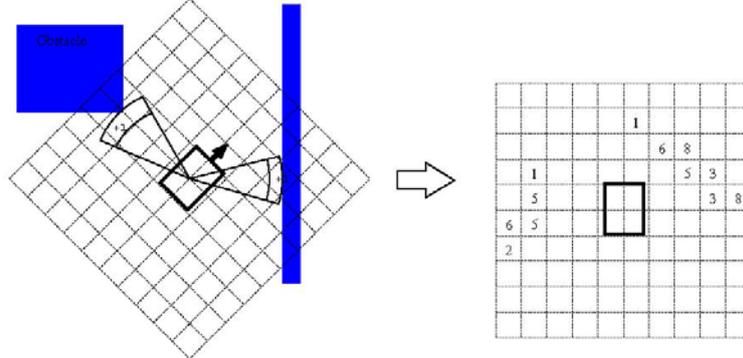


Figure II.7: Grille d'occupation locale construite par la méthode *HIMM*. La grille est construite dans le référentiel du système mobile [Bor91c].

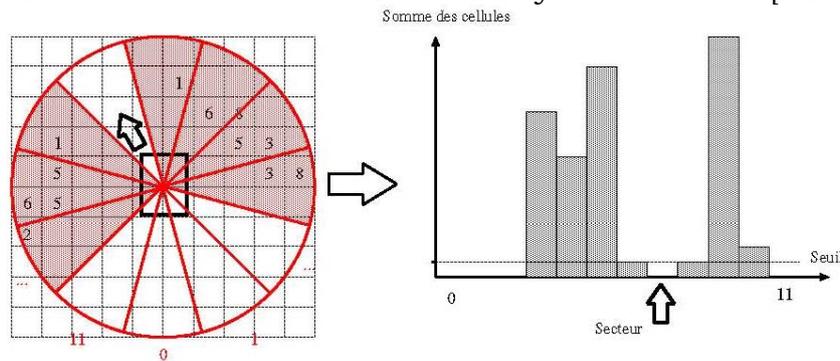


Figure II.8: Calcul de la direction de déplacement basée sur l'histogramme des obstacles [Bor91c].

Un histogramme représentant l'occupation de l'environnement autour du système mobile est ensuite construit à partir de cette grille d'occupation locale. Pour cela, l'environnement est discrétisé en secteurs angulaires contenant chacun la somme des valeurs des cellules correspondantes (Figure II.8). Un seuil permettant de tolérer un certain bruit est ensuite utilisé pour déterminer les directions possibles pour le système mobile: toutes les directions dont la valeur est inférieure au seuil sont prises en compte.

Le choix de la direction est finalement réalisé parmi les directions possibles en fonction de contraintes externes (par exemple la direction la plus proche de la direction du but).

Il existe, cependant, de graves problèmes avec cette méthode.

- ◆ *Choix des paramètres* : Dans cette méthode le choix du seuil qui détermine quand un secteur angulaire est choisi pour la navigation, va permettre de définir le comportement du système mobile. Une valeur trop faible de ce seuil rendra le système mobile incapable de se déplacer en présence d'obstacles, tandis qu'avec une valeur trop grande le système mobile heurte les obstacles.

♦ *Contraintes dynamiques* : Par exemple, un système mobile en mouvement à grande vitesse ne peut pas faire un virage de 90 degrés. Même s'il ne dérape pas, il déviara de la trajectoire déjà calculée. Il ne peut pas, non plus, s'arrêter, si un obstacle apparaît soudainement en face de lui. Les lois de la dynamique gardent le système mobile en mouvement même si les roues s'arrêtent de tourner.

Dans [Ulr98], la méthode VFH a été améliorée par la prise en compte de cette dynamique, et a donné naissance à une nouvelle méthode appelée VFH⁺.

II.2.2.4. Les approches d'optimisation de contraintes (Constrained optimization approaches).

Dans ce qui suit, nous présentons les principales méthodes d'optimisation de contraintes, les plus efficaces pour l'évitement d'obstacles.

a. L'approche basée sur la courbure de la vitesse ou CVA (Curvature Velocity Approach) [Smm96].

La méthode CVA a introduit le concept d'espace de vitesse. Dans ce dernier, une action est généralement décrite par le couple (v, ω) , avec v et ω les vitesses linéaire et angulaire décrivant une trajectoire de forme arc de cercle. Les couples qui auraient pour conséquence une collision avec des obstacles seront exclus de cet espace. Les obstacles sont stockés dans une grille ou plan cartésien, ensuite ils seront transformés dans cet espace de vitesse.

L'espace de vitesse ne considère que les couples de vitesse respectant la dynamique du système mobile. Enfin, à partir des ces couples, le meilleur est sélectionné maximisant une fonction appelée fonction objective linéaire combinant la vitesse, la distance des obstacles le long de l'arc généré et l'erreur de cap pour un cap désiré.

Les principales limitations de la méthode CVA sont :

- ♦ Un temps de calcul supplémentaire exigé par l'utilisation du plan cartésien.
- ♦ La description de l'action sous forme de couple (v, ω) de vitesses limite l'ensemble des possibilités pour les différents types de locomotion; par exemple, il n'est pas possible de décrire avec précision l'action pour les systèmes mobiles de type Ackerman et omnidirectionnel³.
- ♦ La fonction objective doit être paramétrée de manière à intégrer l'évitement d'obstacles en se basant sur le comportement du système mobile lorsqu'il s'oriente vers son but (goal oriented behavior). Comme conséquence, le comportement global du système mobile est difficilement prévisible.
- ♦ En termes d'optimisation de la trajectoire, la méthode CVA possède quelques problèmes connus, spécialement parce qu'elle ne considère pas la connectivité des espaces libres (free space connectivity), elle ne considère que l'espace libre le long de l'arc de cercle.

³ Avec le type Ackerman, les vitesses linéaire et angulaire sont à la fois couplées pour avoir l'angle de déviation (ou de direction). Les systèmes mobiles de type omnidirectionnel nécessitent un degré de liberté en plus, qui définit la direction du mouvement de façon indépendante aux vitesses linéaire et angulaire.

- ◆ Avec cette méthode, le système mobile s'approche trop près des obstacles avant de les éviter et rentre presque difficilement dans un couloir étroit [Kko98].

b. La méthode basée sur la courbure de voie ou LCM (Lane Curvature Method).

La méthode LCM a été proposée par [Kko98] pour surmonter certains inconvénients de la méthode CVA. Cette méthode divise l'environnement en voies libres. La voie qui permet d'optimiser le chemin le long d'une direction donnée est choisie. Ensuite, une direction locale pour rentrer et suivre la meilleure voie est calculée. Les limitations physiques du système mobile et les contraintes de l'environnement sont prises en compte dans cette méthode, ainsi que la connectivité de l'espace libre.

Notons que la méthode LCM souligne l'importance de traiter directement avec l'espace de travail pour faire face aux caractéristiques de l'environnement, tels que le passage dans un endroit étroit. Toutefois, à l'exclusion de cet avantage, cette méthode partage les mêmes problèmes que la méthode CVA, les bonnes performances de ces deux méthodes augmentent leur complexité et leur charge de calcul.

c. La méthode de la fenêtre dynamique ou DWA (Dynamic Window Approach) [Fox97].

La méthode DWA améliore la méthode CVA en considérant, en plus la vitesse du système mobile, ses capacités d'accélération (accélérations admissibles par le système mobile). Connaissant la vitesse du système mobile, l'algorithme DWA détermine une fenêtre dynamique à partir des couples de vitesse (v, ω) atteignables à la prochaine période d'échantillonnage en tenant compte de ses capacités d'accélération (Figure II.9).

Les couples qui auraient pour conséquence une collision avec des obstacles seront éliminés. Enfin, une fonction objective similaire à celle utilisée par la méthode CVA est appliquée pour déterminer le meilleur couple. Cette fonction est en général la somme de plusieurs termes. Ces termes peuvent exprimer une préférence a priori sur les vitesses, une préférence pour les trajectoires s'éloignant le plus des obstacles, ou une préférence de direction si l'on dispose par exemple d'une estimation de la direction d'un but à long terme. Le couple de vitesses minimisant cette fonction au sein de la fenêtre dynamique est alors sélectionné garantissant un déplacement sans collisions avec les obstacles.

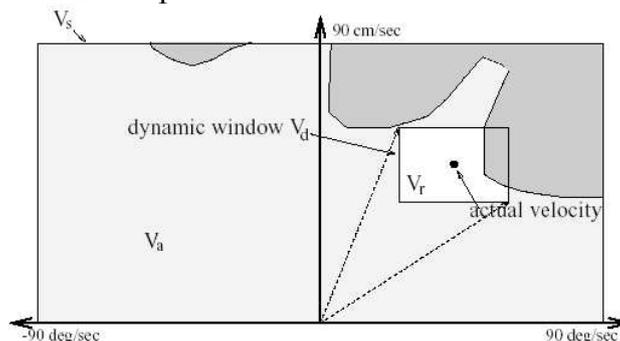


Figure II.9: Fenêtre Dynamique.

Avec la fenêtre dynamique, le système mobile permet de produire un déplacement sûr et régulier, son utilisation est très intéressante pour un système mobile se déplaçant rapidement, ou ayant des capacités d'accélération et de ralentissement limitées. Pour des systèmes qui ont une forte capacité d'accélération et de décélération, on peut considérer que toutes les vitesses sont accessibles presque instantanément. Il peut alors être suffisant de ne considérer que la cinématique, et non la dynamique du système mobile, ce qui se traduit par la prise en compte d'un seul point du graphe, et non d'une fenêtre. La recherche du couple de vitesse est ainsi simplifiée.

II.2.2.5. La méthode Nearness Diagram ou ND [Laf06].

L'objectif principal de la méthode *Nearness Diagram (ND)* est d'identifier la situation dans laquelle se trouve le système mobile parmi cinq situations prédéfinies et de générer la commande adéquate.

A partir des données capteurs du système mobile, la méthode *ND* construit deux diagrammes sous forme d'histogramme: *PND*⁴ et *RND*⁵. *PND* représente la distribution des obstacles autour du système mobile. *RND* quant à lui, représente cette distribution par rapport à une zone de sécurité définie autour du système mobile. La Figure II.10 illustre l'histogramme *PND* construit selon un environnement donné. Chaque "barre" représente un secteur et sa hauteur correspond à la proximité de l'obstacle.

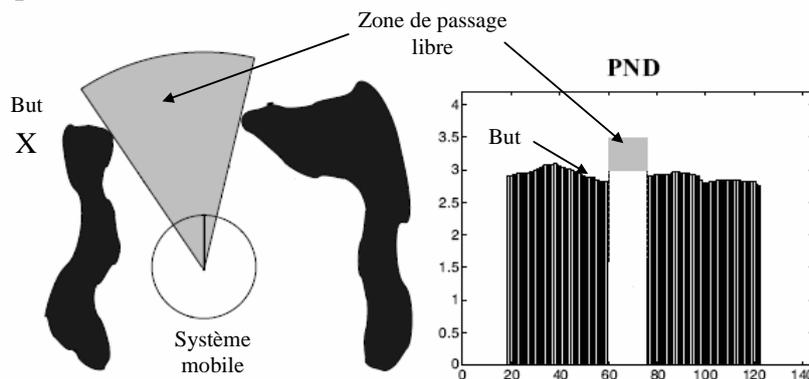


Figure II.10: Construction du diagramme *PND* selon un environnement donné.

Dans cette figure, apparaît la zone de sécurité autour du système mobile. *RND* est construit comme *PND* mais son rôle est différent. Il permet en effet de déterminer si un obstacle pénètre la zone de sécurité (Figure II.11).

Grâce aux diagrammes *RND* et *PND*, la méthode *ND* peut alors analyser la situation dans laquelle le système mobile se trouve et la reconnaître parmi les cinq prédéfinies. Ces situations sont les suivantes:

- 1) *LS1 (Low Safety 1)* : au moins un obstacle est présent dans la zone de sécurité mais d'un seul coté de la région libre la plus orientée vers le but (Figure II.11 et II.12a).
- 2) *LS2 (Low Safety 2)* : plusieurs obstacles sont présents dans la zone de sécurité mais des deux cotés de la zone de passage (Figure II.12b).

⁴ PND : Point Nearness Diagram.

⁵ RND : Robot Nearness Diagram.

- 3) *HSGR (High Safety Goal in Region)* : aucun obstacle ne pénètre la zone de sécurité et le but à rejoindre est dans une zone de passage (Figure II.13a).
- 4) *HSWR (High Safety Wide Region)* : aucun obstacle n'est présent dans la zone de sécurité du système mobile mais le but n'est pas directement atteignable. De plus, la région libre pour se déplacer est large i.e. représente plus d'un quart des secteurs du diagramme ($> 90^\circ$) (Figure II.13b).
- 5) *HSNR (High Safety Narrow Region)* : idem à la situation *HSWR* sauf que la région libre représente moins d'un quart des secteurs du diagramme ($< 90^\circ$) (Figure II.13c).

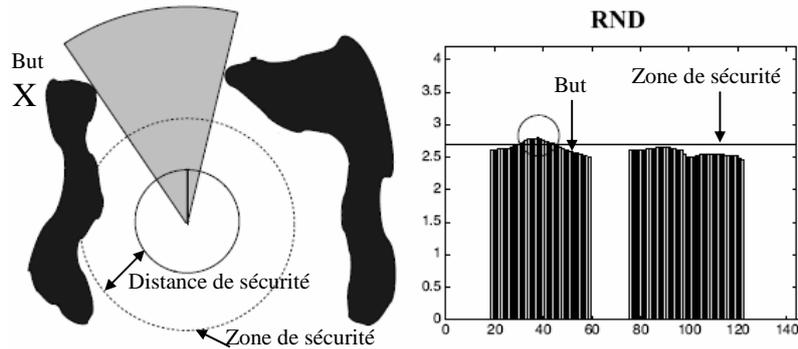


Figure II.11: Construction du diagramme *RND* selon un environnement donné, un obstacle pénètre la zone de sécurité.

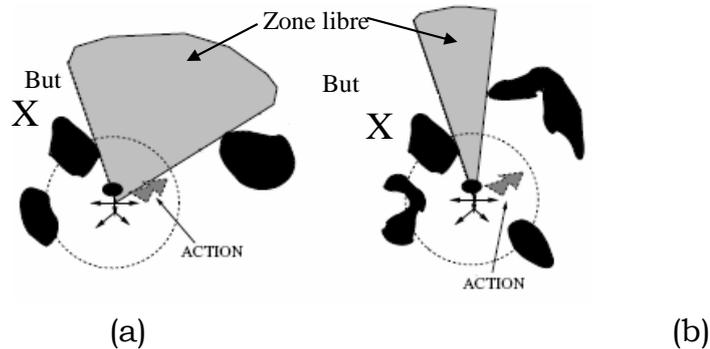


Figure II.12: Types d'actions menées dans les situations *LS1* et *LS2*.

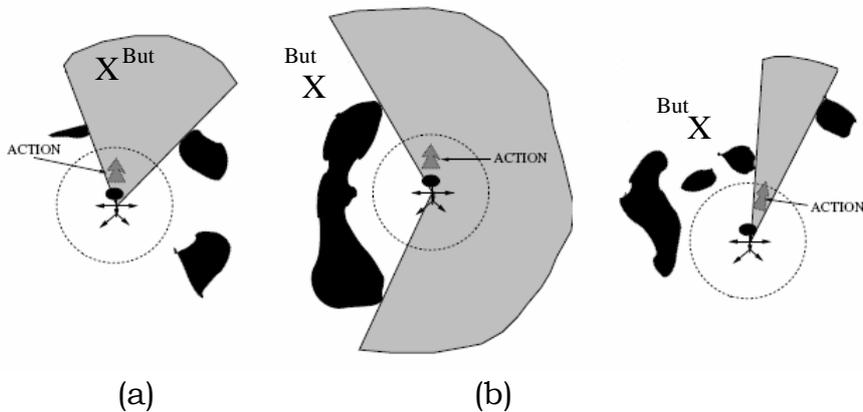


Figure II.13: Différents types d'actions menées dans des situations dites de haute sécurité.

En fonction de la situation dans laquelle se trouve le système mobile, la méthode *ND* va prendre une décision prédéfinie.

Les deux principaux avantages de cette méthode sont les suivants: le premier est sa capacité à gérer des environnements très encombrés, et le second, elle a une prévisibilité intéressante des comportements. Toutefois, elle ne considère ni la forme du système mobile différente du cercle, ni elle tient compte des contraintes dynamiques et cinématiques du système mobile [Fig05].

II.2.2.6. La méthode Elastic Bande ou *EB* [Kha93].

L'idée de base de l'approche *EB* est d'implémenter une navigation locale basée sur les données capteurs par déformation, en temps réel, de la trajectoire calculée par un planificateur global. Les données issues des capteurs embarqués permettent de renseigner le système mobile sur la position des obstacles qui l'entourent.

La bande élastique se représente par un chapelet de disques se chevauchant appelés bulles (Figure II.14). Chaque bulle, d'un diamètre variable, représente un espace libre empruntable par le système mobile. Elles sont soumises à deux types de forces : répulsives et attractives. Les forces répulsives sont générées par les obstacles et les forces attractives s'exercent uniquement entre les bulles consécutives. Ceci a pour effet de maintenir la bande élastique tendue et éloignée des obstacles.

Étant donné que cette méthode repose explicitement sur l'idée d'un chemin prédéfini, l'absence de planification/cartographie la rend non implémentable. En outre, dans certaines situations où les obstacles exigent du système mobile de s'adapter aux sévères trajectoires, cette méthode ne fournit pas de solutions, et propose une re-planification d'une nouvelle trajectoire [Fig05].

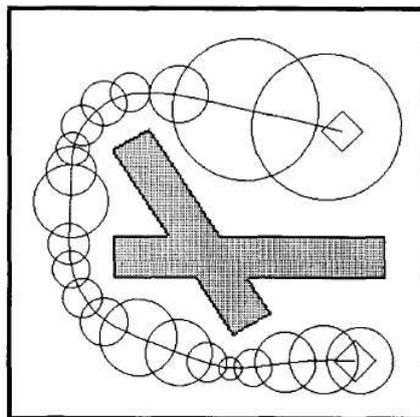


Figure II.14: Chapelet de bulles [Kha93].

II.2.2.7. Autres méthodes.

Plusieurs autres méthodes locales de navigation existent dans la littérature. Il y a celles qui sont basées sur la logique floue par exemple dans [Tza99, Doi02]. L'évitement d'obstacles basé sur la logique floue exige un travail intensif au cours de la phase de conception. Par exemple dans [Doi02], 238 règles ont été utilisées dans un algorithme d'évitement d'obstacle basé sur la logique floue.

La théorie des systèmes dynamiques permet l'utilisation des équations différentielles pour décrire le comportement du système mobile. Habituellement, le comportement souhaité est défini en prenant l'objectif (ou but) comme attractif

et les obstacles comme répulsifs. Par exemple dans [Faj03], un modèle d'un homme en train de poursuivre un but tout en évitant les obstacles a été utilisé pour élaborer un algorithme de poursuite de but avec évitement d'obstacle en simulation. Notez que, définir la relation environnement-système mobile avec les équations différentielles n'est pas toujours simple. En outre, le comportement final n'est pas prévisible.

II.2.3. Les approches globales.

La principale difficulté des approches globales réside dans la modélisation de l'environnement de navigation. Une fois le modèle obtenu et connaissant la position du système mobile dans ce dernier, il est alors possible, après une discrétisation de l'environnement, de calculer une trajectoire (ou un chemin) reliant cette position au but qui doit être atteint.

Un des avantages de ces approches est qu'elles peuvent répondre à la question de complétude : si des solutions existent, l'algorithme en retournera une, s'il n'y a pas de solution, l'algorithme le signifiera. Ces approches tiennent compte des propriétés globales de l'espace de configuration (la carte globale) du système mobile et peuvent intégrer des contraintes spécifiques, qui doivent être satisfaites à chaque étape de la réalisation de la tâche.

II.2.3.1. Espace des configurations [Loz92,Ken99,Bay05].

L'espace des configurations (généralement noté *C-Space*) d'un système mobile est constitué par l'ensemble des configurations qui lui sont accessibles, le système mobile est représenté par un point matériel, et les obstacles de l'environnement sont élargis afin de tenir compte de la forme géométrique du système mobile (Figure II.15). Il est plus facile, par exemple, de faire l'intersection d'un point unique avec les obstacles du *C-Space*, plutôt que de faire l'intersection des obstacles avec un système mobile dans un espace cartésien.

Dans cet espace, on trouve l'ensemble de configurations interdites pour le système mobile noté C_{obs} et l'ensemble des configurations admissibles (appelé également espace libre) par le système mobile noté C_{free} . La séparation de ces deux sous espaces est en général réalisée en utilisant l'une des deux méthodes suivantes :

- l'espace de configuration du système mobile *C-Space* est divisé en cellules, chaque cellule appartient soit au sous espace accessible au système C_{free} , soit au sous espace interdit au système C_{obs} .
- Le contour des obstacles sert à séparer l'espace libre de l'espace interdit.

L'espace de travail du système mobile peut être différent de son espace des configurations. L'espace de travail correspond à l'espace dans lequel se déplace le système mobile. Il peut être en *3D* si le système mobile peut se mouvoir dans l'espace tridimensionnel, ou *2D* si le système se déplace sur le sol. L'espace des configurations prend, quant à lui, en compte tous les degrés de liberté du système mobile. Par exemple, si un système mobile peut se translater suivant les trois axes du repère classique *3D* tout en ayant la possibilité de pivoter sur chacun de ces axes, son espace des configurations est de dimension six. L'espace

des configurations est généralement de plus grande dimension que l'espace de travail.

La difficulté avec cette approche se pose dans la construction de la structure de données contenant les obstacles du C -Space. Le calcul mathématique peut devenir très complexe, et par conséquent, il convient d'examiner si la complexité liée à la construction de cette structure de données est justifiée par une réduction de la complexité des algorithmes de planification [Lat91].

Dans ce qui suit, nous présentons quelques approches proposées dans la littérature pour discrétiser l'espace libre C_{free} .

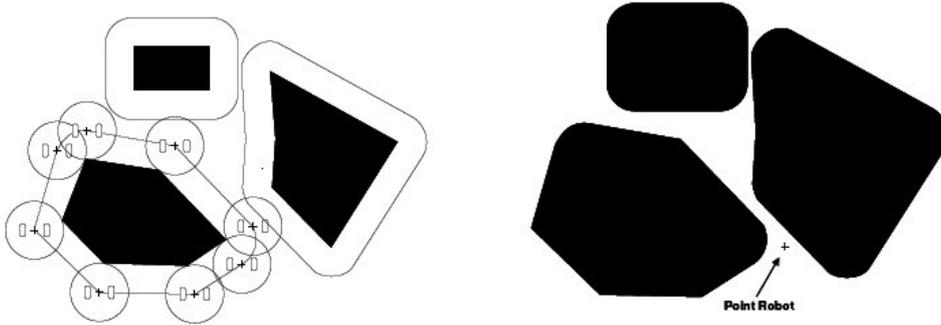


Figure II.15: Reconstruction d'un C -Space : (a) Création de C_{obs} ; (b) le C -Space reconstruit avec un système mobile représenté par un point matériel [Ken99].

II.2.3.2. Discrétisation de l'environnement.

La représentation topologique (§I.3.2) fournit directement une discrétisation de l'environnement réel utilisable pour la navigation. Dans le cas de la représentation métrique (§I.3.1), qui représente l'espace de manière continue, la navigation n'est utilisable qu'après discrétisation de l'espace libre représenté dans la carte de l'environnement.

Pour ce faire, certains modèles intègrent directement cette décomposition au niveau de la cartographie, en construisant une carte topologique parallèlement à la carte métrique [Arl99,Cha85,Thn99,Buh95]. D'autres modèles font appel uniquement à la décomposition de l'espace libre. De plus, différentes techniques, tels les champs de potentiel analogues à ceux décrits dans le paragraphe précédent (§II.2.2.2), permettent de calculer des chemins directement dans le domaine continu, sans phase préalable de discrétisation [Lat91,Dud00a,Mur00].

Il existe deux catégories de méthodes pour discrétiser l'espace de recherche des cartes métriques. Les méthodes de la première catégorie font appel à des décompositions en cellules, de différentes formes, qui permettent de reproduire la topologie de l'espace libre [Lat91, Mur00] (Figure II.16). Les cellules obtenues sont alors utilisées de manière similaire aux noeuds des cartes topologiques dans la navigation, les cellules adjacentes étant considérées comme reliées par une arête.

Les méthodes de la seconde catégorie font appel au précalcul de chemins entre des points répartis dans l'environnement [Lat91,Lau98] (Figure II.17). Les points représentent les noeuds d'une carte topologique, tandis que les chemins précalculés reliant les noeuds représentent les arêtes de cette carte.

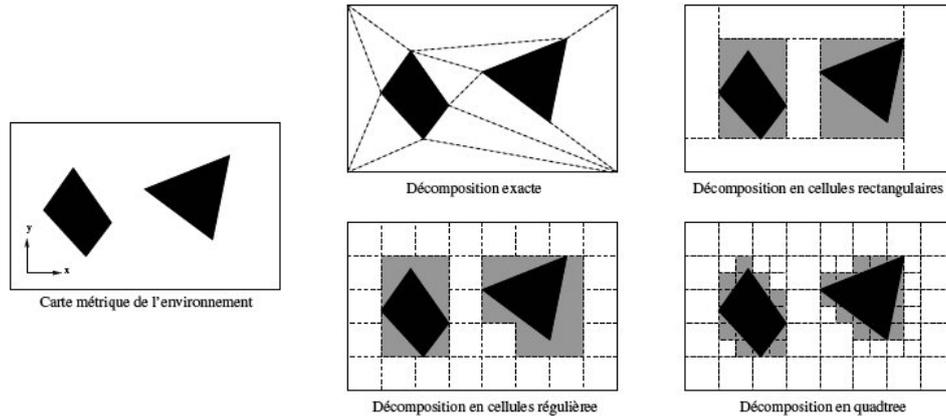


Figure II.16: Décompositions en cellules de l'espace libre dans les cartes métriques.

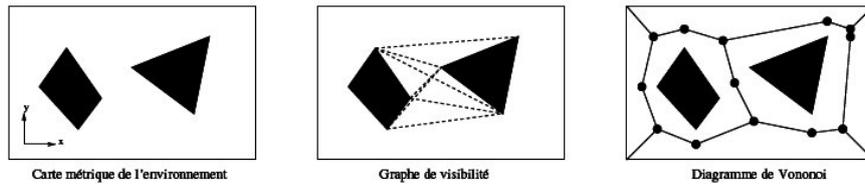


Figure II.17: Décompositions en chemins précalculés dans les cartes métriques.

II.2.3.3. Planification d'un chemin (trajectoire) global.

La planification de chemin se concentre sur les aspects géométriques de la planification de mouvement. Formulé dans l'espace des configurations, le système mobile est représenté par un point matériel dans ce dernier, et tout problème de planification de chemin se ramène alors au problème du calcul d'une courbe continue dans cet espace.

Cette courbe doit relier deux configurations données et ne doit contenir aucune configuration entraînant une collision entre le système mobile et l'un des obstacles fixes de l'environnement. Toute courbe vérifiant ces deux propriétés (continuité et non collision) est une solution, et l'enjeu principal de la planification de chemin consiste, en général, à calculer la courbe la plus courte, *ie*, le chemin géométrique le plus court entre la configuration initiale et la configuration finale.

La différence essentielle entre un chemin et une trajectoire est la dimension temporelle. Un chemin est de nature statique, il représente là où doit passer le système mobile mais ne dit rien quant à la façon dont il va y passer (à quelle vitesse, à quel instant). De fait, la planification de chemin ne s'applique qu'à des environnements statiques. La planification de trajectoire s'impose donc dès lors que l'on souhaite (où que l'on doive) prendre en compte des contraintes dynamiques, *ie* dépendantes du temps. Ces contraintes dynamiques sont de deux types : celles qui concernent l'environnement (présence de mobiles), et celles qui concernent le système mobile (sa dynamique). On notera que ces deux types de contrainte ne sont pas indépendants. S'il est en effet envisageable de planifier une trajectoire en environnement statique (pour minimiser le temps de déplacement ou bien l'énergie consommée), dès lors que l'on se place en environnement dynamique, les capacités dynamiques du système mobile doivent

être prises en compte (ne serait-ce que de façon simplifiée en considérant une vitesse maximale par exemple).

Parmi les méthodes de planification de chemin proposées dans la littérature, nous distinguons trois types de méthodes : les approches par *roadmap*, les approches par décomposition en cellules et les méthodes par fonctions de potentiel.

a. Méthodes par *roadmap* [Ken99].

Les méthodes par *roadmap* sont souvent appelées *skeletonization methods* ou *roadmap methods* dans la littérature anglosaxonne. Dans ces méthodes, un réseau de courbes, appelé *roadmap*, est défini comme un ensemble de courbes monodimensionnelles capturant la connectivité de l'espace libre du système mobile C_{free} . Une fois que la *roadmap* notée \mathcal{R} est construite, elle est utilisée comme un ensemble de chemins "standards". La tâche suivante consiste alors à trouver un chemin approprié entre les points initial et final dans \mathcal{R} .

La difficulté principale dans ces méthodes se situe au niveau de la construction de \mathcal{R} . De nombreuses méthodes ont été proposées pour obtenir \mathcal{R} . Nous citons par exemple les populaires graphes de visibilité, la méthode des cônes généralisés (ou *freeway net*), les diagrammes de *Voronoi* ou encore la méthode des silhouettes.

♦ Méthode de *Voronoi*.

Les méthodes exploitant les diagrammes de *Voronoi* cherchent à obtenir des courbes qui soient les plus distantes possibles des C_{obs} (Figure II.18). Ces diagrammes sont obtenus par des approches de rétroaction [Yap85].

Le diagramme de *Voronoi* est un graphe qui suit une ligne équidistante de deux ou plusieurs obstacles. Un arc droit est formé entre une paire de bords d'obstacles, ou une paire de sommets, tandis qu'un arc courbé est formé entre un sommet et un bord d'obstacle .

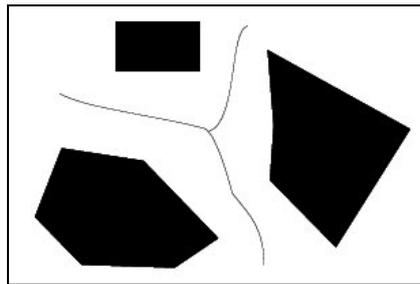


Figure II.18: Diagramme de *Voronoi*.

♦ Méthode de graphe de visibilité [Gho87].

Dans cette méthode, un certain nombre de noeuds sont créés qui correspondent aux sommets du C_{obs} , et les positions de départ et d'arrivée. Un graphe de visibilité est établi reliant par un arc un noeud donné à tous les autres noeuds visibles, à condition qu'il ne traverse pas un obstacle (Figure II.19).

Cette méthode crée des chemins qui longent les limites d'obstacles, tandis que les diagrammes de *Voronoi* (Figure II.18) gardent le système mobile aussi loin que possible des obstacles.

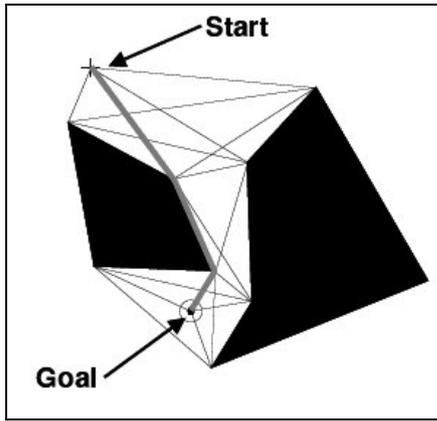


Figure II.19: Exemple d'un graphe de visibilité [Ken99].

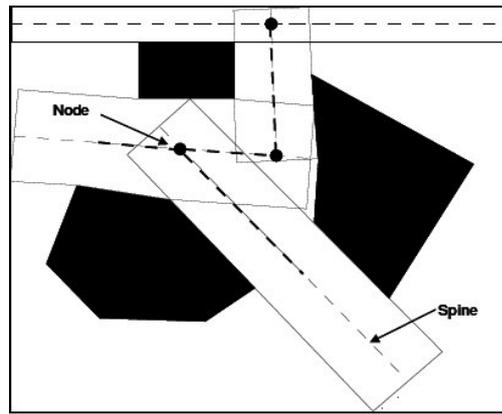


Figure II.20: Méthode de Brooks [Ken99].

◆ Méthode des cônes généralisés (Brooks).

La méthode des cônes généralisés (ou *freeway net* [Bro83]) approxime l'espace libre au moyen d'un certain nombre de cylindres appelés cylindres généralisés (ou *freeway*). Ces cylindres sont placés entre des paires de bords d'obstacles et sont limités par ces derniers (Figure II.20).

Chaque cylindre possède une colonne vertébrale (ou *spine* en anglais), et peut être construit en utilisant ses spines comme des arcs, et l'intersection des spines comme des nœuds. Ainsi, un graphe est construit possédant des propriétés similaires au diagramme de *Voronoi* de telle sorte que le système mobile est maintenu loin des obstacles dans son espace d'évolution.

Cette méthode est relativement rapide, mais incomplète et, par conséquent, elle peut ne pas trouver de chemin même si ce dernier existe, particulièrement dans le cas d'un espace d'évolution encombrant.

◆ Méthode des silhouettes.

Canny [Can87,Can88] utilise une méthode qui génère une *roadmap* qui borde tous les obstacles de l'espace d'évolution. Cependant, il réduit une représentation multidimensionnelle d'un espace de travail en un graphe à une dimension. La silhouette est ce qui serait perçu si l'espace de travail a été fait de plexiglas et l'observateur regardait d'en haut.

La méthode silhouette construit un graphe de manière récursive en utilisant, par exemple, un balayage d'un plan 2D à travers un plan 3D du *C-Space* pour localiser les bords des obstacles, et ensuite, un balayage 1D à travers le plan 2D pour identifier les sous graphes non connectés, et les joindre avec des arcs (Figure II.21). Pour un espace avec plus de trois dimensions, le processus reste le même, sauf le *C-Space* de dimension n est balayé en utilisant un plan de dimension $(n-1)$ (hyper planes), et de manière récursive jusqu'au balayage 1D.

Pour générer un chemin, le point de départ et d'arrivée sont traités comme des nœuds et sont reliés dans le reste du graphe, qui peut être ensuite utilisé pour la recherche du chemin optimum.

La méthode silhouette tend à être utilisée dans l'analyse théorique des espaces d'évolution pour trouver leur complexité, plutôt que dans la planification de chemin réel [Hwa92].

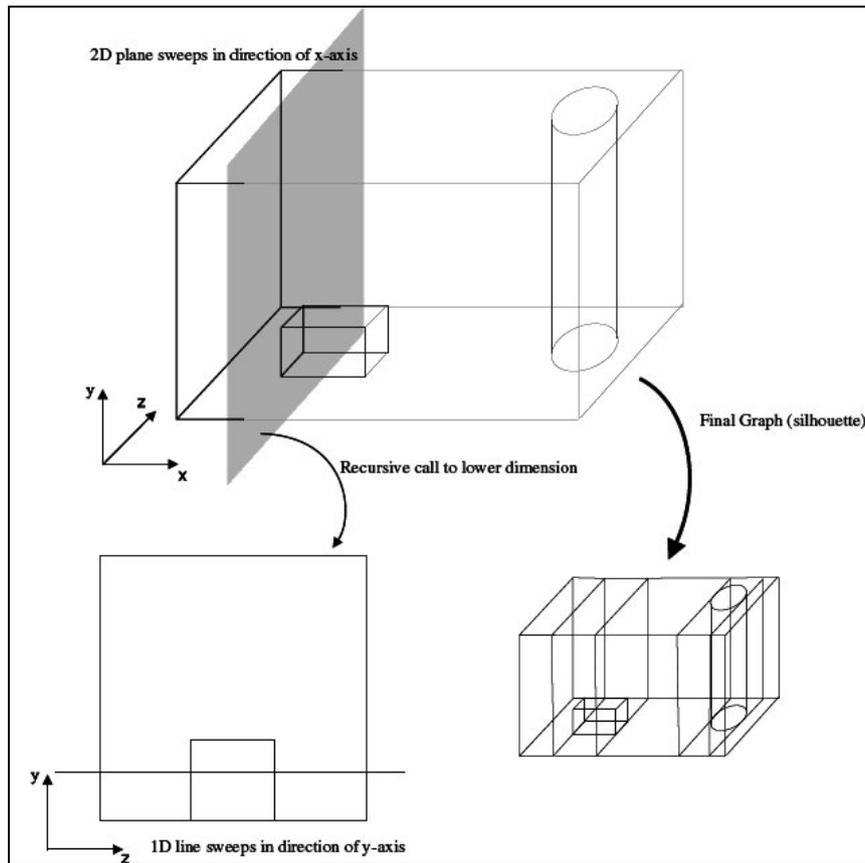


Figure II.21: Exemple de la méthode des silhouettes [Ken99].

b. Méthodes de décomposition cellulaire [Moz95,Ken99,Sie04].

Les méthodes de décomposition cellulaire sont probablement les méthodes les plus populaires dans le domaine de la planification de trajectoires. Elles consistent à partitionner l'espace des configurations libres (ou de travail) C_{free} du système mobile en un ensemble de régions connexes adjacentes appelées *Cellules*. Un graphe représentant les relations d'adjacence entre les cellules est ensuite construit. Ce graphe est communément appelé graphe de connectivité. Le problème se réduit alors à la recherche dans le graphe des connectivités d'un chemin optimal selon un critère bien défini.

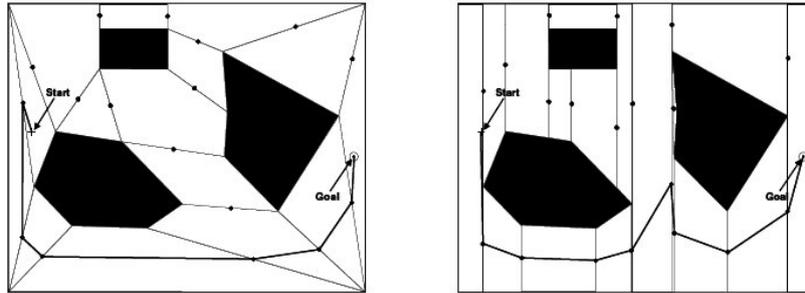
L'approche par décomposition a été adoptée dans plusieurs travaux et a conduit à la mise en œuvre de nombreuses méthodes. Celles-ci sont généralement classées en deux catégories et se distinguent par les modèles de décomposition qu'elles utilisent et leur complétude quant à la résolution du problème de planification.

◆ Méthodes exactes.

Pour représenter exactement l'espace libre C_{free} en utilisant des cellules, un certain nombre de formes irrégulières des cellules doivent être placées dans ce dernier. Ces formes doivent être convexes (chaque point à l'intérieur d'une cellule doit être visible pour tous les autres points de la cellule) et recouvrent exactement l'espace libre C_{free} .

La Figure II.22.a montre l'espace C_{free} divisé en un certain nombre de polygones convexes. La Figure II.22.b montre une représentation qui est aussi exacte, mais qui est beaucoup plus simple à calculer en utilisant des cellules trapézoïdales.

Les chemins générés en utilisant cette forme (trapézoïde) ne sont pas toujours optimaux.



(a) décomposition polygonale. (b) décomposition trapézoïdale.
Figure II.22: Exemples de décomposition exacte de C_{free} [Ken99].

◆ Méthodes approchées.

Ces méthodes se distinguent des précédentes par la simplicité de la structure des cellules utilisées pour la décomposition et l'approximation faite de l'espace libre C_{free} . Cette structure est utilisée pour construire une représentation hiérarchique en cellules élémentaires identiques et permettant l'adaptation de la taille de celles-ci à la géométrie des zones à recouvrir (Figure II.23).

Plusieurs types de formes de cellule sont généralement utilisés en pratique, on peut citer: la représentation en octrees [Fav84], le découpage en tranches [Lag87], ou encore en polyèdres [Sim89], etc.

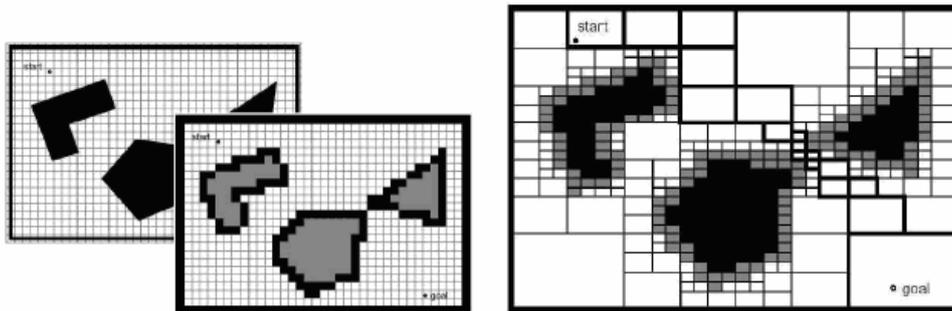


Figure II.23: Décomposition approchée de l'espace libre [Sic04].

En comparant les deux approches de décomposition, il ressort que les méthodes exactes sont plus complètes en théorie que les méthodes approchées [Moz95]. Toutefois, la complexité de leur mise en œuvre, comme en témoigne la solution donnée dans [Scw83], les rend moins appropriées en pratique que les techniques approchées même si celles-ci restent parfois limitées à des espaces de dimension réduite. Cette constatation n'est pas générale puisqu'il apparaît dans [Avn88] qu'une méthode exacte peut conduire dans le cas plan et pour une modélisation polygonale du système mobile et de son environnement à une implantation efficace.

c. Méthodes par fonctions de potentiel [Kor90].

Le principe de ces méthodes est détaillé dans le paragraphe (§II.2.2.2). Les déplacements du système mobile sont calculés itérativement par un algorithme de descente du gradient du potentiel composé d'un premier champ attractif au but et d'un ensemble de champs répulsifs modélisant la présence d'obstacles.

Ces méthodes restent sensibles à l'occurrence de minima locaux engendrant des configurations de blocage ou d'oscillation. Ces minima sont généralement liés à la géométrie et à la répartition des obstacles dans l'espace de travail du système mobile.

II.2.3.4. Recherche d'un chemin optimal.

Connaissant la carte de l'environnement discrétisé, la position et l'orientation du système mobile au sein de cette carte, il existe différentes méthodes pour déterminer ou calculer un chemin permettant à ce système d'atteindre sa cible. Cela s'avère nécessaire étant donné que pour une certaine configuration d'objets dans l'environnement, il peut y avoir plus d'une façon de réaliser la tâche.

La recherche d'un chemin est réalisée en considérant deux aspects :

- Le temps mis par le système de traitement pour trouver le chemin optimal.
- La certitude d'obtenir le chemin optimal.

Les deux aspects sont liés et déterminent deux types d'algorithmes [Hao04]:

- L'algorithme de recherche exhaustive qui assure la détermination du chemin optimal dans tous les cas, mais dont le temps de calcul est long. En effet, ce type d'algorithme dit naïf consiste à énumérer tous les chemins possibles dans un graphe à partir du nœud (cellule) source et choisir le chemin avec le moindre coût. L'augmentation du nombre de nœuds rend ce type d'approche rapidement inutilisable par le fait des besoins en temps de calcul et de capacité mémoire nécessaires.
Parmi les algorithmes les plus utilisés, nous citons l'algorithme de *Dijkstra*, l'algorithme de *Bellman* et l'algorithme de *Lee*.
- L'algorithme de recherche heuristique qui détermine un chemin possible (non obligatoirement optimal) avec un temps de calcul court. Ce type d'algorithme réduit le nombre de nœuds en tenant compte des connaissances particulières du problème à résoudre. Le savoir heuristique diminue le temps de recherche et la capacité mémoire nécessaires, par contre ne garantit pas une solution optimale [Pru88].
Parmi les algorithmes les plus utilisés, nous citons l'algorithme de A^* et l'algorithme D^* .

a. Algorithme de *Dijkstra* [Lau01].

L'algorithme proposé par *Dijkstra*, permet de résoudre le problème de recherche de chemin optimal, de longueur minimale, dans un environnement schématisé par un graphe. Il assure la recherche de tous les chemins possibles du nœud source vers le nœud destination (ou but). Un seul chemin dont le coût est minimum est retenu.

Le principe de cet algorithme repose sur le fait que si un nœud P quelconque du graphe est situé sur le chemin minimal du nœud source S au nœud destination D , alors le chemin minimal de S à P fait partie du chemin minimal de S à D [Pru88]. Cet algorithme ressemble fortement à celui de *Bellman*. Cependant, *Dijkstra* a eu l'idée de marquer chaque nœud visité et empêché ainsi son algorithme de revenir en arrière.

L'algorithme de *Dijkstra* possède une complexité en temps, malgré cela, il est intéressant puisqu'il ne nécessite pas le développement de l'ensemble des chemins possibles. Par contre l'ensemble des nœuds doit être atteint avant d'affirmer que la solution optimale a été trouvée.

b. Algorithme de Bellman [Pru88].

Cet algorithme à correction d'étiquettes a été conçu dans les années 50 par *Bellman* et *Moore*. Il s'agit d'une méthode de programmation dynamique (optimisation récursive) qui consiste à calculer de proche en proche les coûts minima du nœud source à un nœud n par rapport aux coûts minima du nœud source à tous les nœuds prédécesseurs de n .

L'un des inconvénients majeurs de cet algorithme est qu'il peut (dans de nombreux cas) examiner plusieurs fois un nœud déjà visité. Ceci pose aussi un problème de performance non négligeable.

c. Algorithme de Lee [Lau01].

Cet algorithme est bien adapté à la recherche d'un chemin optimal à travers un environnement modélisé par une grille où l'unité de lieu est une case carrée. Le système mobile peut circuler d'une case à une autre adjacente selon les quatre orientations.

Le processus transmet une onde à partir de la case source vers la case but de la manière suivante : Pour chaque case, entourant la case de départ et ne représentant pas un obstacle, est assigné un indice de (1). Par la suite un indice de (2) est assigné à toutes les cases vides non marquées, voisines des cases marquées par (1). Ce processus est reproduit jusqu'à atteindre l'une des deux solutions:

- 1^{ère} solution : On ne peut plus propager l'onde et le but n'est pas atteint. Il n'y a donc pas de trajectoire possible joignant la case de départ à l'arrivée.
- 2^{ème} solution : La case d'arrivée est trouvée. Le chemin est calculé en remontant du but vers la source en sélectionnant, à chaque opération comme case prochaine, une des cases de marquage inférieure adjacente à la case courante.

Cette méthode est très simple à implémenter, mais reste coûteuse en temps et en espace mémoire à cause de la propagation d'onde.

d. Algorithme A* [Hao04].

L'algorithme A^* est un algorithme de recherche dans un graphe : cette recherche est guidée par une heuristique, qui donne pour chaque nœud du graphe un minorant de la distance restant à parcourir avant d'atteindre le nœud final. A chaque étape de la recherche, l'algorithme propage l'ensemble des nœuds que l'on sait atteindre depuis le nœud de départ, en favorisant les nœuds les plus intéressants (ceux dont la somme entre la distance déjà parcourue depuis le nœud de départ et l'estimation de la distance restant à parcourir est minimale).

L'inconvénient de cet algorithme est son coût, et ce pour deux raisons. D'une part, lorsque la construction de l'arbre de recherche atteint un C_{obs} , elle se propage très lentement le long de cet obstacle. De plus, l'algorithme n'échoue qu'après avoir étendu autant que possible l'arbre de recherche, avec une

précision δ , ce qui demande de nombreuses itérations. Cet inconvénient est compensé par le fait que son coût peut être diminué en augmentant δ , c'est-à-dire en diminuant la précision de la recherche.

e. Algorithme D^* [Ste94].

Le nom D^* a été choisi à cause de sa ressemblance avec l'algorithme A^* à l'exception qu'il est dynamique dans le sens où les paramètres de coût entre les cellules ou carrés peut changer durant le processus de résolution du problème de planification de chemin.

L'algorithme génère un chemin global en utilisant un modèle déjà établi de l'environnement. En cours de suivi de ce chemin, le système mobile peut rencontrer un obstacle qui n'était pas défini au préalable, dans ce cas, le système peut re-planifier son chemin pour atteindre son but sans heurter l'obstacle.

Appartenant à la famille des algorithmes de replanification rapide, l'algorithme D^* a pour avantage, par rapport à l'algorithme A^* , qu'il peut être utilisé en ligne. Son inconvénient réside dans le fait que la cinématique des systèmes mobiles est très peu prise en compte.

II.2.4. Approches hybrides.

Afin de permettre à un système mobile de se déplacer sans collision vers un but donné, les recherches en robotique mobile ont donné naissance aux méthodes de navigation autonome. Ces méthodes se divisent en deux familles principales citées ci-dessus. Les approches globales permettent de calculer au préalable un chemin ou un ensemble de trajectoire en se basant sur une connaissance a priori de l'environnement. Les approches locales utilisent l'environnement perçu afin de générer un mouvement à exécuter sur un pas de temps, à la suite duquel un autre mouvement est généré et ainsi de suite. Ces mouvements sont appelés "mouvements partiels".

L'inconvénient majeur des approches globales est leur incompatibilité avec des contraintes de calculs en temps réel, rendant difficile leur utilisation en environnement dynamique. Quant aux approches locales, leur inconvénient principal est celui des minimums locaux : d'un point de vue local, une direction peut sembler la plus judicieuse pour atteindre le but alors qu'en réalité, elle conduit à minimum local (un cul-de-sac, ou un puit de potentiel).

Ainsi, pour pallier aux problèmes inhérents à ces deux familles, un troisième type d'approche a vu le jour, il s'agit de la famille des méthodes hybrides. Ces dernières sont généralement composées d'un algorithme réactif dont les décisions se font en connaissance d'un chemin déjà planifié par une fonction globale.

Parmi les méthodes hybrides les plus citées dans la littérature, nous présentons [Laf06]:

II.2.4.1. La méthode *Vector Field Histogram (VFH^*) [Ulr00b].**

La méthode VFH^* fait suite aux améliorations successives des méthodes VFH [Bor91c] et $VFH+$ [Ulr98] purement réactives. Elle teste les conséquences de plusieurs choix avant de prendre sa décision.

Une fois l'histogramme (§II.2.2.3) construit autour de la position courante du système mobile et discrétisé en secteurs angulaires, l'algorithme *VFH** détermine 1 à 3 directions dans chaque zone de passage libre. Ces directions portent le nom de *directions candidates primaires*. Pour chacune de ces directions, *VFH** calcule la position et l'orientation qu'aura le système mobile après s'être déplacé d'un pas de distance ds . L'histogramme est alors recalculé pour chacune de ces positions afin d'en déterminer des *directions candidates projetées*. Le procédé se répète, de projection en projection, afin d'obtenir un arbre de recherche de profondeur notée ng .

Chaque noeud de l'arbre correspond donc à une position et une orientation, et chaque branche représente une distance à parcourir dans une direction donnée. La distance totale projetée dt est alors déterminée par la relation $dt = ng \cdot ds$.

Pour prendre en compte (de façon approximative) la dynamique du système mobile, les branches de l'arbre ne sont pas rectilignes. En effet, plus le changement de direction est fort et plus les branches de l'arbre sont incurvées (Figure II.24).

Un coût est ensuite attribué à chaque noeud pour qu'une fonction heuristique (A^*) puisse choisir la direction candidate primaire la plus appropriée, i.e. celle appartenant à l'ensemble des directions menant le plus directement au plus près du but. L'arbre est reconstruit chaque fois que le système mobile atteint la 1ère position projetée i.e. le 1er noeud de l'arbre.

La figure II.25 montre un exemple de chemin parcouru en fonction de la profondeur (ng) de l'arbre. La figure II.26 permet de voir les arbres qui ont permis la construction de ces chemins. Les lignes noires indiquent les trajectoires projetées des directions candidates primaires. Les lignes grises montrent les directions dont les coûts des noeuds n'ont pas permis la sélection [Laf06].

Durant tout son trajet, la méthode *VFH** maintient la vitesse de déplacement du système mobile à son maximum. Cette vitesse est cependant réduite proportionnellement à la proximité d'un obstacle. De même, cette vitesse sera réduite lorsqu'un changement de direction sera nécessaire.

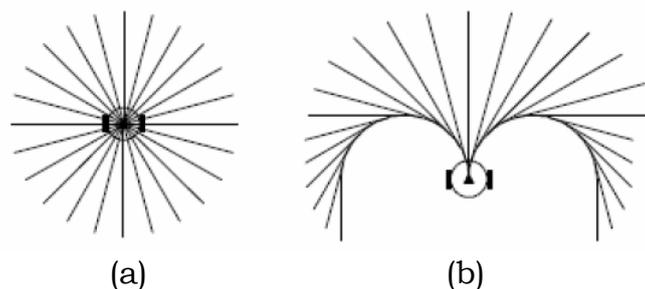


Figure II.24: approximation des trajectoires : a) sans prise en compte de la dynamique du système mobile, b) avec prise en compte de cette dynamique

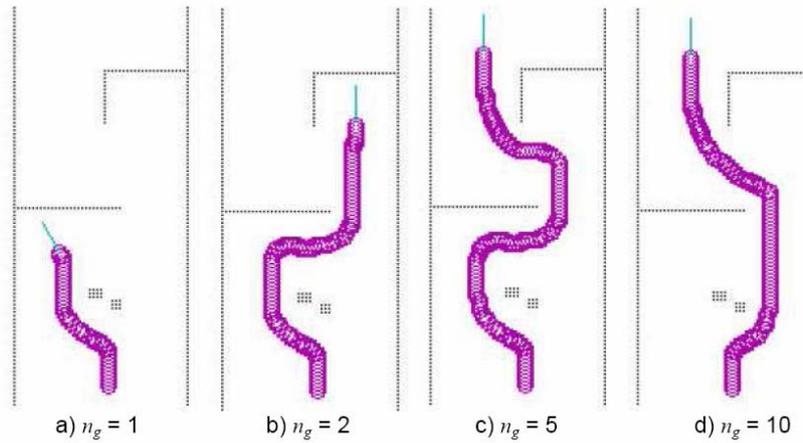


Figure II.25: chemins générés selon la profondeur de l'arbre.

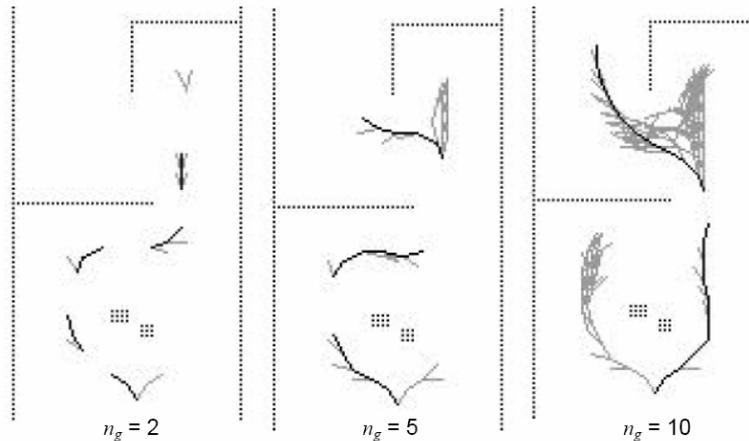


Figure II.26: Arbres générés pour les chemins b), c) et d) de la figure II.25.

II.2.4.2. La méthode *Global Dynamic Window Approach (GDWA)*.

La méthode *GDWA* s'appuie sur la méthode *DWA* (§II.2.2.4.c). En effet, cette dernière étant purement réactive, une fonction permettant de suivre un chemin lui a été ajoutée pour pallier aux problèmes des minimums locaux.

De ce fait, ces deux méthodes (*GDWA* et *DWA*) possèdent la même modélisation et diffèrent lors de la prise de décision.

Le processus décisionnel de la méthode *GDWA* est basé sur une fonction *NF1* permettant de générer un chemin global. La fonction *NF1* utilise une technique de propagation de vague afin de donner une distance au but pour chaque point de l'espace qu'elle parcourt (Figure II.27). Une descente de gradient permet alors d'en extraire le chemin global.

Chaque commande de l'espace de recherche (ou plan des vitesses) est évaluée par une fonction heuristique définie telle que :

$$\Omega_g(\vec{p}, \vec{v}, \vec{a}) = \alpha.nf1(\vec{p}, \vec{v}) + \beta.vel(\vec{v}) + \gamma.goal(\vec{p}, \vec{v}, \vec{a}) + \delta.\Delta nf1(\vec{p}, \vec{v}, \vec{a}) \quad (II.6)$$

avec :

Ω_g est la fonction heuristique de *GDWA*.

\vec{p} est la position courante du système mobile dans le plan (x, y) .

\vec{v} est la vitesse en translation en x et y extraite du plan de vitesses.

\vec{a} est l'accélération en x et y extraite du plan de vitesses.

$nf1()$, $vel()$, $goal()$, $\Delta nf1()$ sont des sous fonctions telles que :

- $nf1()$ valorise les vitesses alignées sur le chemin,
- $vel()$ favorise les vitesses élevées lorsque le système mobile est loin du but et les faibles vitesses lorsqu'il en est proche.
- $goal()$ renvoie 1 si la commande à évaluer fait atteindre le système mobile à son but, sinon, elle renvoie 0,
- $\Delta nf1()$ valorise les commandes qui réduisent la distance au but lors du prochain cycle.

$\alpha, \beta, \gamma, \delta$ sont des coefficients de pondération.

La commande obtenant la meilleure évaluation est choisie pour effectuer le mouvement partiel suivant.

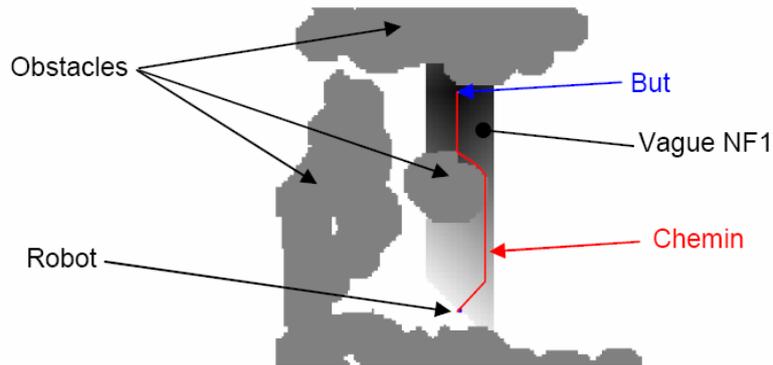


Figure II.27: Génération des distances au but par la fonction $NF1$.

II.2.4.3. La méthode *Global Nearness Diagram (GND)* [Min01,Laf06].

À la manière de la méthode *GDWA*, la méthode *GND* étend la méthode *ND* (§II.2.2.5) [Min04] en utilisant la fonction *NF1* [Phi04] pour pallier aux problèmes des minima locaux. La fonction *NF1* est utilisée pour créer des sous buts sur lesquels se basera la méthode *ND* pour réagir comme s'il s'agissait du but final.

Chaque situation est associée à une trajectoire afin de produire un comportement adéquat. Ainsi, dans les cas de *LS1* et *LS2* (§II.2.2.5), le but des actions est de mener le système mobile dans une situation de haute sécurité (*HSGR*, *HSWR* ou *HSNR*):

- 1) *LS1*: éloigner le système mobile de l'obstacle mais dans la direction la plus orientée vers une région libre proche du but (Figure II.12a). La vitesse choisie est alors proportionnellement inverse à la proximité de l'obstacle.
- 2) *LS2*: éloigner le système mobile des obstacles mais dans la direction la plus orientée vers une région libre proche du but (Figure II.12b). La vitesse est gérée comme pour *LS2*.
- 3) *HSGR*: diriger le système mobile en direction du but à la vitesse maximum (Figure II.13a).
- 4) *HSWR*: choisir la direction au sein de la zone libre appropriée qui longe l'obstacle (Figure II.13b).
- 5) *HSNR*: se diriger au centre de la zone libre adéquate (Figure II.13c).

II.2.4.4. Autres méthodes de planification.

Les algorithmes cités ci-dessus utilisant l'espace de configuration sont très coûteux en temps de calcul quand la dimension de cet espace devient importante.

Des méthodes permettant de répondre à des problèmes complexes peuvent être appliquées pour de grandes dimensions de l'espace de configuration, en relaxant la notion de complétude au bénéfice d'une efficacité pratique et d'une complétude probabiliste. Un planificateur sera dit probabilistiquement complet, si étant donné un problème résoluble, la probabilité d'obtenir la solution tend vers 1 lorsque le temps de calcul tend vers l'infini. Les deux approches les plus efficaces sont les *probabilistic path planner* et *randomized path planner*, des informations plus complètes peuvent être trouvées dans [Bar91,Lan92] pour la première et [Kav96] pour la seconde [Mez01].

II.2.5. Conclusion.

Nous avons présenté le problème général de la navigation d'un système mobile évoluant dans un environnement connu/inconnu a priori, et nous avons passé en revue les plus grandes approches et méthodes de résolution proposées dans la littérature.

Ces méthodes sont soit locales, globales ou hybrides. Nous avons pu voir leur diversité, leurs concepts et leur façon d'aborder la modélisation de l'environnement et d'y réagir. Toutes ces méthodes présentent des facettes intéressantes de par leur conception, mais elles ont toutes un point commun : leur incapacité à assurer la sûreté d'un système mobile évoluant dans un environnement où les obstacles sont mobiles⁶.

II.3. Conclusion.

Pour qu'un système mobile puisse librement naviguer dans un environnement, il est nécessaire de lui fournir la capacité de discrétiser cet espace, non seulement pour se localiser, mais aussi pour avoir la capacité de définir des chemins dans celui-ci.

Dans ce chapitre, nous avons passé en revue les approches les plus utilisées et méthodes de résolution pertinentes proposées dans la littérature pour le problème de la navigation. Ces méthodes se divisent en deux grandes classes: les approches locales et les approches globales.

Les approches locales se basent sur une évaluation de l'espace immédiatement proche du système mobile. Elles peuvent être très simples, ne nécessitent pas de modèle global de l'environnement mais ont un domaine d'application souvent restreint. Du fait de leur simplicité, elles sont généralement exécutées très rapidement et elles permettent de réaliser des tâches de bas niveau, comme l'évitement des obstacles imprévus, essentiels à la sécurité du système mobile.

Les approches globales s'appuient sur un modèle connu de l'environnement, et sur une position du système mobile correctement estimée. Ce modèle mémorise la structure spatiale de l'environnement, indépendamment d'un but précis. Chacune des positions mémorisées dans ce modèle peut alors être utilisée comme but par le processus de planification dont le rôle est de calculer un chemin/trajectoire vers ce but.

⁶ Les méthodes tenant compte de ce type d'environnement ne sont pas traitées dans le cadre de ce travail.

Ainsi, pour pallier aux problèmes inhérents à ces deux dernières, un troisième type d'approche a vu le jour, il s'agit de la famille des méthodes hybrides. Elles ont les avantages des deux approches ci-dessus, et en même temps, éliminent plusieurs de leurs inconvénients.

Après avoir présenté, dans la première partie, un état de l'art concernant les différents types de capteurs utilisés dans un système de perception, les différentes représentations d'environnements et les différents systèmes de localisation et de navigation ainsi que les méthodes développées pour leur résolution. Dans la deuxième partie, nous présentons notre contribution concernant le modèle de l'environnement, la localisation et la navigation du système mobile *ATRV2* dans un environnement qui peut être inconnu a priori.

Dans le chapitre suivant, nous présenterons le système de perception de l'*ATRV2*, nous montrons les principes physiques qui régissent le fonctionnement de ses capteurs, ainsi que les différentes sources d'erreurs inhérentes à leur utilisation. Nous proposons aussi dans ce chapitre le modèle de l'environnement de navigation qui va être utilisé tout au long de ce travail.

Partie II

Localisation et navigation d'un système mobile : cas de l'*ATRV2*.

Chapitre III

Modèle de l'environnement de navigation utilisé.

Après avoir présenté en détail les systèmes de perception utilisés par un système de robotique mobile, nous présentons dans ce chapitre le modèle de l'environnement que nous avons choisi pour la navigation du système mobile *ATRV2*.

Nous commençons d'abord par définir le système de perception utilisé par le système mobile, ensuite, nous définissons la représentation des mesures issues de ce système de perception, en justifiant d'abord le choix du modèle de l'environnement de navigation utilisé tout au long de ce travail, ainsi que le modèle géométrique des capteurs du système de perception.

La méthode de fusion des données capteurs utilisée est aussi présentée dans ce chapitre, ainsi que la validation des travaux de reconstruction de la carte de l'environnement.

III.1. Système de perception utilisé.

Le type d'environnement dans lequel devra se déplacer le système mobile détermine le type de capteurs à utiliser pour la perception : des caméras, des proximètres, et ou des capteurs tactiles. Aucun d'entre eux ne peut à lui seul rendre compte de tout ce que le système mobile doit connaître sur le monde qui l'entoure. La perception de chacun d'eux est partielle et entachée d'erreurs. La solution consistera donc, à faire coopérer un grand nombre de capteurs.

Cette coopération entre capteurs est très intéressante, voire indispensable pour la réalisation des objectifs de la perception. Elle peut se faire entre des capteurs de nature différente (vision et capteurs télémétriques, par exemple) permettant la perception de caractéristiques totalement différentes des objets, ou de même type placés à différents endroits diminuant ainsi les problèmes dus à l'occultation, de la résolution ou du champ de vue.

Dans ce qui suit nous nous intéressons au système de perception du système mobile *ATRV2* à savoir les capteurs à ultrasons et le banc stéréoscopique.

III.1.1. Capteurs à ultrasons [Cou00].

Comme nous l'avons décrit au paragraphe (§.I.1.2.1.a), le capteur à ultrasons présente l'avantage de donner directement une information de distance mais 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 III.1 illustre la situation où deux obstacles possibles produisent la même mesure.

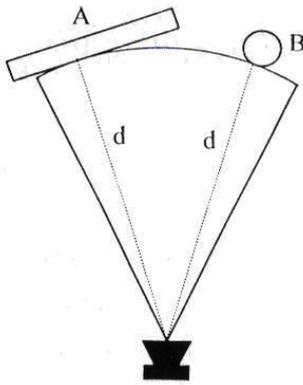


Figure III.1 : Deux obstacles renvoyant la même mesure.

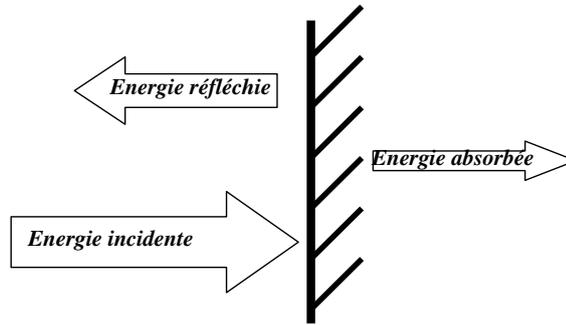


Figure III.2: Répartition des énergies.

Une incertitude sur la distance mesurée, appelée également incertitude radiale, est aussi présente. Elle provient des phénomènes atmosphériques (température, courants d'air, etc.) 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 (6 cm pour le cas des capteurs US du système mobile ATRV2).

Quand une onde acoustique heurte un objet, l'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 être passée à travers elle, ceci dépendant de la surface de l'objet et de l'angle d'incidence du rayon (Figure III.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 (4 m pour le système mobile ATRV2).

Dans le domaine de perception du capteur à ultrasons, 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 III.3).

Si une spécularité intervient, la mesure issue du capteur concerné est inexploitable, ce qui peut être gênant puisque nous n'avons que 12 mesures de capteurs à ultrasons pour décrire l'environnement autour du système mobile ATRV2. Si l'onde ne se réfléchit pas sur un autre obstacle pour revenir sur le capteur, la mesure sera annulée; par contre, si la spécularité est suivie d'une ou plusieurs réflexions sur les obstacles, nous avons alors des réflexions multiples.

➤ **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 III.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.

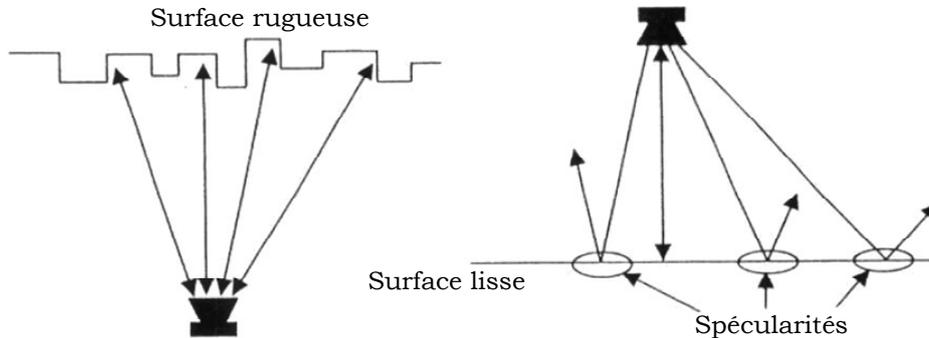


Figure III.3: Phénomène de spécularité.

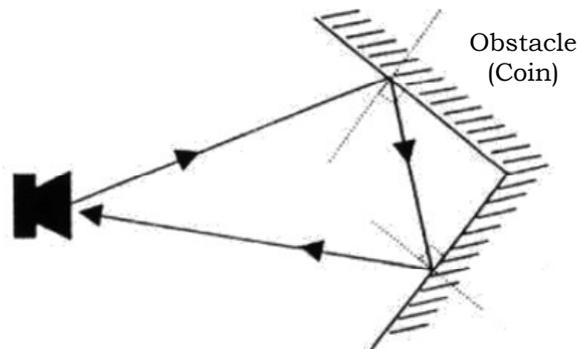
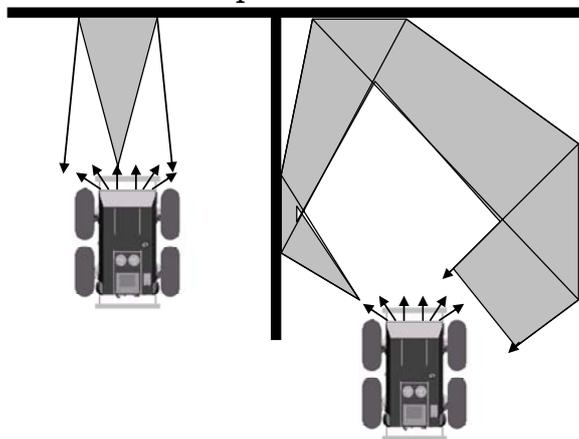


Figure III.4: Exemple de réflexions multiples.



a) Chemin critique direct.

b) Chemin critique indirect.

Figure III.5: Exemple de diaphonie.

➤ **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 système mobile. Si le capteur y se trouve en mode de réception, il interprétera l'onde émise par le capteur x et renvoyée 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 III.5.a). Le capteur y peut aussi recevoir une onde issue de réflexions multiples, il s'agit alors de diaphonie par chemin critique indirect (Figure III.5.b). Là encore ce

type de mesures représente des obstacles fictifs et peut fausser la localisation des objets; le regroupement des capteurs sous forme de noeuds vise à limiter la diaphonie en évitant d'activer simultanément des capteurs voisins.

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 angulaires 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 reconstruction de l'environnement.

Le système mobile *ATR*V2 (Figure III.6.a) est équipé de douze capteurs à ultrasons montés en horizontal autour du système (six à l'avant, deux à l'arrière et deux sur chaque côté). La portée de leur mesure s'étend sur une distance allant de six centimètres jusqu'à quatre mètres. Le lobe principal de leur fonction de sensibilité (ou angle d'ouverture) est contenu dans un angle solide de 30° .

Grâce à ces capteurs, l'*ATR*V2 peut acquérir rapidement une perception panoramique complète à 360° (Figure III.6.b).

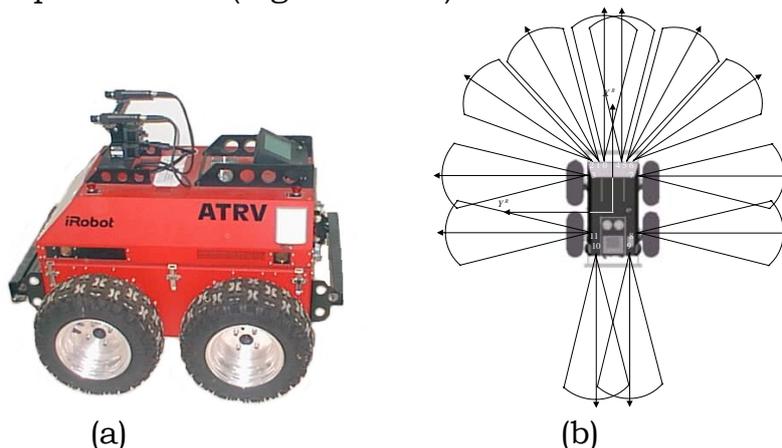


Figure III.6: (a) Le système mobile *ATR*V2, (b) Perception panoramique de l'*ATR*V2.

Cependant, tous ces capteurs sont actionnés en même temps et cela peut provoquer une diaphonie importante entre capteurs (causée par une résolution angulaire faible, des erreurs dues aux réflexions multiples ou des réflexions spéculaires loin du capteur).

III.1.2. Système de vision.

Les systèmes de vision en robotique mobile sont basés sur l'utilisation d'une ou plusieurs caméras *CCD*, très performants en termes de portée, précision et quantité d'informations exploitables. En revanche, l'inconvénient majeur de tels systèmes de perception se situe au niveau de la gestion du flux important de données exploitables: traiter une image demeure une opération délicate et surtout coûteuse en temps de calcul.

Le système de vision du système mobile *ATR*V2 est composé de deux caméras *CCD* couleurs montées sur une tourelle *Pan Tilt* (Figure III.7). Leur utilisation a

pour avantage de fournir des informations très riches sur l'environnement. A partir de ces informations tels que des points et segments de droite extraits de l'image de la scène observée, on peut contrôler le déplacement relatif et la position du système mobile vis à vis d'éléments caractéristiques de l'environnement.



Figure III.7: Système de vision de l'ATRV2.

III.2. Représentation des mesures issues des capteurs embarqués.

III.2.1. Modèle de l'environnement.

Le modèle de l'environnement que nous avons utilisé est le modèle métrique (grille de certitude) basé sur l'algorithme *HIMM* (*Histogramic In Motion Mapping* en anglais) présenté dans [Bor91b]. Il utilise une grille histogramme à deux dimensions pour la représentation des obstacles. Chaque cellule de cette grille possède une valeur de certitude *CV* qui représente la confiance de l'algorithme dans l'existence d'un obstacle à cet endroit.

Ce modèle est exploitable dans tout type d'environnement sans nécessiter d'aménagements particuliers, et peut être utilisé à partir des mesures issues de plusieurs types de capteurs. Chaque capteur construit sa carte locale qui va ensuite être intégrée à une carte locale du système mobile servant à établir la carte globale de l'environnement.

III.2.1.1. La carte locale.

La carte locale est centrée sur le système mobile et sa taille est ajustée sur la plus grande mesure du système de perception embarqué.

A l'arrêt, les cellules occupées par le système mobile sont considérées libres. Lorsque le système mobile se déplace, il utilise sa position, son orientation et les mesures de son système de perception embarqué pour construire sa carte locale.

III.2.1.2. La carte globale.

La carte globale de l'environnement de navigation définit l'espace ou l'endroit dans lequel le système mobile va se mouvoir.

La valeur *CV* de ses cellules est initialisée à zéro. Au cours des déplacements du système mobile, chaque obstacle détecté incrémente la valeur *CV* des cellules qu'occupe cet obstacle, ainsi, les obstacles seront repérés ce qui permet de les éviter.

Toutes les cartes locales sont intégrées pour former une seule carte (carte globale). Chaque carte locale est intégrée à l'ancienne carte globale pour donner une nouvelle carte globale (mise à jour) de l'environnement.

III.2.2. Capteurs à ultrasons.

Un moyen de modéliser le capteur à ultrasons consiste à déterminer la fonction de densité de probabilité $P\left[\frac{r}{d}\right]$ de la mesure de profondeur r connaissant la distance d entre le capteur et l'obstacle [Cou00]. 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 III.8)

$$P\left[\frac{r}{d}\right] = \begin{cases} 1 & \text{si } r = d \\ 0 & \text{si } r \neq d \end{cases} \quad (\text{III.1})$$

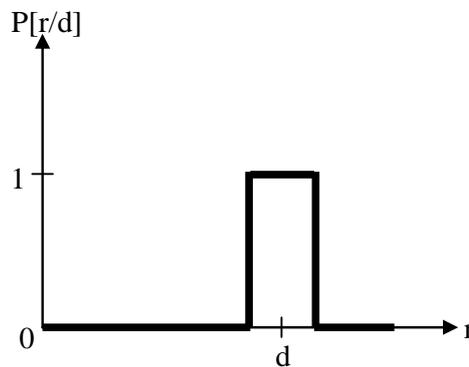


Figure III.8 : Fonction de probabilité d'un capteur idéal.

En enregistrant ce modèle dans une grille, seule la cellule située sur l'axe acoustique du capteur à la distance mesurée est modifiée (Figure III.9). De cette façon, l'incertitude sur la distance mesurée peut être contenue dans la cellule.

Ce modèle a été utilisé pour effectuer des tâches d'évitement d'obstacles pour un système mobile [Bor91b]. Les cellules de la grille représentant l'environnement sont initialisées à la valeur zéro. Puis, au cours du déplacement du système mobile, 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.

Une autre modification a été apportée sur ce modèle, il prend en compte l'espace libre entre le capteur et l'obstacle [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 III.10).

Le problème de ces modèles est qu'ils ignorent l'incertitude angulaire des capteurs à ultrasons. Un capteur de type Polaroid possède un cône d'émission de 24° à 30°, 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 III.11).

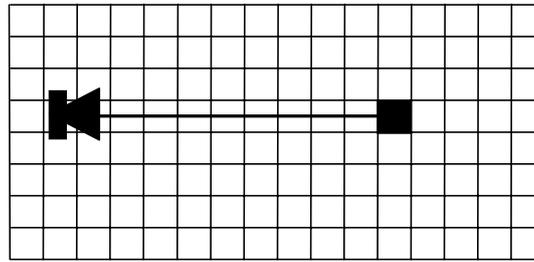


Figure III.9 : Grille pour un capteur idéal.

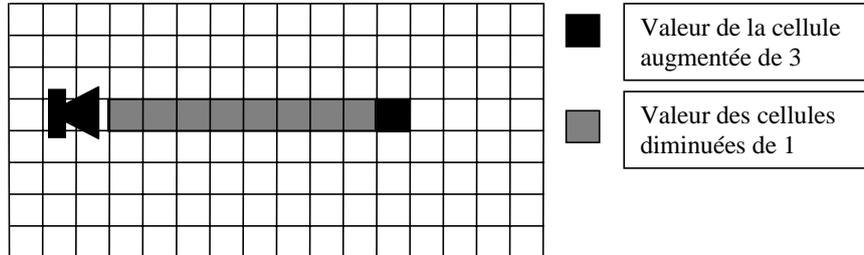


Figure III.10 : Grille pour un capteur idéal prenant en compte l'espace libre.

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 systèmes mobiles consiste à modifier le premier modèle (Figure III.9) en un arc de cercle (Figure III.12) [Men95]; la valeur attribuée à chaque cellule étant inversement proportionnelle au nombre de cellules recouvertes par l'arc entier.

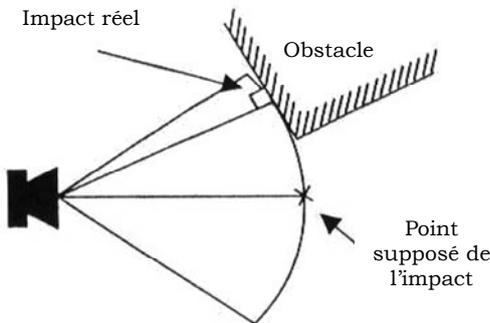


Figure III.11 : Exemple d'obstacle mal perçu.

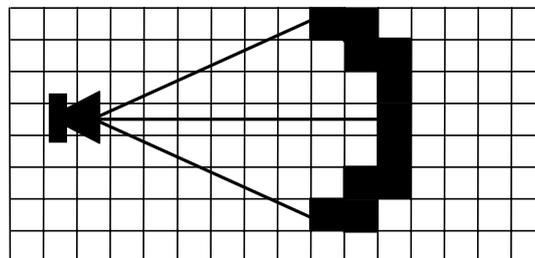


Figure III.12 : Grille prenant en compte l'ouverture angulaire du capteur.

Un autre modèle 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 :

$$P_L = L_r(r).L_a(\theta) \tag{III.2}$$

avec :

$$\begin{cases} L_r(r) = \begin{cases} 1 - \left(\frac{r - R_{min}}{R - \varepsilon - R_{min}} \right)^2 & \text{si } r \in [R_{min}, R - \varepsilon] \\ 0 & \text{sinon} \end{cases} \\ L_a(\theta) = 1 - \left(\frac{2\theta}{\Omega} \right)^2 & \text{si } \theta \in \left[-\frac{\Omega}{2}, \frac{\Omega}{2} \right] \end{cases} \quad (\text{III.3})$$

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 :

$$P_O = O_r(r) \cdot O_a(\theta) \quad (\text{III.4})$$

avec :

$$\begin{cases} O_r(r) = \begin{cases} 1 - \left(\frac{r - R}{\varepsilon} \right)^2 & \text{Si } r \in [R - \varepsilon, R + \varepsilon] \\ 0 & \text{Sinon} \end{cases} \\ O_a(\theta) = 1 - \left(\frac{2\theta}{\Omega} \right)^2 & \text{Si } \theta \in \left[-\frac{\Omega}{2}, \frac{\Omega}{2} \right] \end{cases} \quad (\text{III.5})$$

La figure III.13 illustre ces distributions de probabilité pour un capteur à ultrasons fournissant la mesure R , et la figure III.14 représente ces probabilités en trois dimensions.

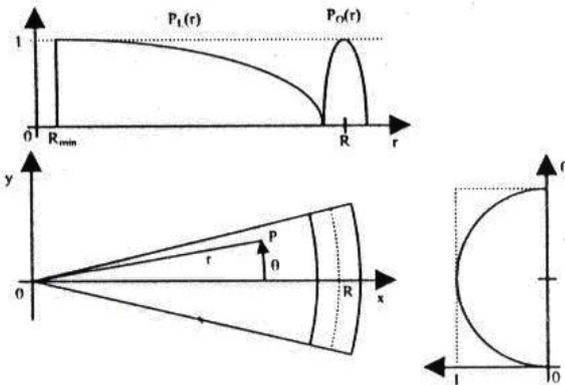


Figure III.13 : Profils de probabilité appliqués au capteur à ultrasons [Elf87a].

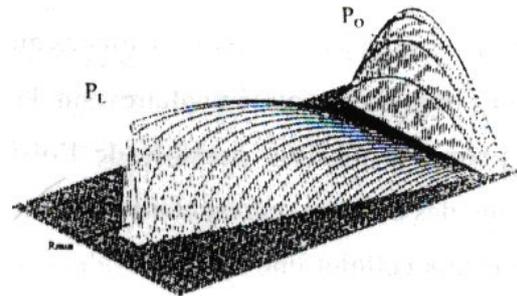


Figure III.14 : Vue des profils de probabilité en 3D.

Un modèle gaussien [Bul96] en deux dimensions (Figure III.15) représentant uniquement l'espace occupé rejoint ce modèle:

$$P\left[\frac{r}{d}, \theta\right] = \frac{1}{2\pi\sigma_r\sigma_\theta} e^{-\left[\frac{1}{2} \left(\frac{(r-d)^2}{\sigma_r^2} + \frac{\theta^2}{\sigma_\theta^2} \right) \right]} \quad (\text{III.6})$$

où σ_r , et σ_θ représentent les écarts types des erreurs radiales et angulaires du capteur.

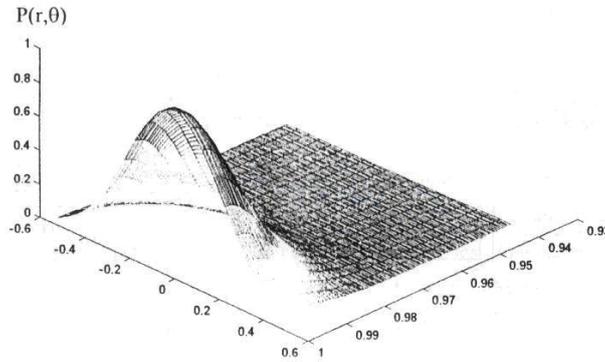


Figure III.15 : Exemple de modélisation d'un capteur à ultrasons par une gaussienne ($\sigma_r = 1cm$, $\sigma_\theta = 12^\circ$ et $d = 1m$) pour $r \in [0,1]$.

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 lié 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.

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 partant de l'idée n'ayant aucune informations complémentaires sur la mesure ultrasonore, notre contribution a consisté à considérer 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 modèle utilise cette idée en décrémentant de 1 les valeurs CV des cellules correspondant à l'espace libre jusqu'à un seuil minimum défini ($Min_Vote_Seuil=0$), et en incrémentant de 3 les valeurs CV des cellules de l'arc de cercle représentant l'obstacle jusqu'à un seuil maximum défini (Max_Vote_Seuil). Les valeurs des cellules restantes ne sont pas modifiées.

Le modèle du capteur à ultrasons ainsi obtenu et sa grille correspondante sont représentés par la figure III.16.

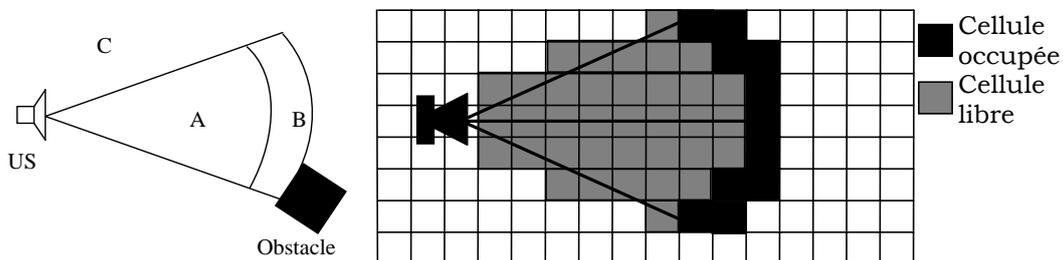


Figure III.16 : Modélisation du capteur à ultrasons.

La règle de mise à jour des valeurs CV des cellules de la grille selon la figure III.16, est exprimée comme suit :

$$CV(X) = \begin{cases} CV(X) + 3 & \text{if } X \in B \\ CV(X) - 1 & \text{if } X \in A \\ CV(X) & \text{if } X \in C \end{cases} \quad (III.7)$$

$CV_{Min} \leq CV(X) \leq CV_{Max}$

Ce modèle présente l'avantage d'éviter le traitement des mesures à l'aide de lois de probabilités lourdes à gérer.

III.2.3. Système de vision.

Connaissant le modèle de projection de chaque caméra et la relation spatiale entre les caméras du banc stéréoscopique (§I.1.2.5), on peut calculer les coordonnées 3D d'un point à partir de ses projections dans les différentes images.

III.2.3.1. Calcul des paramètres du modèle géométrique d'une Caméra : Méthode choisie.

Calibrer une caméra va consister à estimer la matrice de projection M (équations I.8 et I.9). Il suffit de disposer d'un ensemble de points de référence $P_i=(x_i, y_i, z_i)$ (appelés aussi points d'intérêt) appartenant au repère absolu et de leur projection (u_i, v_i) dans l'image numérisée [Cro97].

D'après l'équation (I.8), chaque correspondance donne deux équations :

$$u_i = \frac{m_{11} x_i + m_{12} y_i + m_{13} z_i + m_{14}}{m_{31} x_i + m_{32} y_i + m_{33} z_i + m_{34}} \quad (\text{III.8})$$

$$v_i = \frac{m_{21} x_i + m_{22} y_i + m_{23} z_i + m_{24}}{m_{31} x_i + m_{32} y_i + m_{33} z_i + m_{34}} \quad (\text{III.9})$$

Ces équations sont linéaires par rapport aux coefficients m_{ij} de la matrice de projection M , donc au moins six (06) points non coplanaires suffisent pour les déterminer.

Les équations (III.8) et (III.9) peuvent se réécrire comme une combinaison linéaire des m_{ij} :

$$m_{11} x_i + m_{12} y_i + m_{13} z_i + m_{14} - u_i m_{31} x_i - u_i m_{32} y_i - u_i m_{33} z_i = u_i m_{34} \quad (\text{III.10})$$

$$m_{21} x_i + m_{22} y_i + m_{23} z_i + m_{24} - v_i m_{31} x_i - v_i m_{32} y_i - v_i m_{33} z_i = v_i m_{34} \quad (\text{III.11})$$

On obtient donc $2n$ équations pour n points et on peut écrire ces équations sous forme matricielle :

$$\begin{pmatrix} \cdot & \cdot \\ x_i & y_i & z_i & 1 & 0 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i z_i \\ 0 & 0 & 0 & 0 & x_i & y_i & z_i & 1 & -v_i x_i & -v_i y_i & -v_i z_i \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{pmatrix} = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ u_i m_{34} \\ v_i m_{34} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} \quad (\text{III.13})$$

ou, sous forme plus condensée :

$$Kx = u \quad (III.14)$$

Le système précédent étant homogène (M est définie à un facteur d'échelle près), il faut utiliser une contrainte supplémentaire. Nous utilisons pour cela la méthode de calibrage décrite par Faugeras et Toscani [Fau86,Cro97].

Si l'on écrit M sous la forme suivante :

$$M = \begin{pmatrix} m_1 & m_{14} \\ m_2 & m_{24} \\ m_3 & m_{34} \end{pmatrix} \quad (III.15)$$

$$\text{où } m_i = (m_{i1} \ m_{i2} \ m_{i3}),$$

$$\text{et si l'on pose la contrainte suivante : } \|m_3\| = 1 \quad (III.16)$$

alors le système d'équations (III.13) peut s'écrire sous la forme suivante :

$$Bx_9 + Cx_3 = 0 \quad (III.17)$$

où :

$$B = \begin{pmatrix} \cdot & \cdot \\ x_i & y_i & z_i & 1 & 0 & 0 & 0 & 0 & -u_i \\ 0 & 0 & 0 & 0 & x_i & y_i & z_i & 1 & -v_i \\ \cdot & \cdot \end{pmatrix} \quad (III.18)$$

$$C = \begin{pmatrix} \cdot & \cdot \\ -u_i x_i & -u_i y_i & -u_i z_i & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -v_i x_i & -v_i y_i & -v_i z_i & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad (III.19)$$

$$x_9 = (m_1 \ m_{14} \ m_2 \ m_{24} \ m_3)^t \quad (III.20)$$

$$x_3 = m_3^t \quad (III.21)$$

Le critère à minimiser étant $\|Bx_9 + Cx_3\|^2$ sous la contrainte $\|m_3\|^2 = 1$, après calculs, l'estimation des coefficients de la matrice de projection M est donnée par les étapes suivantes [Hor93] :

1. Construction d'une matrice D telle que :

$$D = C^t C - C^t B (B^t B)^{-1} B^t C \quad (III.22)$$

2. Calcul des valeurs et vecteurs propres de D .
3. x_3 est le vecteur propre correspondant à la plus petite valeur propre.
4. Normalisation de x_3 :

$$x_3 \leftarrow \frac{x_3}{\|x_3\|} \tag{III.23}$$

5. Calcul de x_9 :

$$x_9 = -(B^t B)^{-1} B^t C x_3 \tag{III.24}$$

6. Si $m_{34} \leq 0$ alors $M \leftarrow (-M)$.

A partir des coefficients m_{ij} de la matrice de projection M et grâce aux équations (I.17) on peut donc déterminer les paramètres intrinsèques et extrinsèques de la caméra.

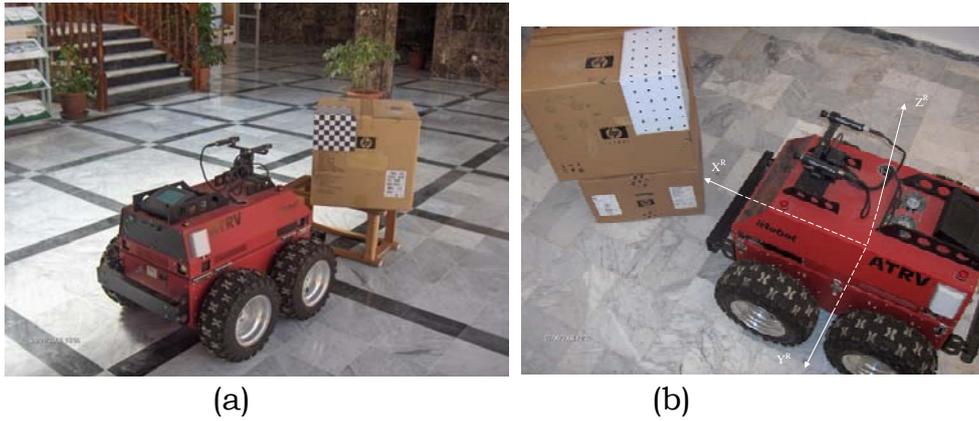


Figure III.17 : Mise en oeuvre du calibrage du banc stéréoscopique du système mobile ATRV2 : (a) en utilisant la première mire pour le calibrage fort, (b) en utilisant la deuxième mire pour le calibrage faible.

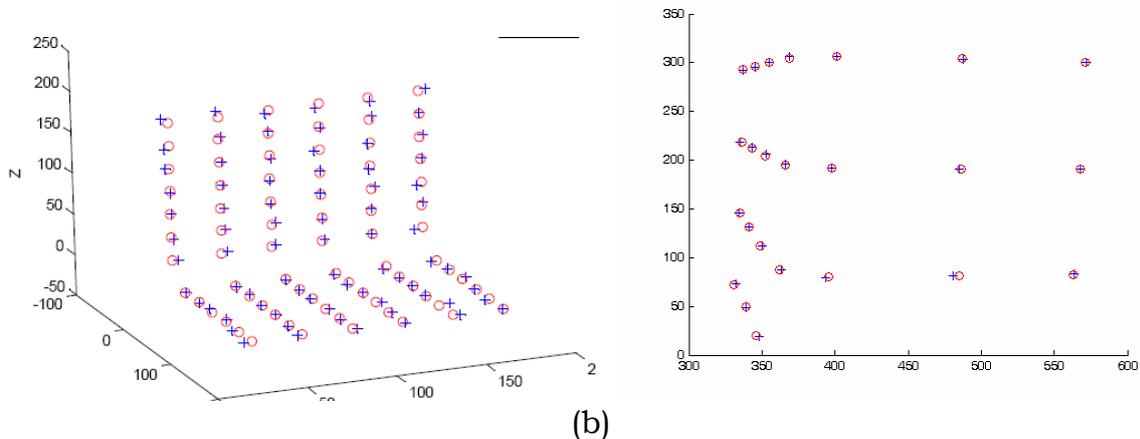
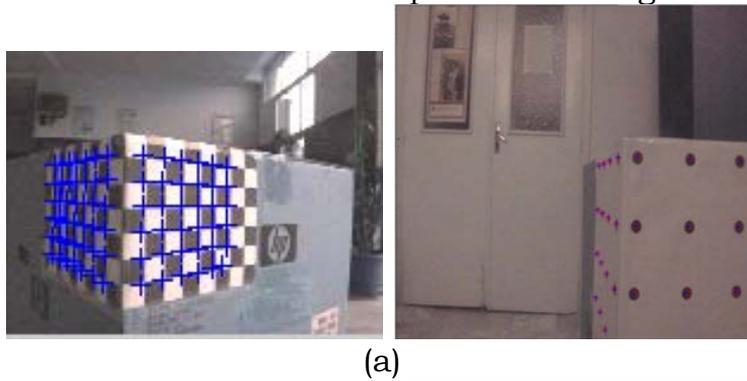


Figure III.18: Reconstruction 3D de la mire : (a) les deux mires utilisées, (b) la reconstruction 3D des deux mires dans le repère caméra gauche.

III.2.3.2. Mise en œuvre du calibrage : Application au banc stéréoscopique du système mobile ATRV2.

Calibrer le banc stéréoscopique du système mobile ATRV2 revient à calibrer chaque caméra indépendamment en utilisant la même mire.

La mire de calibrage fournit des points de calibrage dont la position doit être connue avec une très grande précision dans un repère absolu. La figure III.17 montre les mires de calibrage utilisées : la première pour le calibrage fort est constituée d'un ensemble de petits carrés noirs et blancs de 2.8 cm de longueur pour chacun de leur côté, et la seconde pour le calibrage faible est constituée d'un ensemble de petits cercles noirs de 1.0 cm de diamètre et espacés de 05 cm. Ces motifs ont été choisis pour que leur projection dans l'image numérisée puisse être mesurée avec une grande précision.

Le calibrage commence par le placement en différentes positions de la mire pour qu'elle soit perçue à la fois par les deux caméras. Le calibrage fort a été réalisé sur des paires d'images de 160x120 pixels de résolution, tandis que le calibrage faible a été réalisé sur des paires d'images de 600x560 pixels de résolution. Les corrections de la distorsion radiale des deux caméras n'ont pas été prises en compte pendant le processus de calibrage. Les résultats issus du calibrage sont présentés en annexe C.

Pour valider les résultats du calibrage du banc stéréoscopique du système mobile ATRV2, nous avons reconstruit les points 3D des deux mires utilisées et comparé les valeurs des points reconstruits avec leur vraie valeur acquise lors du calibrage (Figure III.18). Nous remarquons qu'il y a une petite différence entre la position des points réels et les points calculés. Cela peut s'expliquer par le mode manuel de prélèvement des points 3D/2D des deux mires utilisées pour l'estimation des paramètres intrinsèques et extrinsèques des caméras gauche et droite du banc stéréoscopique.

Les points 3D de l'environnement obtenus lors des acquisitions sont filtrés avant d'être sauvegardés dans le modèle de l'environnement choisi. Le filtrage se fait comme suit :

1. Tous les points dont la profondeur (suivant l'axe des X) est supérieure à un seuil donné sont éliminés.
2. Tous les points dont la hauteur (suivant l'axe des Z) est inférieure à un seuil donné sont éliminés.
3. Les points 3D restants sont sauvegardés dans le modèle de la manière suivante : Tous les points sont projetés en un seul point sur le sol (le plan XY) et la valeur CV de la cellule correspondante prendra comme valeur le nombre de points projetés.

III.2.4. Fusion des données capteurs.

Les systèmes qui utilisent plusieurs types de capteurs sont de plus en plus présents dans la littérature. Ils permettent une précision accrue, une plus grande portée de détection et une fiabilité supérieure. C'est pour cette raison que l'usage de différents types de capteurs est maintenant fréquent dans des applications comme la robotique mobile. Ses systèmes comportent de nombreux capteurs qui

acquièrent de l'information complémentaire ou redondante afin d'obtenir une perception plus complète et plus fiable de leur environnement. Mais la façon de gérer l'information provenant de tous ces capteurs demeure une préoccupation.

A cet effet, le système de perception du système mobile *ATRV2* (banc stéréoscopique et douze capteurs US) a nécessité une fusion. La méthode de fusion choisie est une méthode non probabiliste basée sur la théorie des possibilités. Ce choix est justifié par le fait que le modèle de l'environnement de navigation choisi n'utilise pas des fonctions de probabilité pour la reconstruction.

III.2.4.1. Calibrage Capteurs à ultrasons-Banc stéréoscopique.

Le calibrage des systèmes multicapteurs est une étape primordiale pour représenter les données provenant des différents capteurs dans un même repère, ce qui est indispensable avant toute analyse ou fusion de ces données. Le banc stéréoscopique et les capteurs ultrasonores sont montés sur le système mobile *ATRV2* dans un but de coopération et de fusion, leur calibrage est ainsi devenu une nécessité. Le banc stéréoscopique fournit les informations d'intensité sous la forme d'une paire d'images, alors que les capteurs ultrasonores apportent les informations de profondeur sous la forme d'un vecteur de profondeur.

Le calibrage calcule les transformations rigides entre le repère lié au banc stéréoscopique et celui lié à chaque capteur ultrasonore (US). La connaissance de ces transformations permet de projeter les points 2D des repères capteurs ultrasonores dans le repère banc stéréoscopique.

Soient R_{US}^i le repère lié au capteur ultrasonores i (avec $i=1$ à 12) et R_k^C le repère lié à la caméra k (avec $k=G$ ou D) (Figure III.19). Les matrices de passage M_{US}^i et M_k^C des repères R_{US}^i et R_k^C au repère local associé au système mobile *ATRV2* $R^R(O^R, X^R, Y^R)$, sont calculées de telle sorte que pour un point P de

coordonnées $P_{US}^i = [x_{US}^i, y_{US}^i, z_{US}^i]^t$ dans R_{US}^i , et de coordonnées

$P_k^C = [x_k^C, y_k^C, z_k^C]^t$ dans R_k^C , on aura :

$$P^R = \begin{pmatrix} x^R \\ y^R \\ z^R \end{pmatrix} = \begin{cases} M_{US}^i P_{US}^i \\ M_k^C P_k^C \end{cases} \quad (III.25)$$

avec P^R un point de coordonnées $P^R = [x^R, y^R, z^R]^t$ dans R^R ,

La matrice de passage entre les repères capteurs ultrasonores et les repères caméras est donnée par le système suivant :

$$P_k^C = M_k^{C^{-1}} M_{US}^i P_{US}^i \quad (III.26)$$

L'estimation de la matrice de passage M_k^C est obtenue en faisant un calibrage faible et dont les paramètres sont présentés en annexe C.

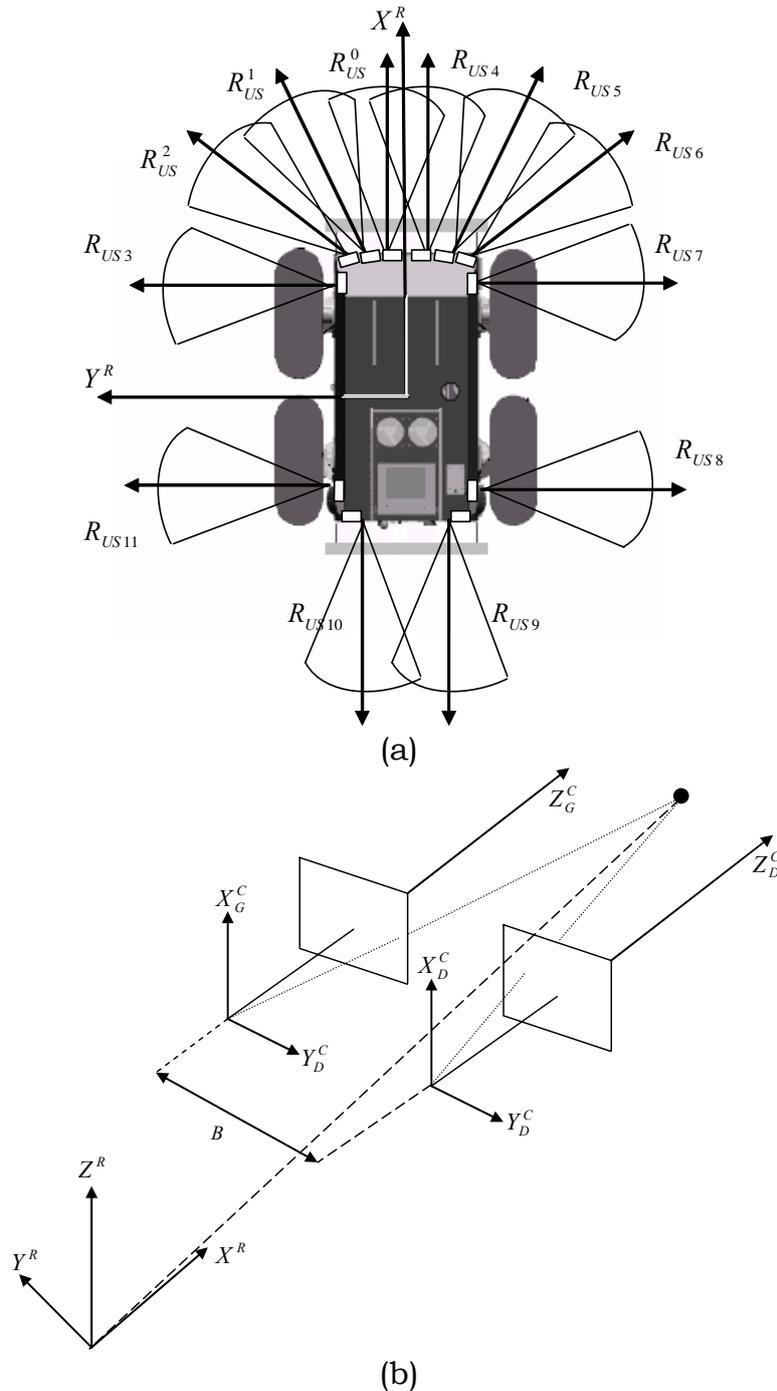


Figure III.19 : (a) Positions et orientations des 12 capteurs ultrasonores embarqués de l'ATRV2, (b) Modèle géométrique du banc stéréoscopique : vue de perspective.

III.2.4.2. La fusion.

Sur la figure III.19, nous remarquons bien l'existence d'un champ de vision commun entre les différents capteurs embarqués du système mobile ATRV2, que ce soit du même type (capteurs US-US, ou Caméras gauche-droite) ou de type différents (capteurs US-Banc stéréoscopique). Dans notre cas, nous nous intéressons au champ de vision commun entre les capteurs ultrasonores et le banc stéréoscopique. Le champ de vision du banc stéréoscopique est à l'intérieur de celui des capteurs à ultrasons avec une certaine limitation en profondeur (Figure III.20).

Grâce au calibrage et au calcul des matrices M_{US}^i et M_k^C , nous sommes capables d'exprimer toutes les données capteurs (US-Banc) du champ commun dans un même repère, et en fusionnant ces données, nous obtiendrons une perception plus complète et plus fiable de l'environnement de navigation du système mobile ATRV2.

Le champ de vision commun est discrétisé en cellules conformément au modèle de l'environnement choisi. Dans ce dernier, chaque cellule possède une valeur de certitude CV qui représente la confiance de l'algorithme dans l'existence d'un obstacle à cet endroit. Cette valeur est modifiée après comparaison, à chaque acquisition, dès que les capteurs ultrasonores ou le banc stéréoscopique détectent un nouvel obstacle.

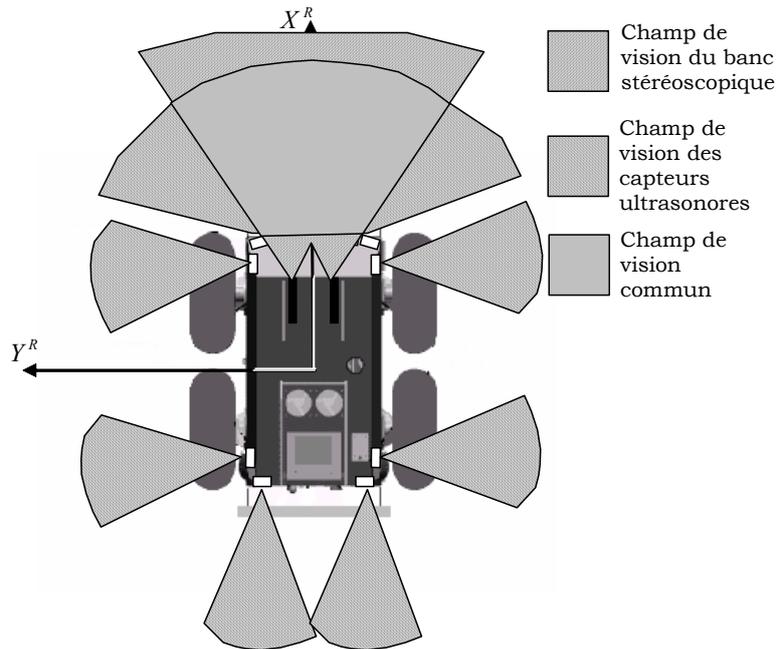


Figure III.20 : Le champ de vision commun entre les capteurs à ultrasons et le banc stéréoscopique.

La méthode de fusion des données US-banc choisie définit une distribution de possibilité pour chaque classe, occupée ou libre, et est calculée pour chaque cellule du champ de vision commun, et donne son degré de possibilité d'appartenir à une classe donnée (libre ou occupée).

Soit π_{US} et π_{Banc} deux distributions de possibilité du champ de vision commun dans $[0,1]$ associées aux capteurs US et au banc stéréoscopique.

Les ensembles de discernement $\Omega_{capteur}$ donnant les classes de chaque capteur, sont les suivants :

$$\Omega_{US} = \{Occupé_{US}, Libre_{US}\} \quad (III.27)$$

$$\Omega_{Banc} = \{Occupé_{Banc}, Libre_{Banc}\} \quad (III.28)$$

L'espace de discernement Ω est l'ensemble de toutes les classes envisageables et possibles. Dans notre cas, l'espace de discernement est le suivant :

$$\Omega = \{F_1, F_2, F_3, F_4\} \quad (III.29)$$

avec :

$$\begin{aligned}
 F_1 &= \text{Occupé}_{US} \cap \text{Occupé}_{Banc} \\
 F_2 &= \text{Occupé}_{US} \cap \text{Libre}_{Banc} \\
 F_3 &= \text{Libre}_{US} \cap \text{Occupé}_{Banc} \\
 F_4 &= \text{Libre}_{US} \cap \text{Libre}_{Banc}
 \end{aligned}
 \tag{III.30}$$

Le tableau ci-dessous récapitule les différents états possibles des cellules appartenant au champ de vision commun des capteurs US-Banc ainsi que la décision à prendre pour la fusion des données.

Tableau III.1 : Récapitulatif des états possibles des cellules appartenant au champ de vision commun pour la fusion.

Type de capteur	Etats de la cellule			
	occupé	occupé	libre	libre
Capteurs US	occupé	occupé	libre	libre
Banc stéréoscopique	occupé	libre	occupé	libre
Ω	F_1	F_2	F_3	F_4
Décision	occupé	Combinaison adaptative	Combinaison adaptative	libre

Si une cellule est perçue libre en même temps par les capteurs ultrasonores et le banc stéréoscopique, alors cet état est confirmé.

Si une cellule est perçue occupée en même temps par les capteurs ultrasonores et le banc stéréoscopique, alors cet état est confirmé. Sa valeur CV prendra la valeur résultante de la combinaison adaptative (maximum de possibilité) en utilisant l'équation (I.25).

S'il y a une différence de perception d'état (libre-occupée ou occupée-libre) d'une cellule donnée par les capteurs ultrasonores et le banc stéréoscopique, la valeur CV de cette cellule prendra la valeur résultante de la combinaison adaptative des deux données capteurs en utilisant la même équation (I.25), déterminant ainsi son état.

Pour mieux comprendre le principe, voici un exemple :

Soient les grilles A et B de la Figure III.21 les portions de l'environnement perçues respectivement par les capteurs ultrasonores et le banc stéréoscopique. La grille C est la grille résultant de la fusion des données obtenue par combinaison adaptative des grilles A et B en utilisant l'équation (I.25) (voir tableau III.2).

30 a	37 b	29 c	20 a	30 b	35 c	66 a	100 b	96 c
0 d	20 e	10 f	0 d	20 e	0 f	0 d	66 e	10 f
0 g	0 h	20 i	0 g	5 h	20 i	0 g	5 h	66 i
A			B			C		

Figure III.21 : Exemple de fusion des données capteurs basée sur la combinaison adaptative.

Pour chaque valeur CV d'une cellule des grilles A et B, on associe une probabilité P représentant son état, calculée comme suit :

$$P = \frac{CV}{Max_Vote_Seuil} \quad (III.31)$$

La mesure de possibilité de cette cellule prendra donc la valeur suivante :

$$\pi_{US \text{ ou } Banc} = P \quad (III.32)$$

En cas de différence de perception d'état d'une cellule donnée par les capteurs ultrasonores et le banc stéréoscopique, un opérateur adaptatif $\pi_{adap} = \pi_{US, Banc}$ est utilisé et calculé de la façon suivante :

$$\pi_{US, Banc} = \max \left(\frac{\min(\pi_{US}, \pi_{Banc})}{h(\pi_{US}, \pi_{Banc})}, \min(\max(\pi_{US}, \pi_{Banc}), 1 - h(\pi_{US}, \pi_{Banc})) \right) \quad (III.33)$$

Pour un maximum de vote $Max_Vote_Seuil=100$ choisi expérimentalement, et d'après l'équation (I.24), la valeur de $h(\pi_{US}, \pi_{Banc}) = 0.30$ (voir tableau III.2).

Les valeurs CV (ou CV_{adap}) des cellules de la grille C sont obtenues à partir des valeurs de $\pi_{US, Banc}$ correspondantes de la manière suivante :

$$CV_{adap} = \pi_{US, Banc} * 100 \quad (III.34)$$

La nouvelle valeur de CV représente donc une nouvelle information plus riche, plus précise et plus fiable sur l'état de la cellule.

Tableau III.2 : Résultats de la fusion des données capteurs basée sur la combinaison adaptative pour les grilles A et B.

Cellule	Classes	Etat	π_{US}	π_{Banc}	$Min(\pi_{US}, \pi_{Banc})$	$Max(\pi_{US}, \pi_{Banc})$	$1-h(\pi_{US}, \pi_{Banc})$	$Min[1-h(\pi_{US}, \pi_{Banc}), Max(\pi_{US}, \pi_{Banc})]$	$\frac{Min(\pi_{US}, \pi_{Banc})}{h(\pi_{US}, \pi_{Banc})}$	$\pi_{US, Banc}$
a	F ₁	o/o	0.30	0.20	0.20	0.30	0.7	0.3	0.66	0.66
b	F ₁	o/o	0.37	0.30	0.30	0.37	0.7	0.37	1	1
c	F ₁	o/o	0.29	0.35	0.29	0.35	0.7	0.35	0.96	0.96
d	F ₄	1/1	0	0	0	0	0.7	0	0	0
e	F ₁	o/o	0.20	0.20	0.20	0.20	0.7	0.2	0.66	0.66
f	F ₂	o/1	0.10	0	0	0.10	0.7	0.1	0	0.1
g	F ₄	1/1	0	0	0	0	0.7	0	0	0
h	F ₃	1/o	0	0.05	0	0.05	0.7	0.05	0	0.05
i	F ₁	o/o	0.20	0.20	0.20	0.20	0.7	0.2	0.66	0.66

cellule correspondant au centre de gravité du système mobile. Les figures III.23c-III.25c sont identiques aux précédentes, un rectangle montrant la position et l'orientation du système mobile a été ajouté en utilisant les données odométriques et les dimensions exactes du système mobile.

Par contre les figures III.26- III.28 montrent la reconstruction de l'environnement lorsque le système mobile est en mouvement (déplacement non autonome). De même, les figures III.26a- III.28a montrent l'image réelle du mouvement du système mobile (la bande noire montre le trajet effectué par le système mobile), les figures III.26b- III.28b montrent le résultat de la reconstruction avec les points verts correspondant aux cellules de la grille occupées par les obstacles et les points rouges indiquant les cellules correspondant au centre de gravité du système mobile en mouvement. Dans les figures III.26c- III.28c des rectangles schématisant le système mobile ont été ajoutés pour décrire son déplacement dans la scène en utilisant les données odométriques et ses dimensions exactes.

Grâce à ces expérimentations, la carte de l'environnement est reconstruite et elle est jugée acceptable, mais toutefois, les problèmes inhérents à l'utilisation des capteurs ultrasonores tels que les réflexions multiples et la diaphonie mènent à des mesures aberrantes faussant ainsi le position des obstacles. En plus, nous remarquons bien que certaines arêtes rectilignes des obstacles ont été perçues courbées par la couronne ultrasonore du système mobile *ATRV2*. Ceci est dû à la prise en compte de tout l'arc de cercle du cône représentant le modèle du capteur ultrasonore adopté.

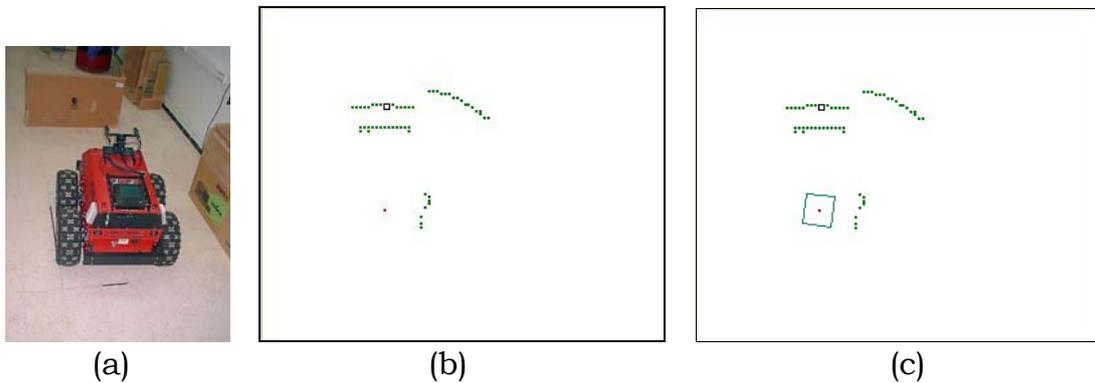


Figure III.23 : Situation réelle statique 1.

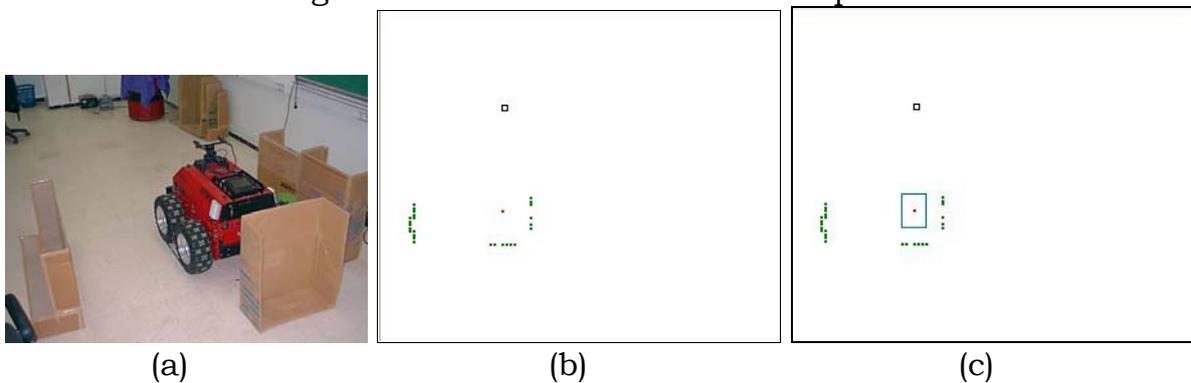


Figure III.24 : Situation réelle statique 2.

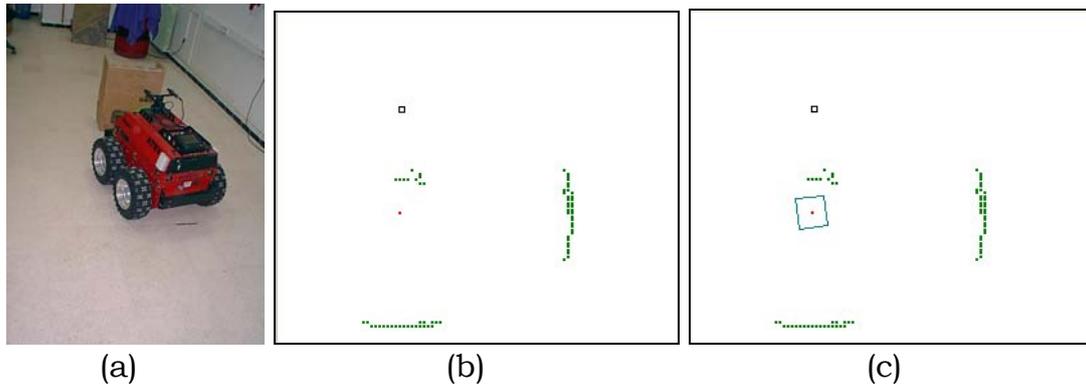


Figure III.25 : Situation réelle statique 3.

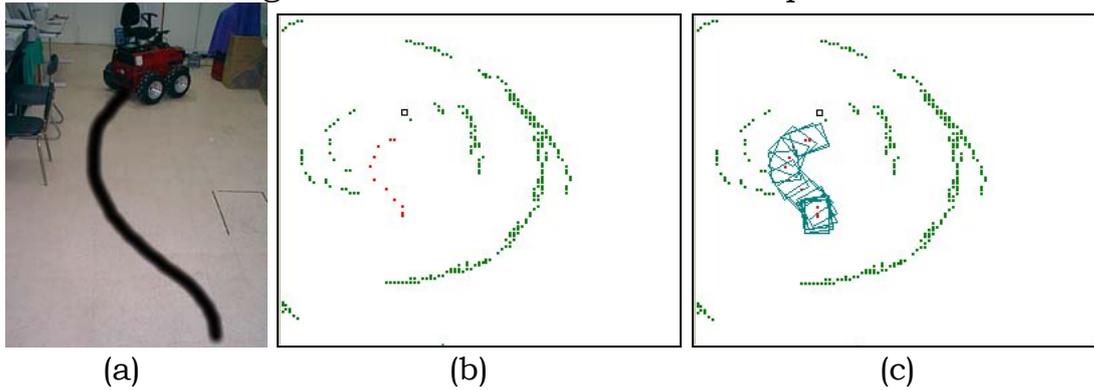


Figure III.26 : Première situation de mouvement du système mobile.

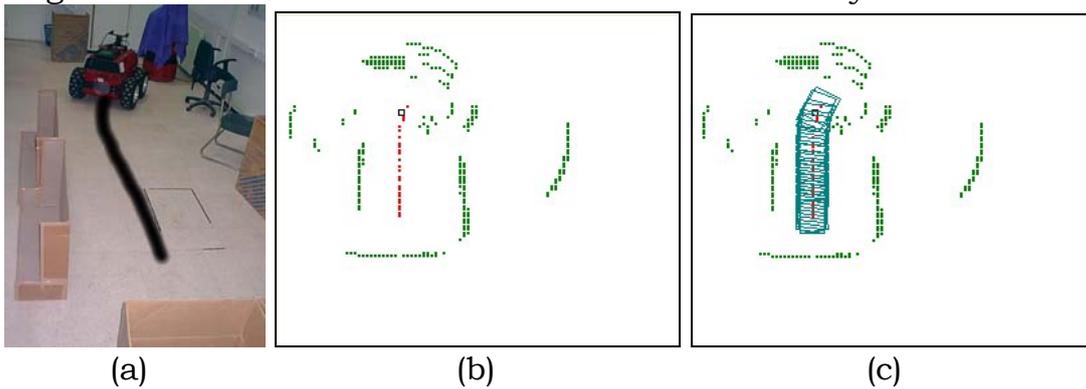


Figure III.27 : Deuxième situation de mouvement du système mobile.

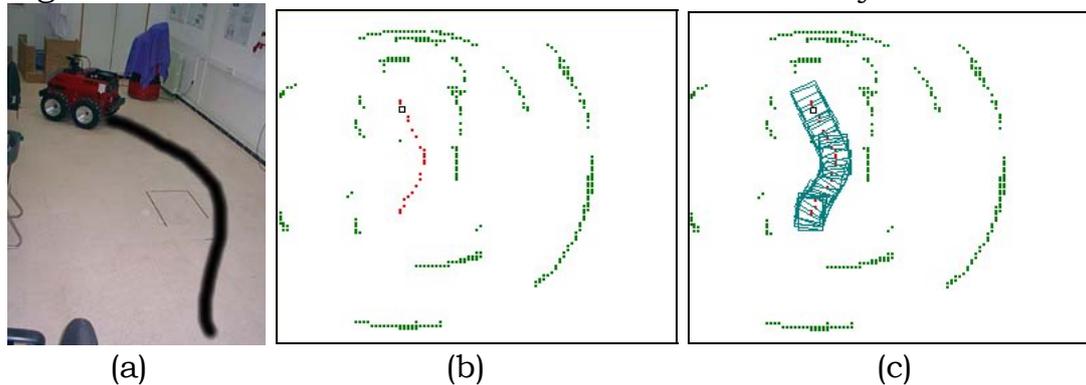


Figure III.28 : Troisième situation de mouvement du système mobile.

III.3.2. Construction basée sur les données du banc stéréoscopique.

Nous présentons dans cette partie la méthode de reconstruction tridimensionnelle de la scène observée par le banc stéréoscopique du système mobile *ATRV2*.

L'algorithme de mise en correspondance que nous avons implémenté est un algorithme de recherche directe (§A.3.2) [Cro97, Dev97] considéré comme l'un des algorithmes d'appariement les plus triviaux. Son principe est défini de la manière suivante :

Nous définissons pour chaque point p_1 de l'image de référence une zone de recherche dans l'image objet (Figure III.29). Nous utilisons une fenêtre de corrélation fixe dans l'image de référence centrée sur le pixel p_1 , ainsi qu'une fenêtre glissante dans l'image objet qui va parcourir toute la zone de recherche (Figure III.30).

Nous calculons à chaque fois le score entre la fenêtre fixe dans l'image de référence et la fenêtre glissante dans l'image objet. L'ensemble des scores obtenu constitue la surface de corrélation associée au point p_1 .

Le point p_2 choisi sera celui qui minimise la mesure de dissemblance ou maximise la mesure de ressemblance.

Nous pouvons remarquer qu'il n'y a aucun autre traitement à faire. Cela montre la simplicité de l'algorithme, mais en même temps, ce dernier possède des inconvénients. L'approche directe reste sensible au bruit vu que lors d'un conflit (lorsque nous avons deux couples présentant le même coût), l'algorithme choisit aveuglément le premier appariement. Cette stratégie peut conduire à des faux appariements.

Des tests préliminaires sont effectués sur plusieurs cas possibles en faisant varier les principaux paramètres tels que le critère de corrélation, la fenêtre de corrélation et l'intervalle de disparité.

Les figures III.31 et III.32 montrent le résultat de l'appariement et de la reconstruction 3D de la scène observée. La figure III.31, montre les résultats obtenus en utilisant tous les points (en niveaux de gris) de la paire stéréo pour l'appariement et la reconstruction 3D, par contre la figure III.32 montre les résultats obtenus en utilisant uniquement les points contours de la paire stéréo pour l'appariement et la reconstruction 3D.

En tenant compte des points contours de la paire stéréo, l'exécution de l'algorithme d'appariement est nettement plus rapide.

A partir des images de la disparité des figures III.31 et III.32, nous remarquons que la qualité de l'appariement est moyenne. Cette dernière influence beaucoup la qualité de la reconstruction 3D de la scène observée. Ceci est due premièrement aux inconvénients de la méthode cités ci-dessus et deuxièmement à la qualité de l'image (l'utilisation d'images un peu floues rend l'opération d'appariement et la reconstruction 3D quasi impossibles).

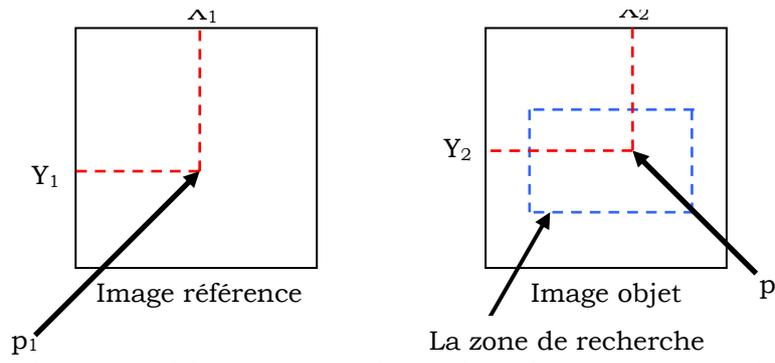


Figure III.29 : La zone de recherche: cas général.

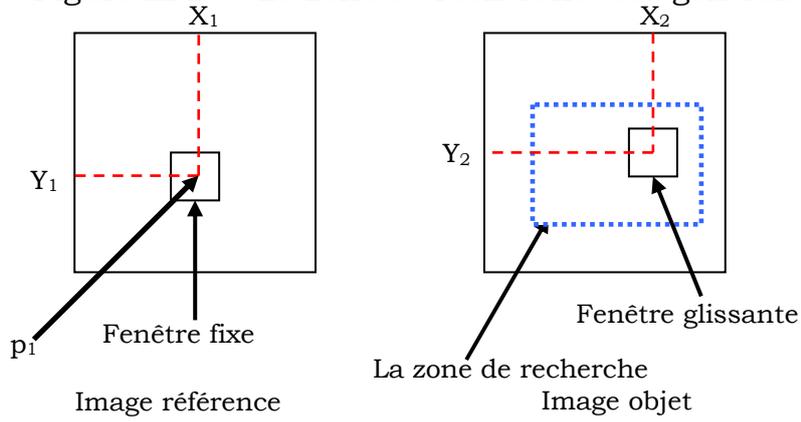


Figure III.30 : Principe de la recherche directe.

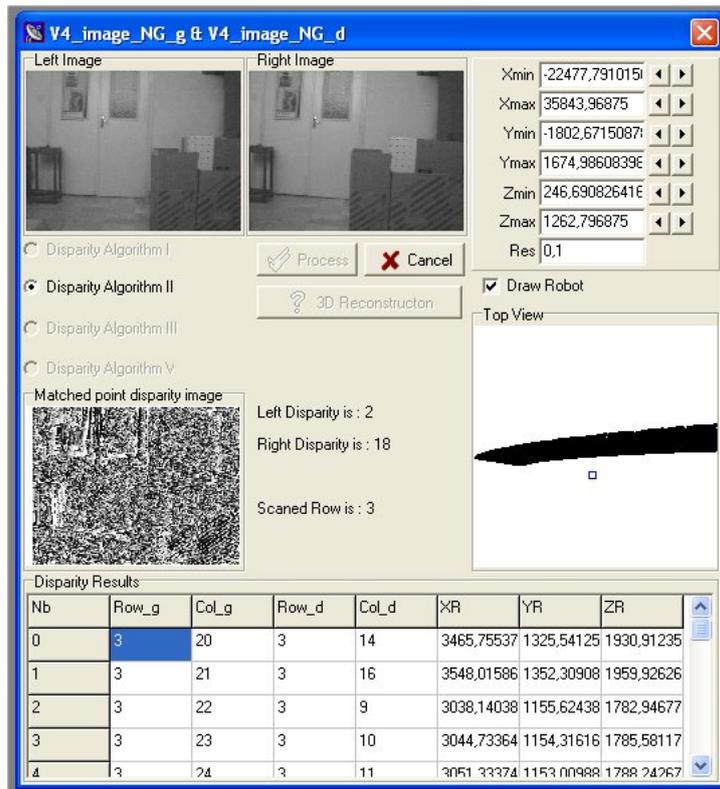


Figure III.31 : Résultats d'appariement et la reconstruction 3D en utilisant tous les points de la paire stéréo.

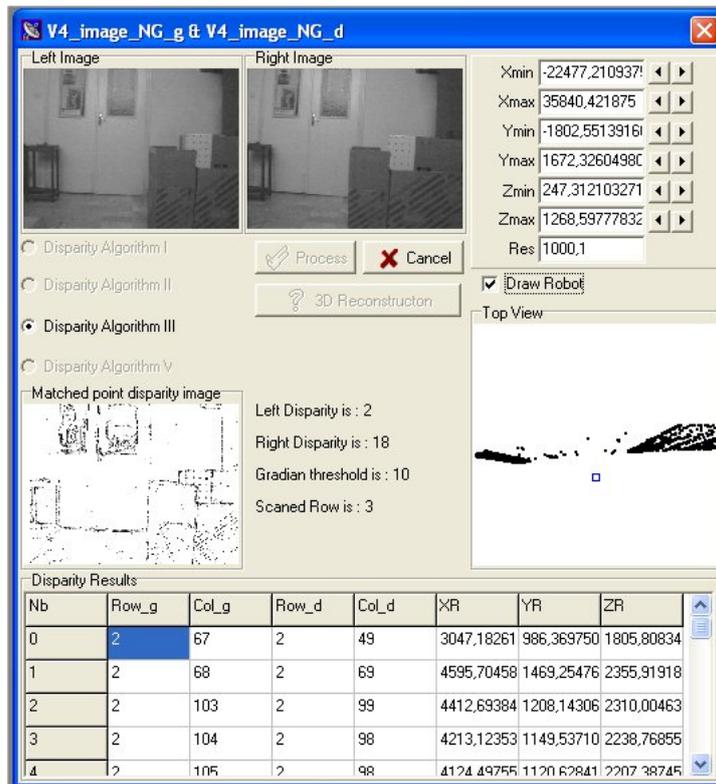


Figure III.32: Résultats d'appariement et la reconstruction 3D en utilisant les points contours de la paire stéréo.

III. 4. Conclusion.

Nous avons décrit dans ce chapitre le système de perception utilisé pour la construction du modèle de l'environnement du système mobile *ATRV2*. Il s'agit d'une ceinture de douze capteurs ultrasonores et un banc stéréoscopique composé de deux caméras *CCD* couleurs montées sur une tourelle *Pan Tilt*. Les capteurs ultrasonores fournissent une perception panoramique complète à 360°, tandis que le banc stéréoscopique fournit des informations de l'environnement uniquement sur l'avant du système mobile *ATRV2*.

Les calibrages du banc stéréoscopique et banc stéréoscopique-Capteurs ultrasonores ont été réalisés. Pour le premier, nous avons utilisé les calibrages faible et fort en utilisant deux mires de calibrage. Pour le second, le calibrage est utilisé pour trouver la transformation entre les repères liés aux capteurs ultrasonores et le banc stéréoscopique, une étape essentielle avant de pouvoir fusionner leurs données. Les données appartenant au champ de vision commun entre ces deux derniers sont fusionnées pour compléter la perception de l'environnement du système mobile *ATRV2*. La méthode de fusion des données capteurs choisie est une méthode non probabiliste basée sur la théorie des possibilités.

La carte de l'environnement du système mobile est déterminée au moyen du système de perception de ce dernier. Nous avons choisi les grilles de certitude comme mode de représentation des mesures et de l'environnement, car elles sont exploitables dans tout type d'environnement sans nécessiter d'aménagements particuliers, et peuvent être utilisées à partir des mesures issues de plusieurs types de capteurs.

Les résultats expérimentaux de la reconstruction de la carte de l'environnement lors des déplacements du système mobile *ATRV2* à partir des situations réelles ont été jugés acceptables, mais toutefois, les problèmes inhérents à l'utilisation des capteurs du système de perception, tels que les réflexions multiples et la diaphonie pour les capteurs ultrasonores, et la sensibilité aux bruits et la robustesse de l'algorithme d'appariement utilisé pour la vision, mènent parfois à des erreurs de mesure faussant la position des obstacles et diminuant ainsi la qualité de la reconstruction.

Dans le chapitre suivant, après une présentation des caractéristiques spécifiques du système mobile *ATRV2*, son architecture matérielle et logicielle, ainsi que sa géométrie, nous proposons notre travail concernant sa localisation et sa navigation sans collision dans un environnement qui peut être inconnu a priori.

Chapitre IV

Localisation et navigation
proposées pour le système
mobile *ATRV2*.

Après avoir présenté succinctement les méthodes de localisation et de navigation les plus utilisées en robotique mobile, nous proposons dans ce chapitre les méthodes de localisation et de navigation dotant le système mobile ATRV2 d'une capacité de réaction et d'action dans un environnement qui peut être inconnu a priori.

Nous commençons d'abord ce chapitre, par présenter les caractéristiques spécifiques du système mobile ATRV2, ensuite, nous présentons la méthode de localisation utilisée basée sur le filtre de *Kalman* étendu (EKF). Cette méthode fusionne les données odométriques (modèle du système) et les données du système de perception embarqué (mesure) pour corriger régulièrement l'estimation relative de la position du système mobile due à l'erreur cumulative engendrée par les capteurs odométriques. Pour déterminer la mesure, la méthode utilise la mise en correspondance des grilles tout en estimant sa zone de recherche et la position autour de laquelle elle effectue la recherche de la position estimée.

A la fin de ce chapitre, nous proposons notre méthode de navigation appelée DVFF. Elle combine une méthode de planification de trajectoire globale basée sur l'algorithme D^* , et une méthode réactive non stop basée sur le principe des champs de forces virtuelles VFF. Cette méthode de navigation utilise les valeurs courantes des capteurs ultrasonores embarqués (sur une petite fenêtre temporelle), et les données provenant d'un modèle interne, pour décider de l'action à effectuer. Elle permet au système mobile ATRV2 de se mouvoir sans crainte de collisions avec les obstacles de son environnement.

IV.1. Le système mobile ATRV2 [Iro01].

Le système mobile ATRV2 (All Terrain Robot Vehicle 2) que nous utilisons est une plate forme expérimentale construite par la société *iRobot* (Figures IV.1-4) pouvant se déplacer sur un terrain accidenté et possédant quatre roues motrices différentielles (4-wheel differential-drive skid-steering configuration) commandées par quatre moteurs indépendants. La paire de roues du même côté roule avec une même vitesse et sont commandées indépendamment de celles du côté opposé. Si les deux paires de roues roulent avec une même vitesse, donc le système mobile se déplace dans le même sens de déplacement des roues, mais si elles roulent avec des vitesses opposées, le système mobile va tourner sur lui même. Et si ces deux paires roulent avec des vitesses différentes (un côté roule plus vite que l'autre), le système mobile suivra une courbe du côté roulant avec la vitesse la plus faible.

L'ATRV2 est contrôlé par le système *rFLEX* de la société *iRobot* fournissant des contrôles standardisés et une interface utilisateur facile à utiliser et conviviale permettant l'interaction avec l'utilisateur à travers un ensemble de menus permettant la gestion des capteurs ultrasonores, les moteurs et autres systèmes embarqués (tels que les capteurs infrarouges et capteurs tactiles). Ce système est

considéré comme le système nerveux central de l'ATRV2 où les câbles, nœuds et Hubs sont faciles à configurer et son réseau est optimisé pour lier les capteurs embarqués et les actionneurs à l'ordinateur de traitement central (cerveau).

Tous les capteurs et actionneurs sont connectés sur le bus *FARnet* d'*iRobot* et contrôlés par un ordinateur embarqué. Ce dernier est équipé d'un processeur Pentium monté sur une carte mère dual processors à 800MHz utilisant *Linux RedHat 6.2* comme système d'exploitation, et est utilisé pour sauvegarder et exécuter tous les programmes et données du système mobile.

L'ordinateur embarqué communique avec un autre ordinateur extérieur par l'intermédiaire de (Figure IV.3):

- Deux antennes *BreezeCom* pour une communication Ethernet sans fils.
- Un port Ethernet (*RJ45*) pour une connexion directe à un réseau local.
- Un port série pour une communication série avec un autre ordinateur.

Le contrôle software se fait par *Mobility™*, logiciel distribué, toolkit orienté objet pour la construction de programmes de contrôle d'un ou plusieurs systèmes mobiles de la famille *iRobot*. Il est basé sur *CORBA*, collecte les données en provenance des capteurs embarqués, les interprète et les exécute. En utilisant *CORBA*, tous les capteurs et actionneurs peuvent être vus comme des ressources Internet. Le logiciel *Mobility™* consiste en :

- Un ensemble d'outils logiciels,
- Un modèle objet pour le logiciel du système mobile,
- Un ensemble de modules de contrôle de base du système mobile,
- Une structure de classes orientées objet pour simplifier le développement du code.

Les capteurs embarqués peuvent être connectés et contrôlés par un simple programme écrit en langage *C* incluant les fonctions et procédures fournies par *Mobility™*.

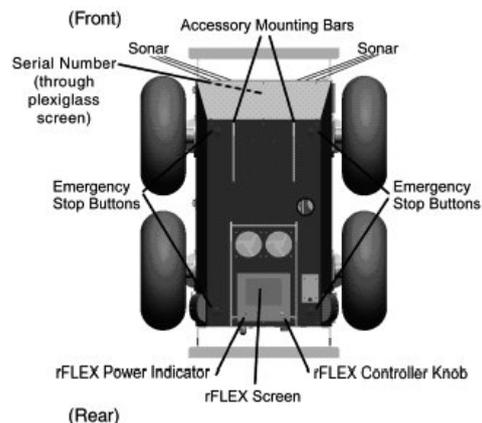


Figure IV.1 : Le système mobile ATRV2.

Figure IV.2 : Vue de haut du système mobile ATRV2 [Iro01].

Chaque roue du système mobile *ATRV2* est munie d'un odomètre qui sert à mesurer les déplacements et les variations angulaires. Les données provenant de ce capteur sont présentées de deux façons :

La première donnant séparément l'information de la translation (la distance totale effectuée en mètres) et de la rotation (l'angle total de rotation en degrés). La seconde donnant la position du système mobile en X et en Y. Dans ce mode, les

mouvements sont automatiquement dressés sur un plan XY où la position du système mobile peut être lue directement.

Malheureusement, ce type de capteur n'est pas toujours précis et donne des informations parfois erronées dans les cas suivants : La rugosité de la surface du sol, les ondulations des roues, et la perte d'adhérence des roues, etc.

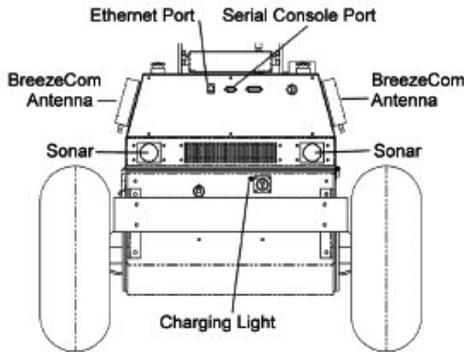


Figure IV.3 : Vue arrière de l'ATRV2 [Iro01].

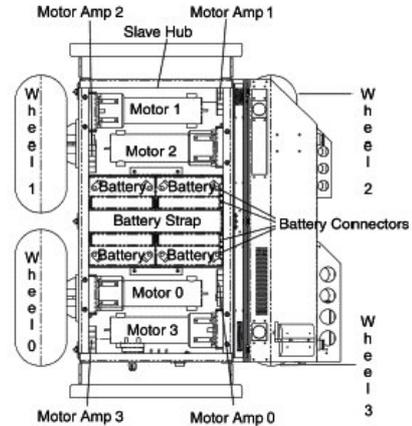


Figure IV.4 : Vue de l'intérieur de l'ATRV2 [Iro01].

IV.1.1. Dimensions.

La géométrie du système mobile ATRV2 est illustrée dans la figure IV.5. Il est représenté par un cube rectangle de taille 105mmx80mmx65mm. Cette représentation nous est utile lors de la définition de l'espace nécessaire pour son déplacement.

L'ATRV2 pèse 120 Kg batteries comprises. Il peut transporter une charge utile de 100 Kg à une vitesse maximale de 1,5 m/s.

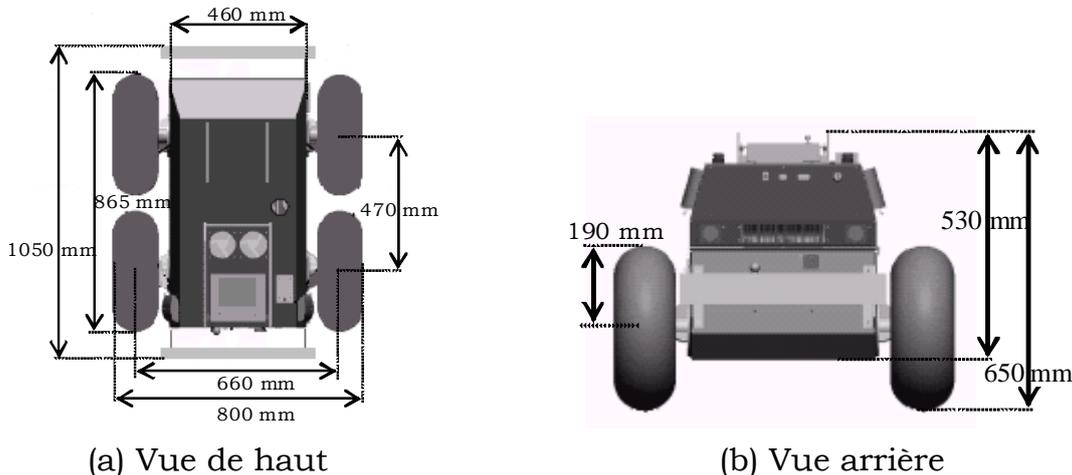


Figure IV.5: Géométrie de l'ATRV2.

IV.1.2. Cinématique.

Un repère local (O_R, X_R, Y_R) est associé au système mobile (Figure IV.6). L'origine de celui-ci est placée sur le point d'intersection des deux diagonales du rectangle formé par les quatre roues du système mobile. L'axe X_R de ce repère est perpendiculaire à l'axe de traction. Le système mobile se déplace dans l'espace plan de repère $R_u(O^u, X^u, Y^u)$. Sa configuration complète dans ce repère peut être

représentée par les trois valeurs (x, y, φ) . En fonction de la variation de ses coordonnées dans le repère R_u , la vitesse linéaire peut être déduite :

$$\begin{cases} \dot{x} = v \cos \varphi \\ \dot{y} = v \sin \varphi \\ v = \sqrt{\dot{x}^2 + \dot{y}^2} \end{cases} \quad (IV.1)$$

Le changement d'orientation du système mobile, c'est à dire, sa vitesse de rotation Ω , est caractérisée par :

$$\Omega = \dot{\varphi} = \frac{d\varphi}{dt} \quad (IV.2)$$

où φ est l'orientation instantanée du système mobile par rapport à R_u .

Le mouvement est créé par la rotation des quatre roues motrices du système mobile. Les deux paires de roues correspondant aux côtés gauche et droit du système possèdent une vitesse de rotation instantanée $\omega_g(t)$ et $\omega_d(t)$.

Sans modifier la cinématique du système mobile ATRV2, ce dernier peut être vu comme un système mobile à deux roues motrices virtuelles roulant avec les mêmes vitesses de rotation instantanées gauche et droite $\omega_g(t)$ et $\omega_d(t)$, et dont l'axe est placé au milieu des deux axes portant ses quatre roues motrices (Figure IV.7).

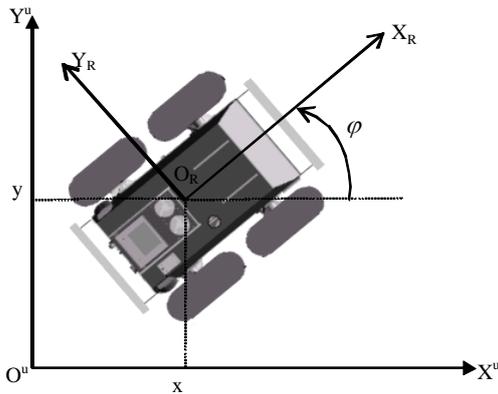


Figure IV.6 : Le repère de l'ATRV2 dans le repère absolu R_u .

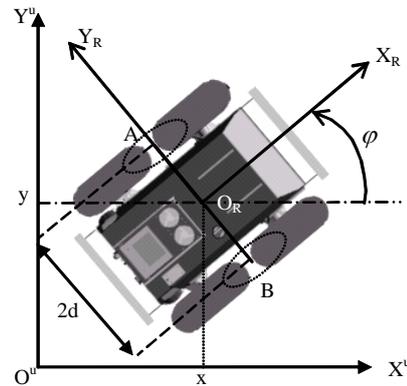


Figure IV.7 : Le modèle cinématique utilisé pour l'ATRV2.

Afin de trouver les équations de déplacement de l'ATRV2, nous pouvons reprendre l'étude présentée dans [Pre91] décrivant le mouvement de deux points d'un même corps (solide). A chaque instant t , il existe un pseudo vecteur $\Omega(t)$ tel que pour tout point P du solide on a :

$$\vec{v}_p = \vec{v}_a + \Omega \wedge \overrightarrow{AP} \quad (IV.3)$$

où A est un point quelconque du solide et Ω , le vecteur vitesse de rotation du corps indépendante du point.

Puisque les deux roues motrices virtuelles sont liées mécaniquement par un axe virtuel, et que leur vitesse est connue (v_g et v_d), alors on peut calculer la vitesse v du centre O_R de l'axe de traction virtuel (Figure IV.8):

$$\vec{v} = \vec{v}_g + \Omega \wedge \overrightarrow{O_R A} \quad (\text{IV.4})$$

$$\vec{v} = \vec{v}_d + \Omega \wedge \overrightarrow{O_R B} \quad (\text{IV.5})$$

De la somme des deux équations (IV.4) et (IV.5), on obtient :

$$2\vec{v} = \vec{v}_g + \vec{v}_d + \Omega \wedge (\overrightarrow{O_R A} + \overrightarrow{O_R B}) \quad (\text{IV.6})$$

Et de la figure IV.8, nous voyons bien que les vecteurs $\overrightarrow{O_R A}$ et $\overrightarrow{O_R B}$ sont égaux et opposés. Donc, cela conduit à :

$$\vec{v} = \frac{\vec{v}_g + \vec{v}_d}{2} \quad (\text{IV.7})$$

Nous pouvons remarquer que ce vecteur vitesse a la même direction que \vec{v}_g et \vec{v}_d . Cette direction correspond à la direction de l'axe X_R du repère du système mobile.

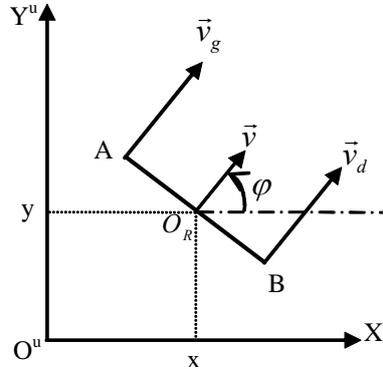


Figure IV.8 : Espace de vitesses de l'ATRV2.

Nous pouvons maintenant faire la soustraction des équations (IV.4) et (IV.5). A cet effet, il faudra remarquer que le vecteur Ω est le résultat d'une paire de torsion orthogonale au plan du mouvement.

$$\vec{v}_d - \vec{v}_g + \Omega \wedge (\overrightarrow{O_R B} - \overrightarrow{O_R A}) = 0 \quad (\text{IV.8})$$

L'expression du vecteur vitesse de rotation du système mobile en fonction des vecteurs vitesses linéaires des roues est donc :

$$\Omega = \dot{\varphi} = \frac{\vec{v}_d - \vec{v}_g}{2d} \quad (\text{IV.9})$$

avec $2d$ la distance du segment AB .

Pour les cas de changement de position du point O_R (Figure IV.8) dans les directions des axes X et Y du repère R_u , ils sont donnés, en module, par :

$$\dot{x} = \frac{v_g + v_d}{2} \cos \varphi \quad (\text{IV.10})$$

$$\dot{y} = \frac{v_g + v_d}{2} \sin \varphi \quad (\text{IV.11})$$

Les équations IV.1, IV.9, IV.10 et IV.11 constituent l'ensemble des équations cinématiques du système mobile ATRV2.

A partir de l'équation (IV.9), nous pouvons expliquer le comportement du système mobile pour les cas présentés par la figure IV.9. Dans le cas des vitesses $v_d = v_g$, illustré par la figure IV.9a, le déplacement est rectiligne et égal à v_d (ou v_g). La vitesse de rotation du système mobile Ω est nulle. Si $v_d < v_g$, figure IV.9b, le déplacement décrit un arc de cercle à droite, c'est à dire, avec une vitesse angulaire $\Omega < 0$. Dans le cas contraire, présenté par la figure IV.9c, $v_d > v_g$, nous obtenons un déplacement décrivant un arc de cercle à gauche correspondant à une vitesse angulaire $\Omega > 0$. Pour le cas spécial, où $v_d = -v_g$, le déplacement sera nul, c'est à dire, $v=0$, et nous aurons une rotation pure sur place avec une valeur $\Omega \neq 0$. Ce dernier cas est présenté par la figure IV.9d pour $\Omega < 0$.

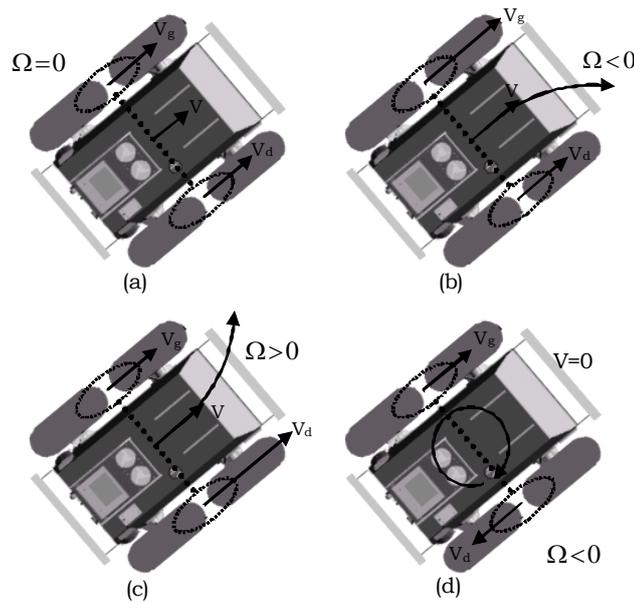


Figure IV.9 : Quatre différents types de mouvement réalisables par l'ATRV2.

Lorsque nous avons un déplacement décrivant un arc de cercle, tout le long de ce dernier, l'axe virtuel est orienté de manière à définir des rayons qui vont converger sur un point de l'espace. Ce point est appelé *Centre Instantané de Rotation (CIR)* (Figure IV.10). Dans le cas d'un déplacement rectiligne, le CIR va être placé à l'infini (Figure IV.9a). D'autre part, pour une rotation pure, comme celle de la figure IV.9d, le CIR va se trouver sur le point milieu $C (O_R)$ de l'axe de traction virtuel.

En se déplaçant sur un arc de cercle de centre CIR et de rayon R (Figure IV.10), les vitesses des roues virtuelles sont différentes de telle façon à avoir des déplacements différents sur le même espace de temps. Le déplacement δs du système mobile lorsqu'il décrit un angle $\delta \varphi$ de rayon R , est donné par :

$$\delta s = R \delta \varphi \quad (IV.12)$$

Il faut remarquer que $\delta \varphi$ et R peuvent être considérés positifs ou négatifs en fonction du sens du déplacement (positif dans le sens trigonométrique).

Chaque roue tourne autour du CIR avec la même vitesse angulaire $\delta \varphi$. Donc nous pouvons écrire :

$$v_g = (R - d) \delta\varphi \tag{IV.13}$$

$$v_d = (R + d) \delta\varphi \tag{IV.14}$$

De ces deux équations on peut extraire l'expression de l'angle de rotation du système mobile $\delta\varphi$ et l'expression du rayon R :

$$\delta\varphi = \frac{v_d - v_g}{2d} \tag{IV.15}$$

$$R = d \frac{v_d + v_g}{v_d - v_g} = \frac{v}{\delta\varphi} \tag{IV.16}$$

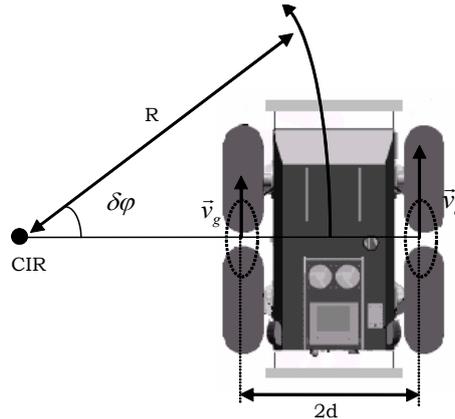


Figure IV.10 : Déplacement de l'ATRV2 dans le cas d'un virage à gauche.

IV.1.3. Holonomie et non Holonomie.

La capacité d'un système mobile de pouvoir se déplacer à partir d'une situation dans toute direction est appelée *holonomie*. Il n'y a pas de contrainte sur les commandes. Cela signifie donc une très grande simplification des problèmes de contrôle [Arm93]. Ce type de système possède le même nombre de déplacements virtuels indépendants que le nombre de coordonnées généralisées pour le décrire.

Par contre, de nombreux systèmes mobiles sont des systèmes *non holonomes*. Pour ce type de systèmes, bien que l'espace de configuration sur le plan soit de dimension trois, le mouvement est produit par deux déplacements indépendants seulement, est donc le système mobile ne possède que deux degrés de liberté [Tou88].

Pour le cas du système mobile ATRV2, le mouvement est créé par deux vitesses : une vitesse v de translation et une vitesse Ω de rotation, qui sont en fonction des vitesses de ses roues (équations IV.7 et IV.9).

Les contraintes cinématiques qui contraignent les mouvements d'un système mobile sont dites *non-holonomes*. Leur caractéristique est de réduire la dimension de l'espace de contrôle dans l'espace tangent à l'espace des configurations en chaque configuration, sans pour autant réduire la dimension de l'espace des configurations [Arm93]. Donc, le système mobile l'ATRV2 est un système *non-holonyme* dans le sens où l'équation de contrainte suivante est toujours satisfaite :

$$\cos \varphi \, dy - \sin \varphi \, dx = 0 \tag{IV.17}$$

IV.2. Approche de localisation utilisée.

L'approche de localisation utilisée est la même que celle développée dans [Bou07a, Bou07b]. Basée sur le filtre de *Kalman* étendu (*EKF*), cette dernière fusionne les données odométriques (modèle du système odométrique) et les données du système de perception embarqué (mesure) pour corriger régulièrement l'estimation relative de la position du système mobile due à l'erreur cumulative engendrée par les capteurs odométriques. Pour déterminer la mesure, l'approche utilise la mise en correspondance de cartes tout en estimant sa zone de recherche et la position autour de laquelle elle effectue la recherche de la position estimée.

IV.2.1 Modèle du système odométrique.

L'odométrie est une technique de localisation qui permet de déterminer la position (x, y, φ) du système mobile par rapport au repère univers en intégrant des translations et rotations élémentaires.

Dans notre cas, le système mobile *ATRV2* fournit directement sa position (x_r, y_r, θ_r) pour chaque période d'échantillonnage par rapport au repère du système mobile initial R_i (Figure IV.11). Cette donnée est utilisée pour exprimer la rotation élémentaire $\delta\varphi$ et la translation élémentaire δd du système mobile.

Pour un déplacement entre deux instants $k-1$ et k , leur correspondent respectivement les mesures odométriques suivantes $(x_{r_{k-1}}, y_{r_{k-1}}, \theta_{r_{k-1}})$ et $(x_{r_k}, y_{r_k}, \theta_{r_k})$.

Nous en déduisons donc :

⇒ Le déplacement élémentaire du système mobile

$$\delta d = \sqrt{(x_{r_k} - x_{r_{k-1}})^2 + (y_{r_k} - y_{r_{k-1}})^2} \quad (\text{IV.18})$$

⇒ et la variation angulaire du système mobile

$$\delta\varphi = \theta_{r_k} - \theta_{r_{k-1}} \quad (\text{IV.19})$$

Nous cherchons à exprimer de manière récurrente, par rapport au repère univers R_u , la position du système mobile (x_k, y_k, φ_k) à l'instant k en fonction de la position $(x_{k-1}, y_{k-1}, \varphi_{k-1})$ à l'instant $k-1$ et les déplacements élémentaires mesurés.

A partir des équations (IV.1 et IV.2), le modèle cinématique est décrit selon le système d'équations suivant:

$$\begin{cases} \dot{x} = v \cos\theta \\ \dot{y} = v \sin\theta \\ \dot{\theta} = \dot{\varphi} \end{cases} \quad (\text{IV.20})$$

Il s'agit d'un modèle non linéaire, pour lequel il n'existe pas de méthode générale de discrétisation [Bon97]. Dans le cas particulier qui nous intéresse, on peut approcher \dot{x} par une différence finie, si la période d'échantillonnage T_e est petite (même chose pour \dot{y} et $\dot{\varphi}$) alors:

$$\dot{x}_{(t=k)} = \frac{x_k - x_{k-1}}{T_e} \quad (\text{IV.21})$$

Le système d'équations (IV.20) du modèle cinématique peut être discrétisé ainsi:

$$\begin{cases} x_k = x_{k-1} + T_e v_{k-1} \cos\varphi_{k-1} \\ y_k = y_{k-1} + T_e v_{k-1} \sin\varphi_{k-1} \\ \varphi_k = \varphi_{k-1} + T_e \omega_{k-1} \end{cases} \quad (\text{IV.22})$$

Nous obtenons donc en faisant apparaître les mesures des déplacements élémentaires δd et $\delta\varphi$, le modèle cinématique devient:

$$\begin{cases} x_k = x_{k-1} + \delta d \cos\varphi_{k-1} \\ y_k = y_{k-1} + \delta d \sin\varphi_{k-1} \\ \varphi_k = \varphi_{k-1} + \delta\varphi \end{cases} \quad (\text{IV.23})$$

Ce système d'équations décrit le déplacement d'un système mobile en ligne droite suivant la direction φ_{k-1} sur la longueur δd , puis effectue une rotation sur place de $\delta\varphi$ [Bon97].

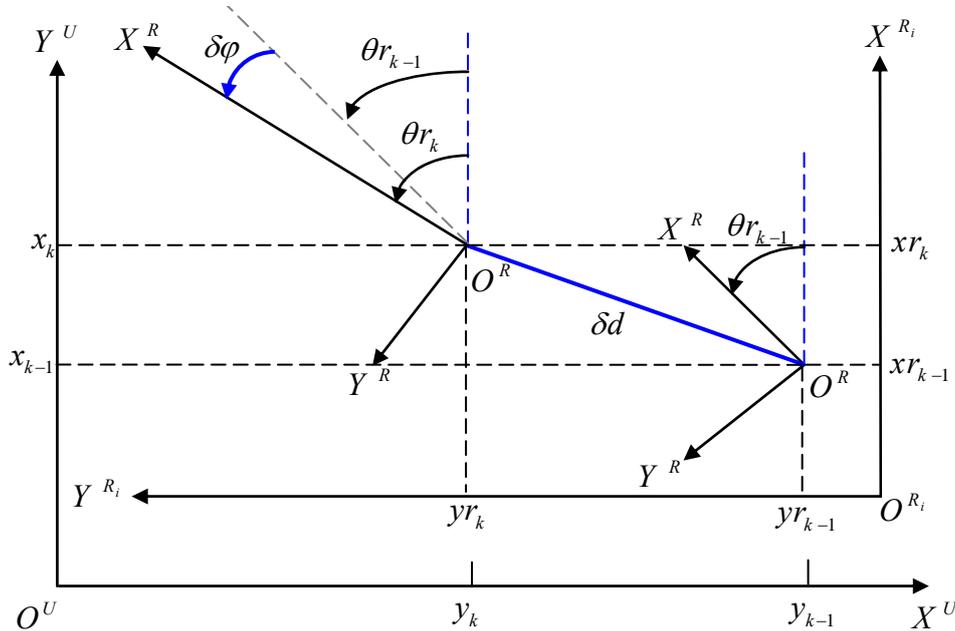


Figure IV.11 : Position de l'ATRV2 dans le repère univers R_u en fonction des données odométriques.

Dans le cas où le système mobile décrit une trajectoire non rectiligne, et en supposant que le mouvement est circulaire, le déplacement élémentaire δd s'écrit donc comme suit :

$$\delta d = \rho \delta\varphi \quad (\text{IV.24})$$

où ρ représente le rayon de courbure (Figure IV.12).

De la Figure IV.12, nous pouvons écrire :

$$\|M_{k-1}H\| = \|HM_k\| \approx \rho \sin(\delta\varphi/2) \quad (\text{IV.25})$$

$$\begin{aligned} \|M_{k-1}M_k\| &\approx 2\rho \sin(\delta\varphi/2) = 2 \frac{\delta d}{\delta\varphi} \sin(\delta\varphi/2) \\ &= \delta d \frac{\sin(\delta\varphi/2)}{\delta\varphi/2} \end{aligned} \quad (\text{IV.26})$$

et

$$\varphi_k = \varphi_{k-1} + \delta\varphi/2 \quad (\text{IV.27})$$

on obtient alors :

$$\begin{cases} x_k = x_{k-1} + \Delta x \\ y_k = y_{k-1} + \Delta y \\ \varphi_k = \varphi_{k-1} + \delta\varphi \end{cases} = \begin{cases} x_k = x_{k-1} + \delta d \frac{\sin(\delta\varphi/2)}{\delta\varphi/2} \cos(\varphi_{k-1} + \delta\varphi/2) \\ y_k = y_{k-1} + \delta d \frac{\sin(\delta\varphi/2)}{\delta\varphi/2} \sin(\varphi_{k-1} + \delta\varphi/2) \\ \varphi_k = \varphi_{k-1} + \delta\varphi \end{cases} \quad (IV.28)$$

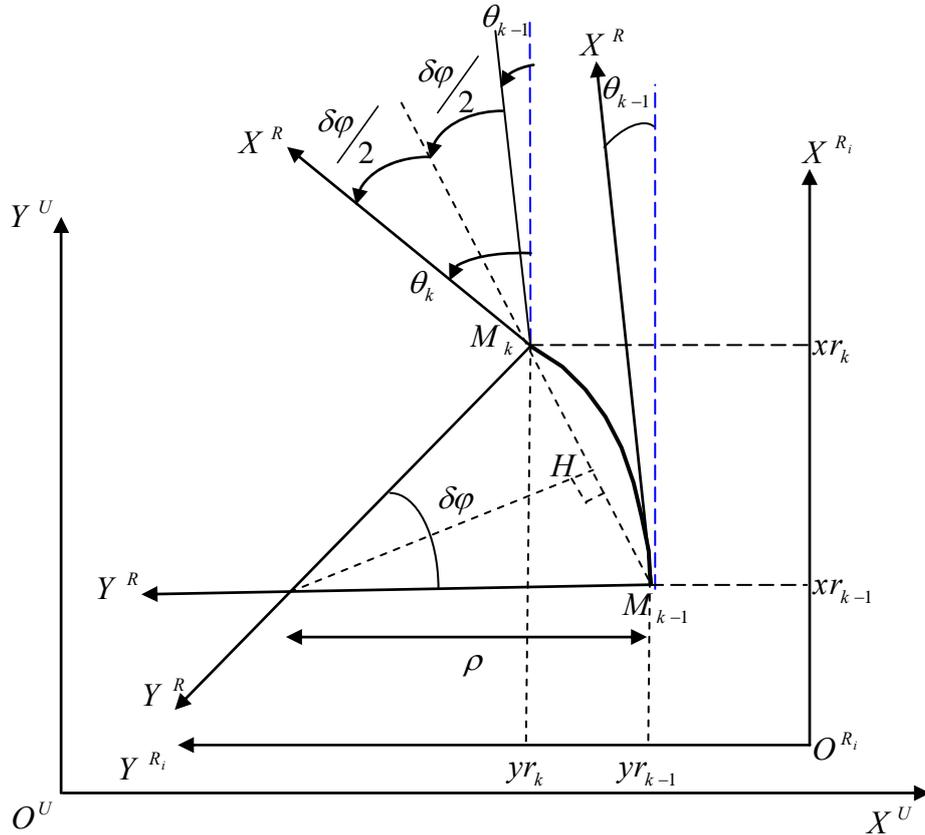


Figure IV.12 : Déplacement circulaire entre deux instants d'échantillonnage.

Si de plus $\delta\varphi$ est petit, ce qui suppose que la période d'échantillonnage est suffisamment petite par rapport à la dynamique du système mobile, alors le noyau de Fourier est égal à un à l'ordre deux près [Bou07a]:

$$\begin{cases} x_k = x_{k-1} + \delta d \cos(\varphi_{k-1} + \delta\varphi/2) \\ y_k = y_{k-1} + \delta d \sin(\varphi_{k-1} + \delta\varphi/2) \\ \varphi_k = \varphi_{k-1} + \delta\varphi \end{cases} \quad (IV.29)$$

Ce dernier modèle odométrique est l'un des plus utilisés [Bon97]. Par rapport au modèle décrit par le système (IV.23), il apparaît une correction $\delta\varphi/2$ dans les termes trigonométriques. C'est donc sur des trajectoires où le cap du système mobile varie, que ce modèle est plus précis. Pour un système mobile se déplaçant principalement en ligne droite, ou bien à variation de cap lente, le terme $\delta\varphi/2$ n'est pas nécessaire [Bon97]. Néanmoins, comme il complique peu les expressions mathématiques, nous choisissons de le conserver pour la suite de notre travail.

En tenant compte principalement des contraintes odométriques, on introduit dans nos équations le bruit du modèle d'évolution (normally distributed noise) $W = (0, Q)$ (de moyenne nulle et de variance Q).

Donc le modèle que nous utilisons est le suivant :

$$\begin{cases} x_k = x_{k-1} + \delta d \cos(\varphi_{k-1} + \delta\varphi/2) + W_x \\ y_k = y_{k-1} + \delta d \sin(\varphi_{k-1} + \delta\varphi/2) + W_y \\ \varphi_k = \varphi_{k-1} + \delta\varphi + W_\varphi \end{cases} \quad (\text{IV.30})$$

Enfin pour faire apparaître les mesures odométriques, on remplace δd et $\delta\varphi$ par leurs expressions, le modèle devient alors :

$$\begin{cases} x_k = x_{k-1} + \sqrt{(xr_k - xr_{k-1})^2 + (yr_k - yr_{k-1})^2} \cos(\varphi_{k-1} + (\theta r_k - \theta r_{k-1})/2) + W_x \\ y_k = y_{k-1} + \sqrt{(xr_k - xr_{k-1})^2 + (yr_k - yr_{k-1})^2} \sin(\varphi_{k-1} + (\theta r_k - \theta r_{k-1})/2) + W_y \\ \varphi_k = \varphi_{k-1} + (\theta r_k - \theta r_{k-1}) + W_\varphi \end{cases} \quad (\text{IV.31})$$

Par analogie ce système peut se mettre sous forme matricielle de la manière suivante :

$$\begin{pmatrix} x_k \\ y_k \\ \varphi_k \end{pmatrix} = \begin{pmatrix} \sqrt{(xr_k - xr_{k-1})^2 + (yr_k - yr_{k-1})^2} \cos(\varphi_{k-1} + (\theta r_k - \theta r_{k-1})/2) & 0 & 0 \\ 0 & \sqrt{(xr_k - xr_{k-1})^2 + (yr_k - yr_{k-1})^2} \sin(\varphi_{k-1} + (\theta r_k - \theta r_{k-1})/2) & 0 \\ 0 & 0 & (\theta r_k - \theta r_{k-1}) \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \varphi_{k-1} \end{pmatrix} + \begin{pmatrix} W_x \\ W_y \\ W_\varphi \end{pmatrix} \quad (\text{IV.32})$$

On note $X_k = [x_k, y_k, \varphi_k]^t$ la posture du système mobile (vecteur d'état), et $U = (xr, yr, \theta r)$ le vecteur obtenu à partir des odomètres pour une période d'échantillonnage entre l'état $k-1$ et k .

Finalement le système précédent peut se mettre sous la forme vectorielle :

$$X_k = f(X_{k-1}, U, W) \quad (\text{IV.33})$$

avec f fonction vectorielle continue et dérivable.

Or, le modèle interne (système d'équation IV.32) du système mobile ATRV2 est non linéaire, dans ce cas, il faut calculer le Jacobien de ce système, tel que :

$$A = \nabla f_{X_k} = \frac{\delta X_k}{\delta X_{k-1}} \quad (\text{IV.34})$$

c'est-à-dire,

$$A = \begin{bmatrix} \nabla f_{11} & \nabla f_{12} & \nabla f_{13} \\ \nabla f_{21} & \nabla f_{22} & \nabla f_{23} \\ \nabla f_{31} & \nabla f_{32} & \nabla f_{33} \end{bmatrix} = \begin{bmatrix} \frac{\delta x_k}{\delta x_{k-1}} & \frac{\delta x_k}{\delta y_{k-1}} & \frac{\delta x_k}{\delta \varphi_{k-1}} \\ \frac{\delta y_k}{\delta x_{k-1}} & \frac{\delta y_k}{\delta y_{k-1}} & \frac{\delta y_k}{\delta \varphi_{k-1}} \\ \frac{\delta \varphi_k}{\delta x_{k-1}} & \frac{\delta \varphi_k}{\delta y_{k-1}} & \frac{\delta \varphi_k}{\delta \varphi_{k-1}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -\sqrt{(x_{r_k} - x_{r_{k-1}})^2 + (y_{r_k} - y_{r_{k-1}})^2} \sin(\varphi_{k-1} + (\theta_{r_k} - \theta_{r_{k-1}})/2) \\ 0 & 1 & \sqrt{(x_{r_k} - x_{r_{k-1}})^2 + (y_{r_k} - y_{r_{k-1}})^2} \cos(\varphi_{k-1} + (\theta_{r_k} - \theta_{r_{k-1}})/2) \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{IV.35})$$

IV.2.2. Le modèle de mesure.

La mesure est une position (x, y, φ) du système mobile ATRV2 dans l'environnement (repère R_u). Elle est exprimée en fonction du vecteur d'état X_k . Ce dernier est une position $m_k = (x_k, y_k, \varphi_k)$ avec un bruit V additif. Ce lien se traduit par l'équation suivante :

$$m_k = H(X_k, V) \quad (\text{IV.36})$$

Compte tenu du fait que le bruit a une moyenne nulle, l'équation (IV.36) devient:

$$m_k = H(X_k, 0) \quad (\text{IV.37})$$

Cette même relation pourra prendre la forme matricielle :

$$\begin{bmatrix} x_k \\ y_k \\ \varphi_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \varphi_k \end{bmatrix} \quad (\text{IV.38})$$

Donc la matrice H a pour valeur :

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{IV.39})$$

Il apparaît à partir des équations de mesure (IV.37 et IV.38), que la fonction de mesure H est linéaire, et dans ce cas la mesure prendra la forme :

$$m_k = HX_k \quad (\text{IV.40})$$

Cette mesure m_k représente la position du système mobile calculée par l'algorithme de mise en correspondances de cartes.

IV.2.3. Le modèle d'incertitude.

Le modèle d'incertitude représente les bruits qui excitent le système. En général, nous n'avons pas de connaissance a priori des corrélations des différentes sources d'erreurs. Chacune des matrices de covariance du bruit du processus Q et de l'erreur de mesure R sont supposées diagonales [Bou07a]. Q et R sont exprimées comme suit:

$$Q = \text{diag}[\sigma_{Qx}^2 \quad \sigma_{Qy}^2 \quad \sigma_{Q\varphi}^2] \quad (\text{IV.41})$$

$$R = \text{diag}[\sigma_{Rx}^2 \quad \sigma_{Ry}^2 \quad \sigma_{R\varphi}^2] \quad (\text{IV.42})$$

Les covariances sont déterminées expérimentalement. Pour la matrice de covariance Q , nous avons fait une comparaison entre les résultats lus par les odomètres et les positions réelles du système mobile, nous avons pris la moyenne

d'erreur comme covariance sur la position, donc $\sigma_{Qx} = \sigma_{Qy} = 200mm$, équivalents à deux cellules de la carte globale (§III.2.1.2) et $\sigma_{Q\varphi} = 10^\circ$.

Tandis que pour la matrice de covariance R , nous avons comparé les positions fournies par la mesure et les positions réelles. Nous avons pris la moyenne d'erreur, c'est-à-dire, $\sigma_{Rx} = \sigma_{Ry} = 100mm$ (une cellule de la carte globale) et $\sigma_{R\varphi} = 5^\circ$. Ces valeurs correspondent aux pas avec lesquels nous avons translater la carte locale par rapport à la carte globale dans l'algorithme de mise en correspondance.

IV.2.4. Prédiction de la position du système mobile.

Elle consiste à calculer la position et l'orientation estimée du système mobile ATRV2 à l'état k , connaissant sa position et son orientation à l'état $k-1$, ainsi que les données odométriques lues. La position estimée à l'état k peut être calculée comme suit:

$$X_k^- = f(X_{k-1}, U, 0) \quad (IV.43)$$

équivalent à :

$$\begin{pmatrix} x_k^- \\ y_k^- \\ \varphi_k^- \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sqrt{(xr_k - xr_{k-1})^2 + (yr_k - yr_{k-1})^2} \sin(\varphi_{k-1} + (\theta r_k - \theta r_{k-1})/2) \\ 0 & 1 & \sqrt{(xr_k - xr_{k-1})^2 + (yr_k - yr_{k-1})^2} \cos(\varphi_{k-1} + (\theta r_k - \theta r_{k-1})/2) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \varphi_{k-1} \end{pmatrix} \quad (IV.44)$$

Sa covariance sera aussi utile, et sa valeur est définie par l'équation :

$$P_k^- = AP_{k-1}A^T + Q \quad (IV.45)$$

A l'état initial, la position initiale X_0 est connue, tandis que sa covariance P_0 est estimée initialement de quelques centimètres pour la position ($\sigma_x = \sigma_y = 200mm$) et quelques degrés pour l'orientation ($\sigma_\varphi = 10^\circ$). En pratique, la valeur de P_0 est choisie assez grande pour englober toutes les erreurs possibles, du moment que la valeur de la covariance P diminue avec la convergence du filtre.

IV.2.5. La mesure.

La mesure donne l'information sur la position du système mobile. Elle est déterminée à partir des données lues des capteurs ultrasonores embarqués et sauvegardées dans une carte locale, pour être mise en correspondance avec la carte globale représentant l'environnement.

La mise en correspondance a pour but de déterminer la position et l'orientation du système mobile dans son environnement. Le système mobile se trouvant dans une position quelconque dans la carte globale, il reconstruit sa carte locale en utilisant ses capteurs embarqués. Partant d'une position et d'une orientation données, nous recherchons la translation et la rotation qui fera coïncider au mieux ces deux cartes.

▪ *Zone de recherche* : Elle est calculée à partir de la matrice de covariance de l'état estimé P_k^- , où σ^2 en x , y et φ représente l'intervalle d'erreur dans lequel se trouve cette position estimée. Nous sommes donc sûr que la position réelle du

système mobile se trouve dans un cercle de rayon 2σ autour de la position estimée, c'est pourquoi nous ferons une recherche pour la mise en correspondance dans une zone de $\pm 2\sigma_x$ et $\pm 2\sigma_y$ autour de la position estimée respectivement selon l'axe des abscisses X^u , et l'axe des ordonnées Y^u , et $\pm 2\sigma_\phi$ autour de l'orientation estimée.

▪ *La mise en correspondance* : Elle consiste à appairer la carte locale et la carte globale pour toutes les translations et rotations possibles dans la zone de recherche (dans le repère R^u). La position du système mobile est initialisée par la valeur X_k^- donnée par le filtre de *Kalman* étendu. Un gain initialisé à la valeur 0 est calculé pour toutes les translations et rotations possibles (équation IV.47). La valeur maximale de ce coefficient représente la mesure m_k qui servira par la suite pour le calcul de l'estimée finale de la position du système mobile par le filtre de *Kalman* étendu.

IV.2.6. Estimation finale de la position du système mobile.

L'estimée finale de la position du système mobile est calculée par l'équation suivante :

$$X_k = X_k^- + K_k (m_k - HX_k^-) \tag{IV.46}$$

avec K_k le gain de *Kalman* qui a pour expression :

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{IV.47}$$

Il est aussi important de calculer la covariance de l'état estimé final, qui va être utilisé dans le prochain cycle de localisation.

$$P_k = P_k^- - K_k HP_k^- \tag{IV.48}$$

L'état estimé X_k et sa covariance P_k doivent converger vers un résultat meilleur en évoluant d'un cycle à l'autre.

La figure ci-dessous montre le diagramme illustrant l'algorithme du filtre de *Kalman* étendu utilisé.

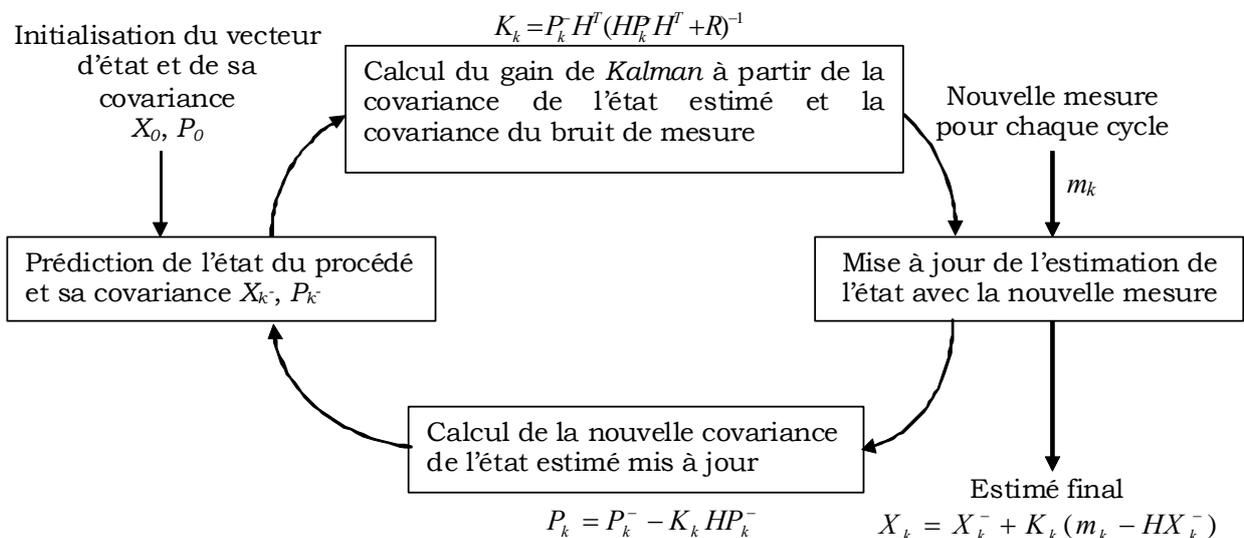


Figure IV.13 : L'algorithme du filtre de *Kalman* étendu utilisé.

IV.3. Approche de navigation proposée.

L'approche de navigation proposée appelée *DVFF* combine deux algorithmes existants, l'algorithme de planification de chemin optimal D^* pour le calcul de la direction globale du chemin permettant au système mobile ATRV2 de s'orienter vers sa destination, et l'algorithme de navigation non stop basé sur le principe des champs de forces virtuelles *VFF* (*Virtual Force Field en anglais*) pour l'évitement d'obstacles. La principale motivation de cette combinaison est de développer un algorithme hybride de navigation robuste comportant les avantages de ces deux algorithmes, et en même temps, éliminant plusieurs de leurs inconvénients.

IV.3.1. Algorithme de planification de chemin optimal D^* .

IV.3.1.1. Introduction.

L'algorithme D^* présente l'avantage par rapport à l'algorithme A^* d'être dynamique dans le sens où les paramètres de coût entre les cellules ou carrés peuvent changer durant le processus de résolution du problème de planification de chemin.

L'algorithme génère un chemin global en utilisant un modèle déjà établi de l'environnement appelé carte. Au cours du suivi du chemin global, le système mobile peut rencontrer un obstacle qui n'était pas défini au préalable, dans ce cas, il re-planifie sa trajectoire pour atteindre son but sans heurter l'obstacle.

L'algorithme D^* utilise le graphe d'état. Chaque état est une position du système mobile dans la carte, et les arcs joignant ces états représentent le coût de passage d'un état à un autre [Ste94].

IV.3.1.2. Algorithme développé.

L'algorithme de planification de chemin optimal basé sur D^* développé est inspiré essentiellement de celui présenté dans [Ste94] et modifié afin de bien s'adapter au type de représentation de l'environnement choisi. Chaque cellule de l'environnement lui est affecté un certain nombre d'attributs tels que : *back pointer* (b), *arc cost* (c), *tag* (t), *path cost* (h), et *key* (k), en plus des valeurs de certitude CV utilisées par l'algorithme (§III.2).

Nous avons ajouté un nouveau attribut wc (*walkable en anglais* ou *traversable en français*) pour donner l'information sur l'état (libre ou occupée) de chaque cellule de l'environnement. Dans [Ste94], les auteurs affectent un coût très important aux cellules occupées par des obstacles pour les différencier des cellules libres.

Cet attribut est utilisé pour assurer la rapidité de la convergence de l'algorithme, et en même temps réduire le temps de traitement de la mise à jour de la carte de l'environnement. Il prend la valeur *false* si la cellule est occupée par un obstacle ou elle est à proximité de cet obstacle selon les dimensions du système mobile, et *true* ailleurs (Figure IV.14).

Dans l'algorithme développé le C -space n'est pas utilisé, le système mobile n'est pas considéré comme un point matériel (prise en compte de la contrainte géométrique), et les tailles des obstacles ne sont pas élargies par une marge de

sécurité. Les cellules de la carte entourant ces obstacles, selon les dimensions du système mobile, ne sont pas prises en compte par cet algorithme.

Chaque cellule de la carte à l'exception de celles entourant les obstacles, possède une estimation du coût du chemin vers le but (ou la position destination), et un *back pointer* vers l'un de ses voisins indiquant la direction géométrique au but (nord, sud, est, ouest, nord-ouest, sud-ouest, nord-est et sud-est) appelée dans notre cas direction globale du chemin. La figure IV.15 montre comment calculer la direction globale du chemin conformément à la position du *back pointer*. Cette direction est très importante pour un système mobile se déplaçant dans l'environnement. De n'importe quelle cellule de la carte de l'environnement, le système mobile peut connaître la direction lui permettant d'atteindre sa position destination. La figure IV.16 montre le résultat du calcul des directions globales du chemin d'une carte simulée pour un point de départ et un point d'arrivée donnés. En suivant ces directions, le système mobile peut atteindre son point d'arrivée sans faire d'opération de suivi de chemin minimisant ainsi le temps de traitement perdu dans le contrôle utilisé par cette dernière.

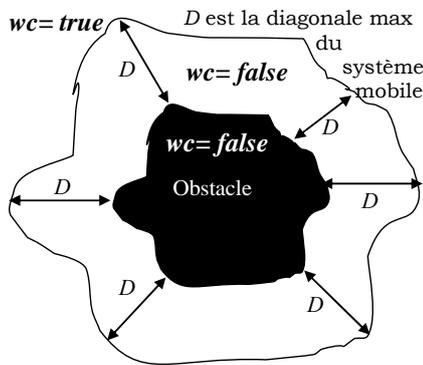


Figure IV.14: L'attribut *wc* selon les dimensions du système mobile.

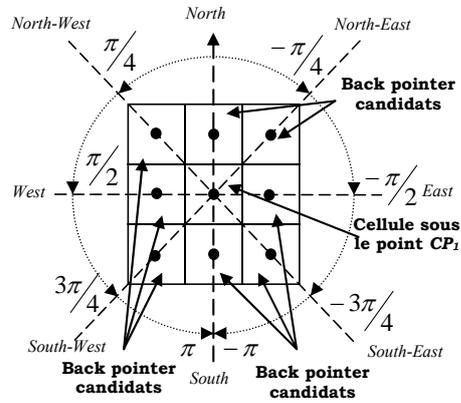


Figure IV.15: Calcul de la direction globale du chemin conformément à la position du *back pointer*.

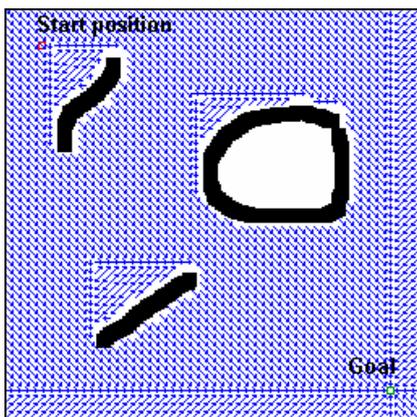


Figure IV.16: Direction du chemin global à partir d'un point de départ et un point d'arrivée donnés.

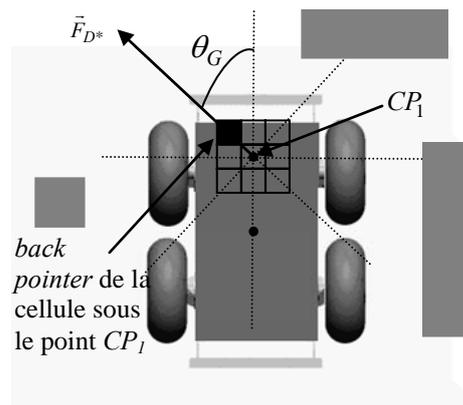


Figure IV.17 : Calcul de la direction du chemin global au point CP_1 .

Quand le système mobile se déplace, il utilise le *back pointer* de la cellule se trouvant sous son point CP_1 , détermine sa direction θ_G (direction globale du chemin) et applique un vecteur fictif dans cette direction (Figure IV.17). Cette direction est utilisée pour calculer la commande du mouvement conduisant sans

collision le système mobile vers son but. Le point du système mobile $CP1$ est un point situé sur son axe longitudinal, son emplacement optimal diffère pour différents systèmes mobiles de forme rectangulaire.

Le coût pour que le système mobile traverse de la cellule Y à la cellule X est un nombre positif donné par la fonction *arc cost* $c(X,Y)$ par l'équation suivante:

$$c(X,Y) = \begin{cases} 10 & \text{si } Y \text{ est horizontale ou verticale à } X \\ 14 & \text{si } Y \text{ est sur la diagonale de } X \\ & \text{avec } X \text{ et } Y \text{ voisins} \end{cases} \quad (\text{IV.49})$$

Pour chaque cellule N , l'algorithme maintient une estimation de la somme des *arc costs* de N à la cellule destination donnée par la fonction *path cost*, comme suit:

$$h(N) = |x_N - x_G| + |y_N - y_G| \quad (\text{IV.50})$$

avec (x_N, y_N) et (x_G, y_G) sont respectivement les coordonnées des cellules N et destination.

L'algorithme de planification de chemin optimal basé sur D^* implémenté se compose principalement de deux fonctions: $Init_D^*$ et $Case_D^*$ (voir annexe E).

La fonction $Init_D^*$ est utilisée seulement au début de l'algorithme pour calculer le chemin initial optimal à partir de la position de départ au but (G). Lorsque la valeur de la fonction *arc cost* de l'arc d'une cellule donnée a été modifiée, dans ce cas, il ya un nouveau obstacle détecté, la fonction $Case_D^*$ est utilisée par le système mobile pour calculer le nouveau chemin optimal à partir de la position du système mobile jusqu'à son but

Initialement, la valeur de l'attribut *tag* de toutes les cellules est mise à NEW , $h(G)$ est mis à zéro, et le point destination G est placé dans $OpenList$. La fonction $Init_D^*$ est appelée et exécutée jusqu'à ce que la cellule de la position de départ soit retirée de $OpenList$ ou alors la valeur -1 est renvoyée. A ce moment, soit une séquence de cellules formant le chemin optimal a été calculée, soit le chemin optimal n'existe pas.

Si le chemin existe, soit le système mobile continu à suivre les *backpointers* de la séquence calculée jusqu'à atteindre son but, soit il découvre une erreur dans la valeur de la fonction *arc cost* (détection d'un nouveau obstacle). Dans ce cas, la fonction $Case_D^*$ est immédiatement utilisée pour corriger et calculer une nouvelle séquence, pour que le système mobile puisse continuer à suivre les *backpointers* et atteindre son but.

IV.3.1.3. Résultats de simulation.

L'algorithme planification de chemin optimal basé sur D^* développé a été écrit en langage $C++$ *Builder* version 6 sous l'environnement Windows et implémenté sur un PC offboard. Le logiciel obtenu intègre tous les algorithmes et procédures développés, et permet le choix et la variation des paramètres liés à l'application (la position du point de départ et d'arrivée, l'emplacement des obstacles, etc.) (Figure IV.18).

Dans cet algorithme, la contrainte géométrique du système mobile ATRV2 n'est pas prise en compte dans le calcul du chemin optimal. Le système mobile est donc considéré comme un point matériel occupant une cellule de l'environnement.

L'algorithme a été testé dans plusieurs types d'environnements avec diverses formes d'obstacles (en utilisant le logiciel *Paint* de *Windows*). Les coordonnées des points de départ et d'arrivée, et la résolution de l'environnement (taille des cellules) sont choisies par l'utilisateur, ainsi que les modifications apportées sur l'environnement après le calcul du chemin optimal (l'utilisateur peut ajouter ou retirer des obstacles de l'environnement).

Pour illustration, après sélection d'un environnement et sa résolution ainsi que le choix des coordonnées des points de départ et d'arrivée, la fenêtre *D Star Form* (Figure IV.18) affiche le chemin optimal du point de départ vers le point d'arrivée. Nous remarquons que l'algorithme calcule parfaitement le chemin le plus optimal pour atteindre son point d'arrivée. Si nous voulons modifier l'environnement en ajoutant des cellules occupées par des obstacles, il suffit de cliquer sur le box *Check Obstacles* de la fenêtre *D Star Form*. Cette opération change l'attribut des cellules sélectionnées qui va passer de l'attribut libre ($w = 1$) à l'attribut occupée ($w = 0$).

Pour visualiser le nouveau chemin optimal calculé, il suffit de cliquer sur le bouton *Show global path* de la même fenêtre. Le calcul du nouveau chemin optimal va engendrer une modification dans les directions globales du chemin (Figures IV.19).

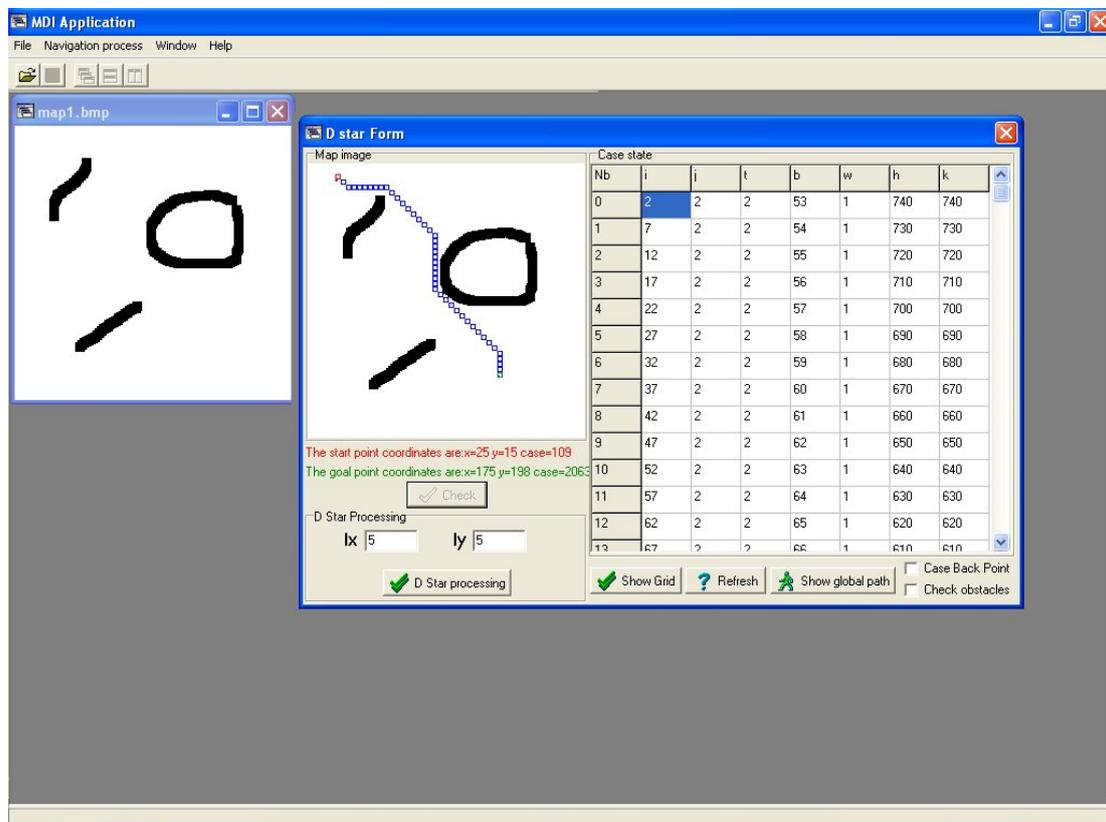


Figure IV.18 : L'interface du logiciel de recherche de chemin optimal développé basé sur l'algorithme D^* .

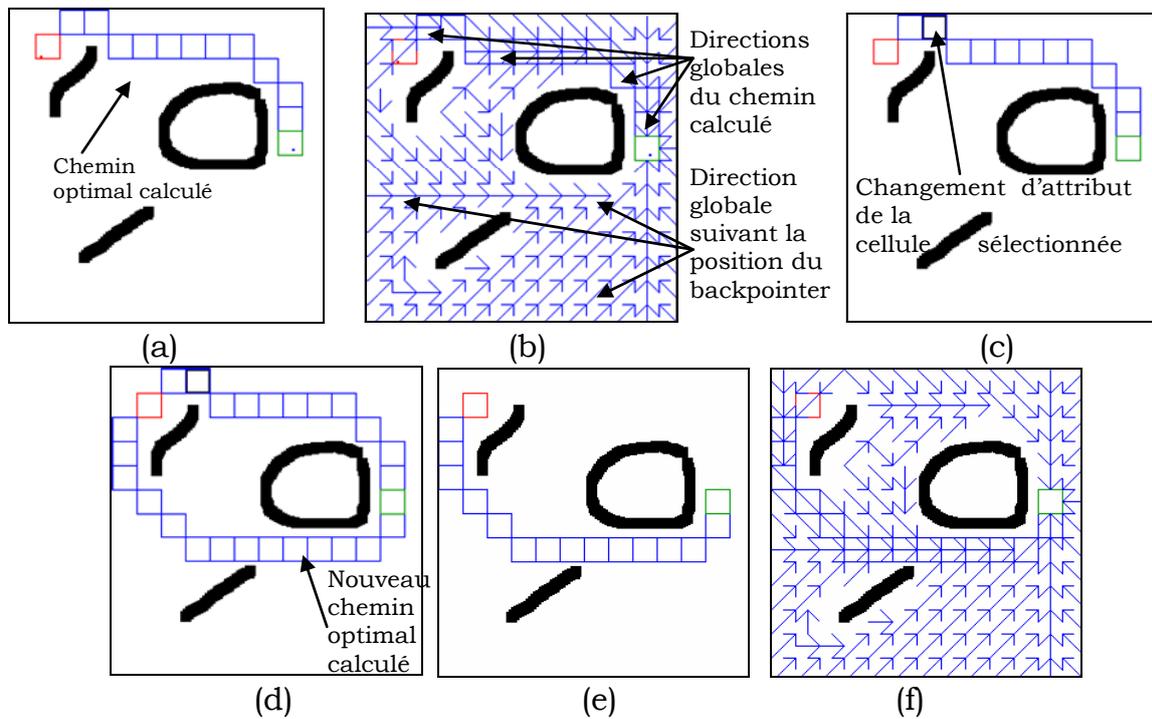


Figure IV.19: (a) Chemin optimal calculé entre les points de départ et d'arrivée, (b) Directions globales du chemin des cellules de la carte de l'environnement, (c) Changement d'attribut (Libre \rightarrow Occupée) d'une cellule appartenant au chemin optimal, (d,e) Nouveau chemin optimal calculé après le changement de l'attribut de la cellule sélectionnée, (f) Nouvelles directions globales du chemin calculées après le changement de l'attribut de la cellule sélectionnée.

IV.3.1.4. Résultats expérimentaux.

L'algorithme décrit ci-dessus a été implémenté sur le *PC* embarqué du système mobile *ATRV2* et testé à partir de plusieurs situations réelles en tenant compte de la contrainte géométrique du système dans le calcul du chemin optimal, c'est-à-dire, les cellules de l'environnement entourant les obstacles (selon les dimensions du système mobile) ne seront pas prises en compte par l'algorithme.

Dans ce qui suit, nous présentons les résultats de trois expériences réelles qui ont été effectuées au sein de la division Productique et Robotique du *CDTA*, et illustrées dans les figures IV.20-IV.22. Dans ces expériences, initialement l'environnement et la position des obstacles sont complètement inconnus par le système mobile *ATRV2*, seules les positions de départ et d'arrivée sont connues à l'avance. La vitesse de translation a été limitée à $v_c = 20 \text{ cm/s}$ et la vitesse de rotation a été limitée à $0,6 \text{ rd/s}$.

Dans chaque expérience et avant toute exécution de mouvement, le système mobile calcule son chemin optimal initial reliant sa position de départ et sa position d'arrivée (Figure IV.20.a- IV.22.a). A ce stade, chacune des cellules formant l'environnement possède une évaluation du coût du chemin menant le système mobile vers sa position d'arrivée, et un *back pointer* qui pointe vers l'un de ses voisins indiquant la direction vers cette position. Cette direction représente pour le système mobile la direction globale du chemin vers sa position d'arrivée.

Ensuite, le système mobile commence à se déplacer en suivant la direction globale, du chemin optimal, de la cellule se trouvant sous son point CP_1 (le système mobile suit ces directions mais pas le chemin optimal calculé). En même temps que son déplacement, le système mobile perçoit (à travers ses capteurs US) son environnement en découvrant d'éventuelles différences avec l'environnement reconstruit (détection d'un nouveau obstacle par exemple). Une fois ses différences trouvées, le système mobile replanifie son chemin pour rejoindre son point d'arrivée sans risque de collisions.

Les figures IV.20.b- IV.22.b et IV.20.c- IV.22.c montrent la grille de certitude de l'environnement reproduite durant chaque expérience. Les cellules occupées par les obstacles sont représentées par des carrés de couleur bleue. Les cellules traversées par le centre de gravité du système mobile sont représentées par des carrés de couleur rouge. Le rectangle de couleur verte représente le système mobile en format réel. Les figures IV.20.d- IV.22.d montrent les expérimentations réelles.

La première expérience est présentée dans la Figure IV.20. Dans cette expérience, la position de départ et la position d'arrivée ont été pris alignés, et aucun obstacle ne se trouve entre ces deux derniers. Le système mobile suit parfaitement les directions globales du chemin initialement calculées.

Dans la deuxième expérience, les points de départ et d'arrivée ne sont pas alignés (Figure IV.21). Il n'y aucun obstacle se trouvant sur le chemin du système mobile. Nous remarquons qu'en suivant les directions globales du chemin, le système mobile atteint bien son point d'arrivée en décrivant un chemin presque identique à celui initialement calculé.

La troisième expérience est similaire à la première, les points de départ et d'arrivée sont pris alignés à l'exception d'un obstacle se trouvant entre les deux, bloquant ainsi le chemin optimal initialement calculé. En suivant les directions globales du chemin optimal, le système mobile détecte l'obstacle, replanifie un nouveau chemin optimal lui permettant de l'éviter. En évitant ce dernier, le système mobile entre dans une zone dont les cellules ne possèdent aucune information sur les directions globales du chemin parce qu'elles entourent l'obstacle et que l'algorithme ne les prend pas en compte. Tant que le système mobile reste dans cette zone, il sera comme aveugle, c'est-à-dire, il détecte l'obstacle mais le chemin optimal ne change pas, il augmente les appels à la fonction *Case_D** qui restent sans succès, car elle ne prend pas en compte les cellules qui entourent le système mobile. Sans information a priori, le système mobile exécute la dernière direction globale du chemin optimal lue.

En conclusion, en prenant en compte la contrainte géométrique, le système mobile peut se trouver dans une zone à risque. Il faut donc lui donner une autre information (locale) lui permettant de sortir de cette zone.

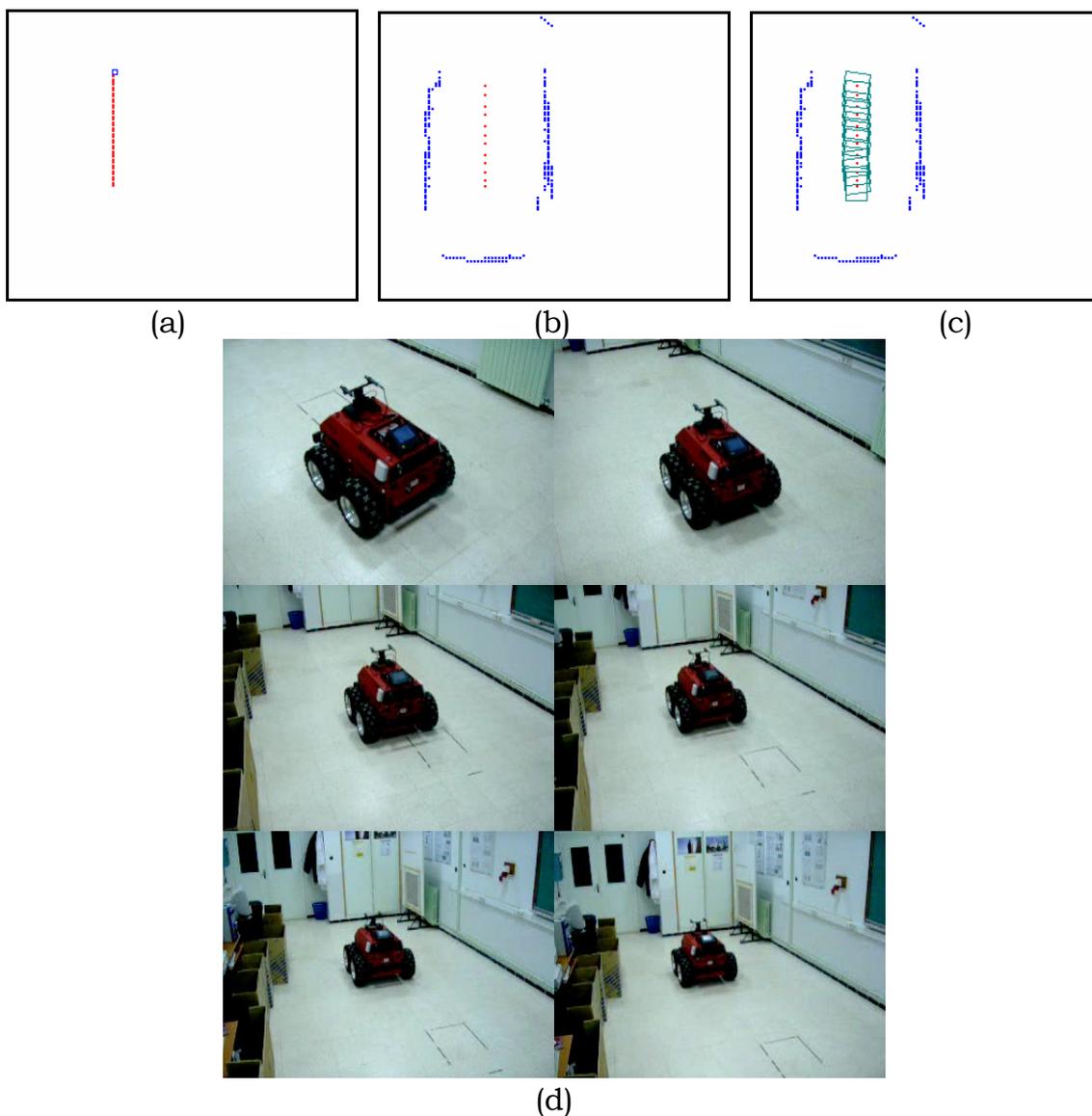
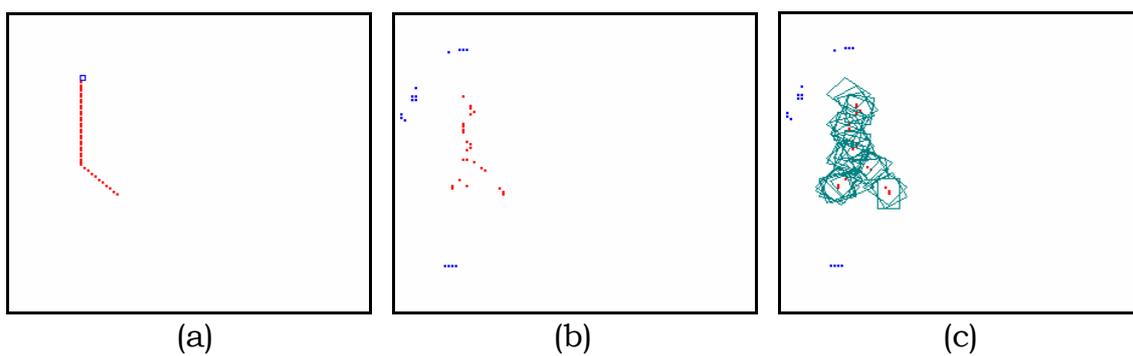


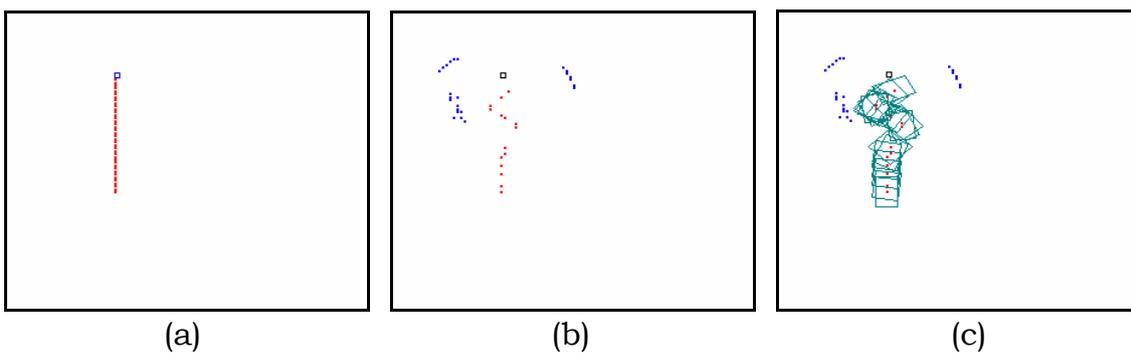
Figure IV.20 : Première expérimentation : (a) le chemin optimal initialement calculé; (b et c) le chemin emprunté par l'ATRV2; (d) La scène réelle.





(d)

Figure IV.21 : Deuxième expérimentation : (a) le chemin optimal initialement calculé; (b et c) le chemin emprunté par l'ATRV2; (d) La scène réelle.



(a)

(b)

(c)



(d)

Figure IV.22 : Troisième expérimentation : (a) le chemin optimal initialement calculé; (b et c) le chemin emprunté par l'ATRV2; (d) La scène réelle.

IV.3.2. L'algorithme Champs de Forces Virtuelles ou *VFF* (Virtual Force Field).

IV.3.2.1. Introduction.

L'algorithme de navigation à base des champs de forces virtuelles est un algorithme de navigation non stop n'utilisant que les données courantes des capteurs US (sur une petite fenêtre temporelle), et non des données provenant d'un modèle interne, pour décider de l'action à effectuer. Cet algorithme dérive des méthodes dites à base de champs de potentiels réalisant l'association perception-action à l'aide d'une fonction générant des chemins cherchant à minimiser un critère donné (la fonction potentiel).

La méthode à base des champs de forces virtuelles (*VFF*) a été développée par *Borenstein* et *Koren* dans [Bor88]. Son but est de permettre à un système mobile rapide de contourner un obstacle imprévu détecté par ses capteurs embarqués.

Les mesures des différents capteurs embarqués sont intégrées dans une grille de dimension deux appelée *grille histogramme*. Cette grille est inspirée des travaux de *Moravec* et *Elfes* dans [Mor85] mais seule la cellule correspondant à la mesure

d'un capteur a son degré d'activation incrémenté de manière à obtenir une mise à jour plus rapide.

Lorsque le système mobile se déplace, de manière à limiter son champ de vue, seuls les échos capteurs situés dans une fenêtre (appelée fenêtre active) de dimensions $w_s \times w_s$, fixée par l'utilisateur, sont considérés (Figure IV.23). Chaque cellule occupée par un obstacle, de coordonnées (x_l, y_m) dans la grille, exerce sur le système mobile une force répulsive élémentaire \vec{F}_{lm} dirigée vers le système mobile et dont l'amplitude est fonction du degré d'activation de la cellule ainsi que de la distance la séparant du système mobile. La force répulsive finale \vec{F}_{rep} est alors la somme de toutes ces forces élémentaires. Elle est calculée comme suit :

$$\begin{cases} \vec{F}_{lm} = \frac{F_{cr} W^n C_{lm}}{d^n(l,m)} \left(\frac{x_l - x_0}{d(l,m)} \vec{i} + \frac{y_m - y_0}{d(l,m)} \vec{j} \right) \\ \vec{F}_{rep} = \sum_{l,m} \vec{F}_{lm} \end{cases} \quad (IV.51)$$

où F_{cr} est une constante positive, W la largeur du système mobile, C_{lm} le degré d'activation de la cellule de coordonnées (x_l, y_m) , (x_0, y_0) la position du système mobile, $d(l,m)$ la distance séparant la cellule du système mobile et n est un nombre positif.

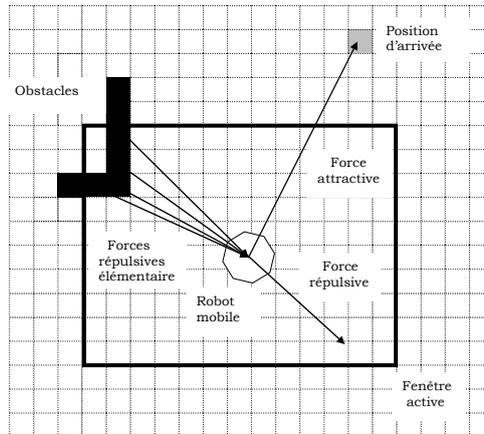


Figure IV.23 : Principe du Champ de Forces Virtuelles.

Simultanément à cette force de répulsion \vec{F}_{rep} , le système mobile est également soumis à une force d'attraction \vec{F}_{att} provenant du but, elle est calculée par l'équation suivante:

$$\vec{F}_{att} = F_{ct} \left(\frac{x_t - x_0}{d_t} \vec{i} + \frac{y_t - y_0}{d_t} \vec{j} \right) \quad (IV.52)$$

où F_{ct} une constante positive, (x_t, y_t) les coordonnées du point d'arrivée et d_t la distance séparant le système mobile du but.

La somme de ces deux forces (\vec{F}_{rep} et \vec{F}_{att}) donne le vecteur résultant \vec{R} et dont la direction indique la direction devant être suivie par le système mobile. Il est exprimé par la formule suivante :

$$\vec{R} = \vec{F}_{att} + \vec{F}_{rep} \tag{IV.53}$$

La vitesse linéaire est quant à elle fonction de l'angle θ entre le vecteur vitesse courant et le vecteur \vec{F}_{rep} , et a pour valeur:

$$V = \begin{cases} V_{max} & \text{si } \|\vec{F}_{rep}\| = 0 \\ V_{max} (1 - |\cos \theta|) & \text{si } \|\vec{F}_{rep}\| > 0 \end{cases} \tag{IV.54}$$

La vitesse est maximum si aucun obstacle n'est présent ou s'ils sont tous latéraux ($\cos \theta = 0$).

Nous allons maintenant décrire l'algorithme de navigation que nous avons réalisé basé sur les champs de forces virtuelles.

IV.3.2.2. Algorithme de navigation développé.

L'algorithme développé est une extension des travaux présentés dans [Dje05] et similaire au concept détaillé dans [Bor91a], mais diffère dans le sens où il utilise uniquement l'orientation du vecteur \vec{R} pour calculer les vitesses de translation et de rotation du système mobile ATRV2. L'orientation du vecteur \vec{R} détermine le cap du système mobile ATRV2 lui permettant de rejoindre son point d'arrivée désigné.

Le vecteur \vec{R} est obtenu par l'équation IV.53, la force attractive \vec{F}_{att} est obtenue par l'équation IV.52, tandis que la force répulsive est divisée en deux forces : force répulsive frontale et force répulsive latérale, à cause de la forme rectangulaire du système mobile ATRV2 [Bor91a].

Ces forces sont obtenues en tenant compte des obstacles qui se trouvent à l'intérieur d'une fenêtre de dimension $w_s \times w_s$, et dont le centre est le centre de gravité du système mobile ATRV2.

a. Calcul de la force répulsive frontale.

Au cours de son déplacement, le système mobile ATRV2 calcule les forces répulsives que les obstacles frontaux (détectés par les six capteurs US frontaux) exercent sur lui et plus précisément au point CP_1 (Figure IV.24). La somme de toutes ces forces répulsives donne la force frontale \vec{F}_F avec la direction θ_F . Cette force pousse le système mobile loin des obstacles qui lui sont plus proches.

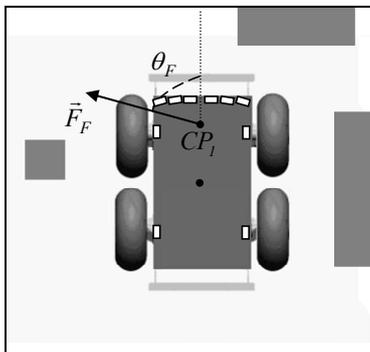


Figure IV.24 : Calcul de la force répulsive frontale \vec{F}_F .

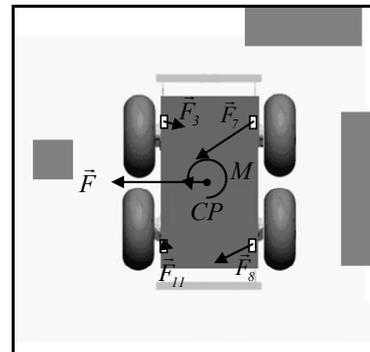


Figure IV.25 : Calcul des forces latérales.

b. Calcul de la force répulsive latérale.

Une force répulsive \vec{F}_i est appliquée à chaque capteur à ultrasons C_i des deux côtés du système mobile ATRV2. La résultante de ces forces sera ensuite remplacée par une simple force \vec{F} et un moment \vec{M} appliqués au centre de gravité CP du système mobile (Figure IV.25). Ensuite, ces derniers seront décomposés en deux couples de forces (F_{1m}, F_{2m}) et deux forces $(\vec{F}_{1f}, \vec{F}_{2f})$ appliqués sur les points CP_1 et CP_2 (Figure IV.26).

Finalement, toutes les forces appliquées au point CP_1 (\vec{F}_F, \vec{F}_{1m} et \vec{F}_{1f}) peuvent être combinées pour calculer le vecteur résultant \vec{R} qui déterminera l'orientation du système mobile (Figure IV.27). Le vecteur \vec{R} est calculé par la formule suivante :

$$\vec{R} = \alpha \vec{F}_{1m} + \beta \vec{F}_{1f} + \gamma \vec{F}_F + \delta \vec{F}_{att} \tag{IV.55}$$

avec α, β, γ et δ des coefficients positifs déterminés expérimentalement.

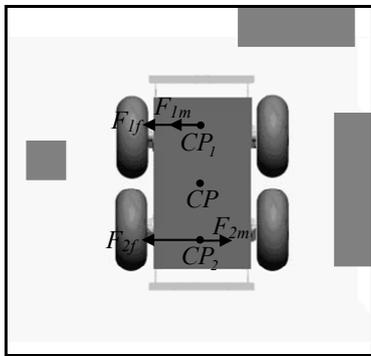


Figure IV.26 : Décomposition des forces latérales.

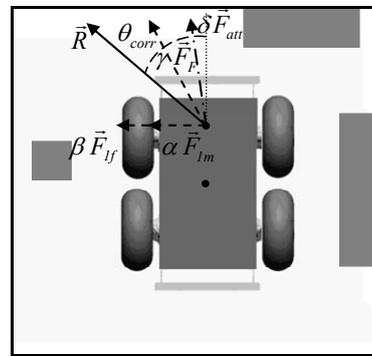


Figure IV.27 : Calcul du vecteur résultant \vec{R} .

IV.3.2.4. Résultats expérimentaux.

L'algorithme de navigation réactive développé ci-dessus a été implémenté sur le système mobile ATRV2, et testé à partir de plusieurs situations réelles de l'environnement. Les données acquises par les différents capteurs embarqués ont été affichées par le programme d'analyse et d'affichage présenté dans la figure III.22.

Dans ce qui suit, nous analysons les résultats expérimentaux de trois expériences réelles effectuées dans notre division, et illustrées dans les figures IV.28- IV.30. Dans chaque expérience, le système mobile se déplace dans un environnement inconnu et ignore l'emplacement des obstacles. Il démarre d'une position initiale et se déplace vers une position finale (but). Les coordonnées cartésiennes de ces deux points sont connues à l'avance par le système mobile.

Dans ces figures, les points rouges représentent le déplacement du centre de gravité du système mobile, les rectangles verts représentent la forme du système mobile en déplacement, et le petit carré blanc représente la position finale.

Le choix des coefficients (α, β, γ) n'est pas simple et à notre connaissance, il n'y a pas de méthode pour les déterminer efficacement et de façon automatique. Nous les avons choisis expérimentalement.

La première expérience décrit un déplacement rectiligne du système mobile entre sa position initiale et sa position finale sans obstacles frontaux. Le système mobile se déplace presque en ligne droite suivant la direction de la force résultante \vec{R} calculée (Figure IV.28.e). Les forces répulsives étaient négligeables pour influencer la direction de la force résultante, et ceci est dû à l'absence d'obstacles proches au système mobile.

La deuxième expérience est semblable à la première, sauf qu'il y a présence de quelques obstacles qui croisent le chemin du système mobile. Au départ, les forces attractive et résultante ont la même direction, le système se déplace donc en ligne droite. Lorsqu'il entre dans la zone d'influence des obstacles (les obstacles sont à l'intérieur de la fenêtre active), il subit des forces répulsives au point où elles modifient sa direction lui permettant ainsi de les éviter (Figure IV.29.e). Une fois sorti de cette zone, le système mobile se dirige directement vers sa position finale.

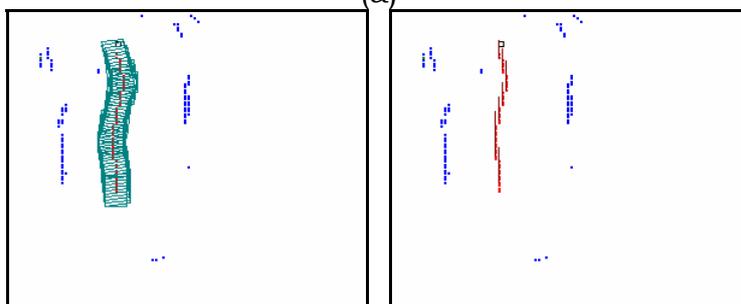
La troisième expérience est utilisée pour analyser le comportement du système mobile lorsqu'il entre dans un minimum local. Dans cette expérience les points de départ et d'arrivée ont été pris alignés, et un ensemble d'obstacles en forme de U est positionné entre ces deux points (Figure IV.30.a).

Initialement, le système mobile se déplace en ligne droite vers sa position d'arrivée. Lorsque les premiers obstacles entrent dans la fenêtre active, l'amplitude des forces répulsives augmente en conséquence, jusqu'à ce qu'elles forcent le système mobile à dévier complètement à droite. Le système mobile essaye d'éviter le plus possible ces obstacles, les forces répulsives sont au maximum de leur amplitude lorsque le système mobile est plus proche d'eux. Et à chaque fois qu'il s'éloigne des obstacles, la force attractive le rappelle vers la position d'arrivée. Ce phénomène se répètera indéfiniment jusqu'à l'arrêt du système mobile par l'utilisateur.

L'inconvénient majeur de cette méthode de navigation est qu'elle repose sur le calcul du cap pour éviter les obstacles. Ce cap est déterminé par la direction du vecteur \vec{R} , qui lui même résulte de la somme de deux forces attractive et répulsive. Lorsque le système mobile est loin des obstacles (pas d'obstacles dans la fenêtre active) le vecteur \vec{R} est identique à la force attractive par conséquent le système mobile se dirige vers la position destination. Lorsque le système mobile est près des obstacles (présence d'obstacles dans la fenêtre active) la force répulsive est plus importante que la force attractive, d'où le vecteur \vec{R} est presque identique à la force répulsive et donc le système mobile s'éloigne des obstacles. Le calcul du cap est uniquement en fonction des données se trouvant à l'intérieur d'une fenêtre de dimension $w_s \times w_s$, donc le système mobile possède une information très restreinte sur son environnement, ce qui explique qu'il emprunte un même chemin plusieurs fois. S'il possédait une connaissance globale (direction globale du chemin) de son environnement, il serait facile pour lui d'éviter les minima locaux.

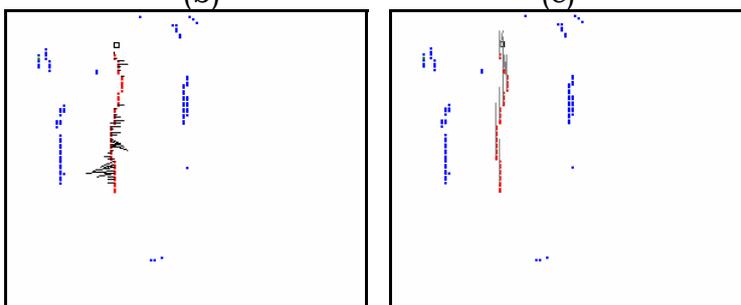


(a)



(b)

(c)



(d)

(e)

Figure IV.28 : Première expérience: (a) Navigation réelle de l'ATRV2, (b) Chemin emprunté par l'ATRV2, (c) les forces attractives obtenues, (d) les forces répulsives obtenues, (e) les forces résultantes \vec{R} obtenues.



(a)

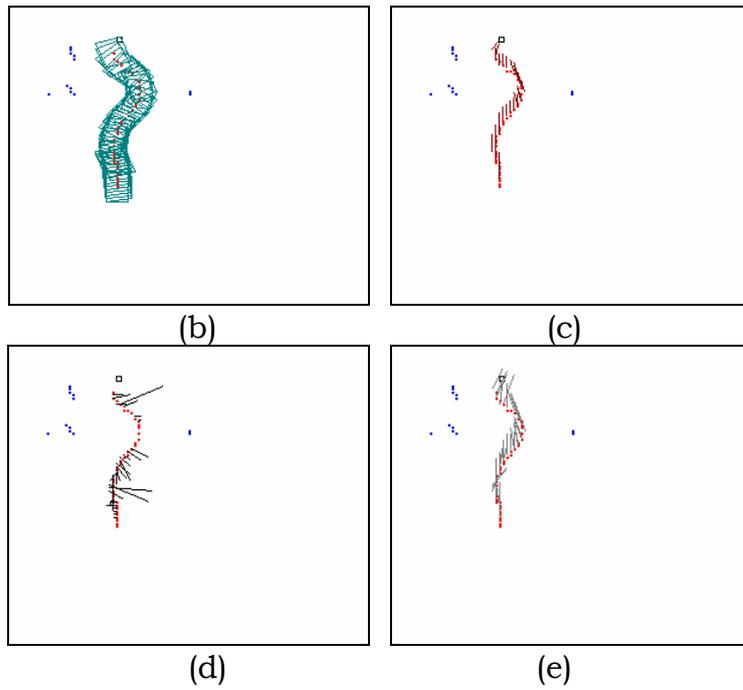


Figure IV.29 : Deuxième expérience: (a) Navigation réelle de l'ATR2, (b) Chemin emprunté par l'ATR2, (c) les forces attractives obtenues, (d) les forces répulsives obtenues, (e) les forces résultantes \vec{R} obtenues.



(a)

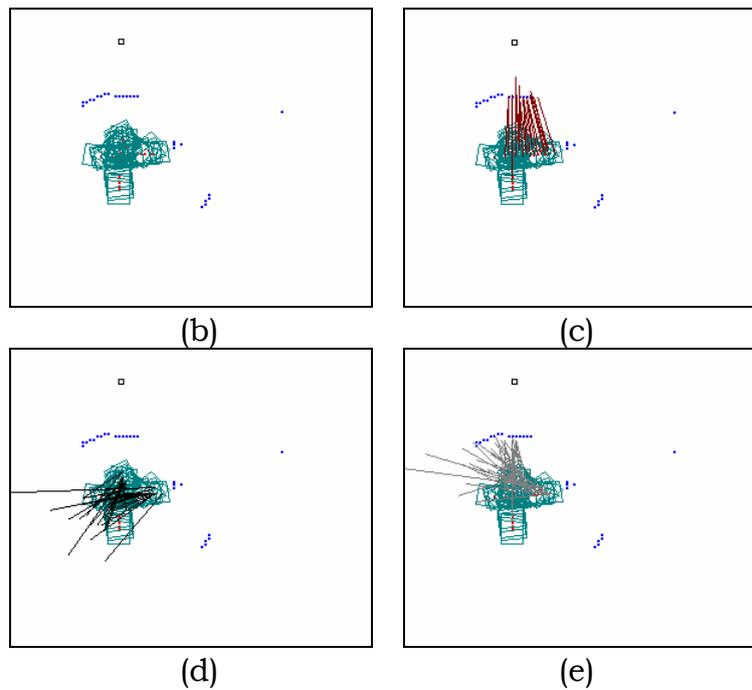


Figure IV.30 : Troisième expérience: (a) Navigation réelle de l'ATRV2, (b) Chemin emprunté par l'ATRV2, (c) les forces attractives obtenues, (d) les forces répulsives obtenues, (e) les forces résultantes \vec{R} obtenues.

IV.3.3. L'approche DVFF.

En combinant l'algorithme de planification de chemin optimal D^* et l'algorithme de navigation non stop basé VFF , nous proposons une nouvelle approche de navigation nommée $DVFF$ qui fait déplacer et de manière robuste un système mobile dans un environnement quelconque (connu, inconnu ou semi connu).

Le diagramme de cette approche est détaillé dans la figure IV.31. Il comporte principalement deux modules :

- Module VFF : Génère le vecteur local \vec{F}_{VFF} avec l'orientation θ_{VFF} . Il est pris en compte lorsque le système mobile est à proximité des obstacles. En suivant la direction θ_{VFF} le système mobile est protégé d'éventuelles collisions avec des obstacles qui lui sont proches.
- Module D^* : Génère le vecteur global \vec{F}_{D^*} avec l'orientation θ_G . θ_G est la direction globale du chemin permettant au système mobile de se diriger vers sa position d'arrivée.

Le module D^* peut être facilement utilisé avec le module VFF parce qu'ils génèrent tous les deux une direction. Ces directions (ou vecteurs) peuvent être utilisées séparément ou combinées pour donner une seule direction. La direction sélectionnée sera utilisée pour calculer la commande du mouvement permettant au système mobile d'éviter les obstacles et atteindre son point d'arrivée.

En suivant cette direction, le système mobile perçoit, à travers ses capteurs embarqués, son environnement en découvrant d'éventuelles divergences avec l'environnement initial (détection d'un nouvel obstacle par exemple). Une fois ses divergences constatées, le système mobile replanifie son chemin optimal, en

appelant la fonction $Case_D^*$, pour rejoindre son point d'arrivée sans risque de collisions.

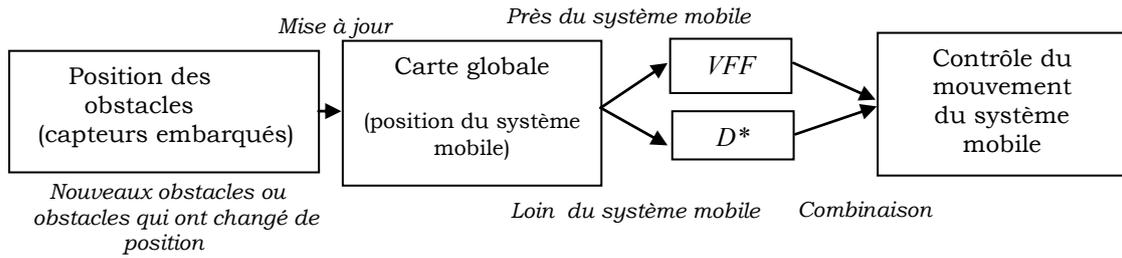


Figure IV.31: Diagramme de l'approche de navigation DVFF.

IV.3.3.1. Premier algorithme [Dje07].

Dans cet algorithme, le système mobile prend en compte, en même temps, les vecteurs générés par les deux modules VFF et D^* .

Le système mobile suit le cap résultant de la somme vectorielle des ces deux derniers. Le vecteur résultant \vec{F} permet au système mobile d'atteindre son but en évitant les obstacles. Actionné au point CP_1 , ce vecteur a pour expression (Figure IV.32):

$$\vec{F} = \vec{F}_{VFF} + \vec{F}_{D^*} = \alpha \vec{F}_{Vf} + \beta \vec{F}_{1m} + \gamma \vec{F}_F + \delta \vec{F}_{D^*} \quad (IV.56)$$

avec α, β, γ et δ des coefficients choisis expérimentalement.

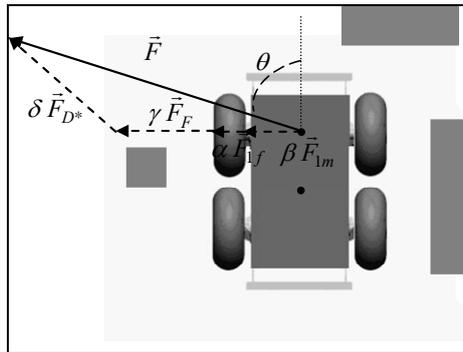


Figure IV.32: Calcul du vecteur résultant \vec{F} .

IV.3.3.2. Deuxième algorithme [Dje09a, Dje09b].

Dans cet algorithme, le système mobile prend en compte, séparément, les vecteurs générés par les deux modules VFF et D^* . Le module VFF ne tient pas compte de la position du point d'arrivée dans le calcul du vecteur local \vec{F}_{VFF} .

Le système mobile utilise la direction du chemin global θ_G sous le point CP_1 si elle existe (le système mobile est loin des obstacles) pour atteindre son point d'arrivée. Et si θ_G n'existe pas, dans ce cas, le système mobile est trop près des obstacles (zone à risque), il utilise la direction locale θ_{VFF} générée par le module VFF lui permettant de s'éloigner des obstacles.

La figure ci-dessous illustre l'organigramme de ce deuxième algorithme et montre comment le système mobile combine ces deux directions (locale et globale) pour une navigation en toute sécurité à partir de sa position initiale vers son point d'arrivée:

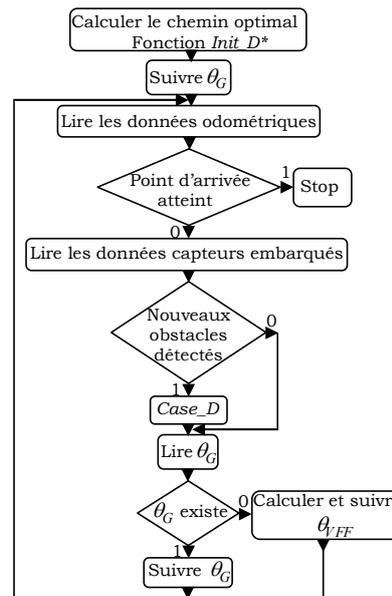


Figure IV.33: Organigramme du deuxième algorithme.

IV.3.3.3. Résultats expérimentaux.

Les deux algorithmes *DVFF* développés ci-dessus ont été implémentés sur le système mobile *ATRV2* et testés à partir de plusieurs situations réelles montrant l'avantage de la combinaison entre l'algorithme de planification de chemin optimale *D** et l'algorithme de navigation non stop basée *VFF*.

Cinq expérimentations (figures IV.34-IV.38) sont présentées illustrant la navigation du système mobile *ATRV2* à travers plusieurs parcours avec différents emplacements d'obstacles. L'indice (a) montre la scène réelle et (b) la grille de certitude de l'environnement reproduite durant chaque expérience. Les cellules occupées par les obstacles sont représentées par des carrés de couleur bleue. Les cellules traversées par le centre de gravité du système mobile sont représentées par des carrés de couleur rouge. Le rectangle de couleur verte représente le système mobile en format réel, et le petit carré blanc représente la position d'arrivée.

Les positions des obstacles et l'environnement sont totalement inconnues du système mobile, seules les positions de départ et d'arrivée sont connues à l'avance. La vitesse de translation a été limitée à $v_c = 20 \text{ cm} / s$ et la vitesse de rotation a été limitée à $0,6 \text{ rd} / s$.

Dans chaque expérience, le chemin optimal initial en tenant compte de la contrainte géométrique du système mobile est calculé, c'est-à-dire, les cellules de la carte entourant les obstacles, selon les dimensions du système mobile, ne sont pas prises en compte par l'algorithme *D** développé.

La première expérience est illustrée par la figure IV.34 (a et b). Elle consiste à faire déplacer le système mobile en ligne droite sans contraintes extérieures (sans obstacles). En utilisant les deux algorithmes *DVFF*, le système mobile arrive à atteindre son point d'arrivée en prenant un chemin rectiligne. Pour le premier

algorithme les deux vecteurs \vec{F}_{VFF} et \vec{F}_{D^*} ont la même direction (dirigées vers le point d'arrivée), et pour le deuxième algorithme le système mobile suit uniquement la direction globale du chemin, dirigée vers le point d'arrivée en absence d'obstacle.

La deuxième expérience (Figure IV.35) illustre la solution donnée par le premier algorithme *DVFF* au problème montré par la troisième expérience illustrée par la figure IV.22. Au départ, le système mobile se déplace en ligne droite car les vecteurs \vec{F}_{VFF} et \vec{F}_{D^*} ont la même direction (dirigés vers le point d'arrivée), et quand il entre dans la zone à risque près des obstacles, où il y a absence d'information sur la direction globale du chemin θ_G ($\|\vec{F}_{D^*}\|=0$), l'information restante (direction locale θ_{VFF}) permet alors au système mobile de sortir de cette zone en évitant les obstacles.

En faisant plusieurs tests, nous avons remarqué qu'en prenant en compte la direction de la résultante des vecteurs \vec{F}_{VFF} et \vec{F}_{D^*} cela ne nous donne pas toujours de bons résultats, surtout dans le cas où l'angle entre ces deux vecteurs est important (cas où le système mobile est pris dans un minimum local). Dans ce cas, la résultante penchera vers le vecteur dominant (qui possède le plus grand module) et dont la direction n'est pas toujours la bonne pour que le système mobile puisse atteindre son point d'arrivée en empruntant un chemin optimal.

La troisième expérience est illustrée par la figure IV.36 (a et b). Elle est similaire à la première expérience, les points de départ et d'arrivée sont pris alignés à l'exception d'un long mur d'obstacles se trouvant entre les deux, coupant ainsi le chemin optimal initialement calculé. En utilisant le deuxième algorithme *DVFF*, le système mobile commence par calculer le chemin optimal initial (l'environnement est inconnu au départ) et suivre ensuite, la direction globale du chemin θ_G sous son point CP_1 jusqu'à arriver à son point d'arrivée. Quand le système mobile détecte le mur, il replanifie un autre chemin optimal, et dès qu'il se trouve dans la zone à risque (près du mur), son module local *VFF* calcule une direction θ_{VFF} qui lui permet de s'éloigner de ce mur et éviter une collision.

Dans la quatrième expérience (Figure IV.37), les points de départ et d'arrivée sont pris alignés mais à gauche du système mobile ATRV2. En utilisant le deuxième algorithme *DVFF*, le système mobile réagit de la même manière que dans la troisième expérience (suivi de la direction globale du chemin θ_G sous son point CP_1 et replanification du chemin optimal une fois les divergences avec l'environnement initial constatées) jusqu'à ce qu'il atteigne son point d'arrivée.

La cinquième expérience illustre la solution donnée par le deuxième algorithme *DVFF* au problème montré par la troisième expérience illustrée par la figure IV.30, quand le système mobile est pris dans un minimum local. Le système mobile ATRV2 réagit parfaitement et sort de son minimum local grâce aux nouvelles directions globales du chemin optimal recalculé quand il a détecté les obstacles frontaux (Figure IV.38).

Le deuxième algorithme *DVFF*, appelé dans la suite de ce document, algorithme *DVFF*, parce qu'il est beaucoup plus robuste que le premier, et que la position du point d'arrivée est prise uniquement dans le calcul du chemin

optimal, c'est-à-dire, que le système mobile n'est pas constamment attiré par le point d'arrivée, et la force attractive n'est pas combinée avec les forces répulsives pour le calcul de la direction finale lorsque le système mobile entre dans la zone à risque. Le système mobile suit, à tout moment de son déplacement, une direction (θ_G ou θ_{VFF}) sous son point CP_1 , qui lui permet d'atteindre son point d'arrivée.

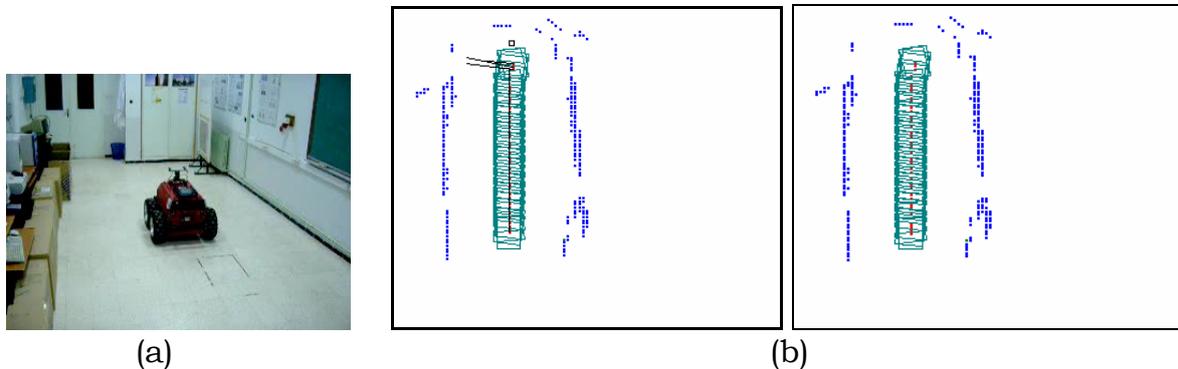


Figure IV.34: Première expérience : (a) Scène réelle; (b) la grille de certitude de l'environnement reproduite durant l'expérience par les deux algorithmes *DVFF* (le premier à gauche et le second à droite).

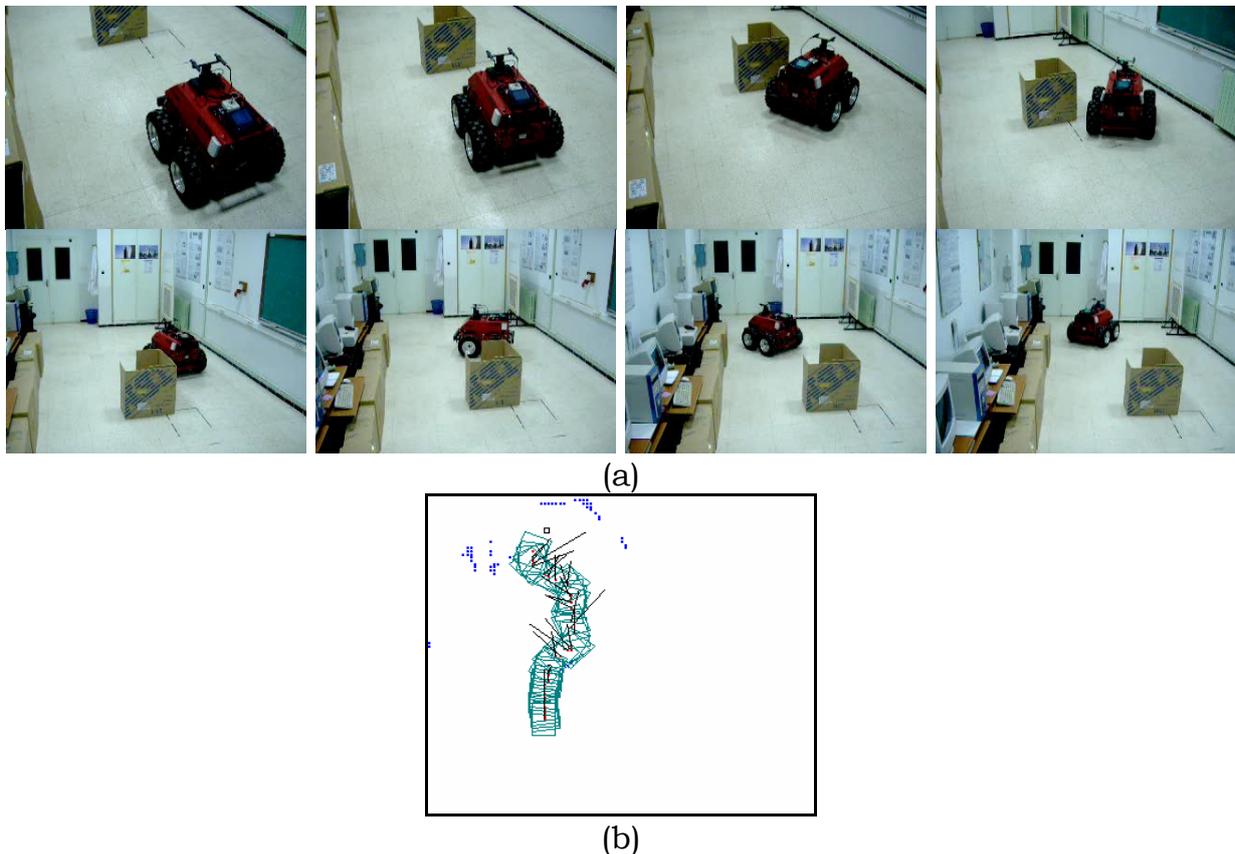


Figure IV.35: Deuxième expérience : (a) Scène réelle; (b) la grille de certitude de l'environnement reproduite durant l'expérience avec les forces résultantes calculées par le premier algorithme *DVFF*.

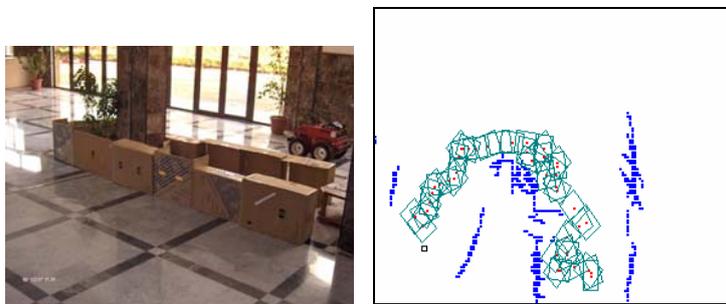


(a)



(b)

Figure IV.36: Troisième expérience: (a) Scène réelle; (b) la grille de certitude de l'environnement reproduite durant l'expérience par le deuxième algorithme DVFF.



(a)

(b)

Figure IV.37: Quatrième expérience: (a) Scène réelle; (b) la grille de certitude de l'environnement reproduite durant l'expérience par le deuxième algorithme DVFF.

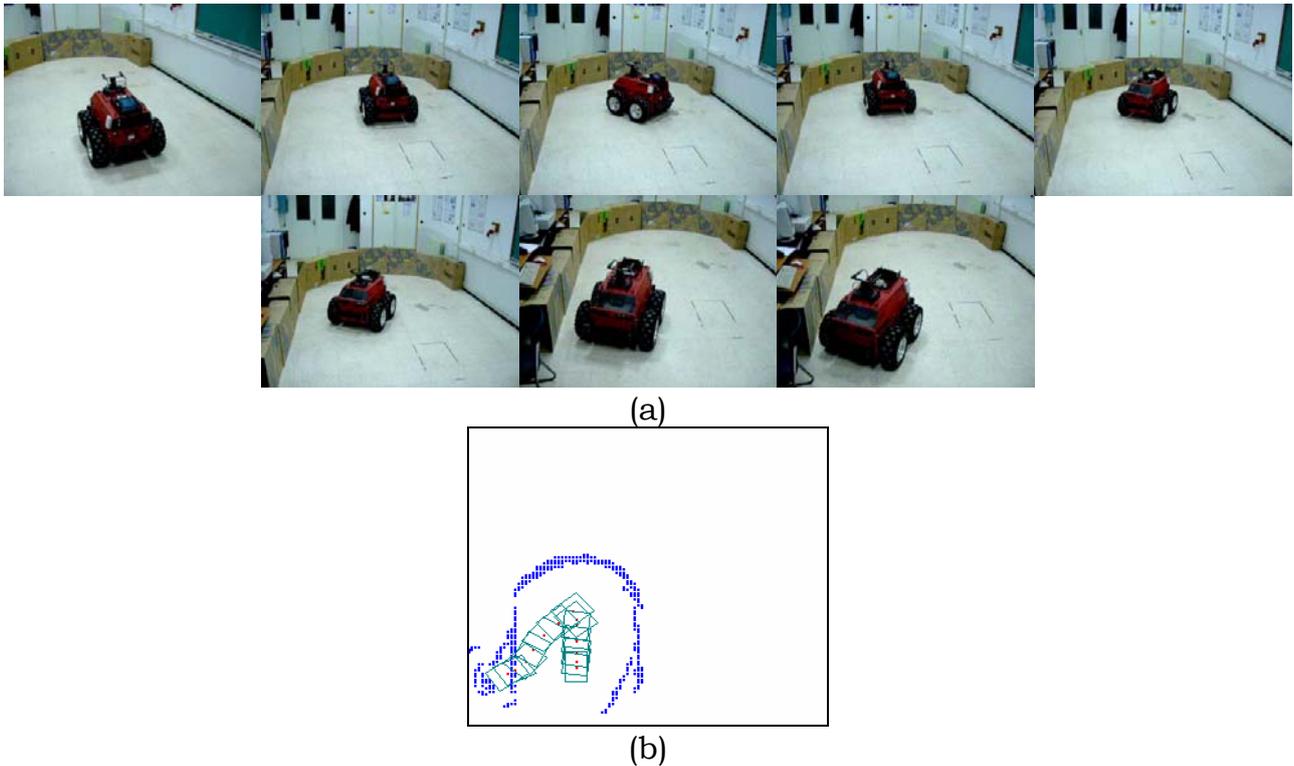


Figure IV.38: Cinquième expérience: (a) Scène réelle; (b) la grille histogramme de l'environnement reproduite durant l'expérience par le deuxième algorithme DVFF.

IV.4. Contrôle de mouvement du système mobile ATRV2.

Le système mobile ATRV2 est une plateforme à quatre roues motrices différentielles. Son modèle cinématique est décrit par les équations (IV.1) et (IV.2):

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \quad (IV.57)$$

avec (\dot{x}, \dot{y}) vitesses linéaires cartésiennes, et (v, ω) vitesses linéaire et angulaire (translation et rotation) et qui ont pour expression :

$$\begin{cases} v = \frac{v_R + v_L}{2} \\ \omega = \frac{v_R - v_L}{2d} \end{cases} \quad (IV.58)$$

avec v_R et v_L vitesses linéaires des roues fictives gauche et droite et d la distance entre ces deux dernières (Figure IV.7).

Si les deux roues roulent en avant avec la même vitesse ($v_R = v_L$), le système mobile se déplace vers l'avant. Si elles roulent en sens opposé ($v_R = -v_L$), le système mobile tourne autour de lui-même, et si elles roulent vers l'avant avec des vitesses différentes ($v_R \neq v_L$), le système mobile décrit un virage du côté roulant avec la vitesse la plus faible, de rayon R autour de son ICR (*Instantaneous Center of Rotation*). La valeur de R est obtenue par l'expression suivante (Figure IV.39) :

$$R = d \frac{v_R + v_L}{v_R - v_L} \quad (\text{IV.59})$$

Quand la direction de déviation θ_{Dev} (θ_{VFF} ou θ_G) obtenue à partir de l'algorithme de navigation DVFF (détaillé ci-dessus) est connue, ainsi que l'orientation du système mobile θ_{Rob} donnée par le système de localisation, la direction (ou déviation) de commande θ_c donnée comme consigne au système mobile est calculée par l'équation suivante :

$$\Delta\theta = \theta_{Dev} - \theta_{Rob} \quad (\text{IV.60})$$

$$si(-\pi \leq \Delta\theta \leq \pi) \text{ alors } \theta_c = \Delta\theta.$$

$$sinon si(\Delta\theta < -\pi) \text{ alors } \theta_c = 2\pi + \Delta\theta. \quad (\text{IV.61})$$

$$sinon si(\Delta\theta > \pi) \text{ alors } \theta_c = \Delta\theta - 2\pi.$$

Les valeurs des vitesses linéaire et angulaire désirées (consignes) v_d et ω_d , données au système mobile, peuvent être calculées en fonction de la valeur de la direction de commande θ_c de la manière suivante:

$$si \left(-\pi/180 \leq \theta_c \leq \pi/180 \right) \begin{cases} \text{marche avant.} \\ v_d = v_c. \\ \omega_d = 0. \end{cases} \quad (\text{IV.62})$$

$$sinon si \left(\theta_c = \pi/2 \right) \begin{cases} \text{tourner à gauche.} \\ v_d = 0. \\ \omega_d = \frac{v_c}{d}. \end{cases} \quad (\text{IV.63})$$

$$sinon si \left(\pi/180 \leq \theta_c < \pi/2 \right) \begin{cases} \text{tourner à gauche.} \\ R = D / \text{tg } \theta_c. \\ v_d = \frac{v_c}{2} \left(1 + \frac{R-d}{R+d} \right). \\ \omega_d = \frac{v_c}{2d} \left(1 - \frac{R-d}{R+d} \right). \end{cases} \quad (\text{IV.64})$$

$$sinon si \left(\pi/2 \leq \theta_c \leq 175\pi/180 \right) \begin{cases} \text{tourner à gauche puis marche avant.} \\ v_d = 0. \\ \omega_d = \frac{v_c}{d}. \end{cases} \quad (\text{IV.65})$$

$$sinon si \left(\theta_c = -\pi/2 \right) \begin{cases} \text{tourner à droite.} \\ v_d = 0. \\ \omega_d = -\frac{v_c}{d}. \end{cases} \quad (\text{IV.66})$$

$$\text{sinon si } \left(-\frac{\pi}{2} < \theta_c \leq -\frac{\pi}{180}\right) \left\{ \begin{array}{l} \text{tourner à droite.} \\ R = \frac{D}{\text{tg } \theta_c} \\ v_d = \frac{v_c}{2} \left(1 + \frac{R+d}{R-d}\right) \\ \omega_d = \frac{v_c}{2d} \left(\frac{R+d}{R-d} - 1\right) \end{array} \right. \quad (\text{IV.67})$$

$$\text{sinon si } \left(-\frac{175\pi}{180} \leq \theta_c \leq -\frac{\pi}{2}\right) \left\{ \begin{array}{l} \text{tourner à droite puis marche avant.} \\ v_d = 0 \\ \omega_d = -\frac{v_c}{d} \end{array} \right. \quad (\text{IV.68})$$

$$\text{sinon Situation de Blocage.} \quad (\text{IV.69})$$

avec v_c constante fixée expérimentalement.

La valeur de la direction de commande θ_c détermine l'angle de déviation que le système mobile doit faire. Si cette valeur est comprise entre $-175\pi/180$ et $175\pi/180$, le système mobile est dans une situation de blocage, il recalcule un nouveau chemin optimal vers son but, et si cette situation persiste, le système mobile doit tourner sur lui-même de π (ou $-\pi$) et faire ensuite un déplacement vers l'avant.

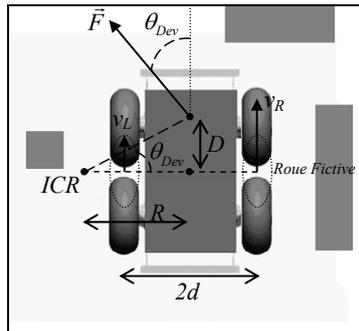


Figure IV.39: Modèle cinématique du système mobile ATRV2 et sa commande de mouvement.

IV.5. Conclusion.

Dans ce chapitre, nous avons présenté en premier les caractéristiques spécifiques du système mobile ATRV2, nous avons défini son architecture matérielle et logicielle, ainsi que sa géométrie. La cinématique et le caractère non holonome du système mobile sont des caractéristiques extrêmement importantes pour le développement de notre travail.

Nous avons développé ensuite la méthode de localisation basée sur l'algorithme *EKF* utilisant la position fournie par les odomètres et la position fournie par la mise en correspondance des grilles pour déterminer une estimation de la position réelle du système mobile.

A la fin de ce chapitre, nous avons décrit en détail l'approche de navigation que nous avons développée, appelée *DVFF*. Elle combine deux méthodes existantes, l'algorithme de planification de chemin global D^* pour calculer la direction globale du chemin permettant au système mobile de se diriger vers son point d'arrivée, et

l'algorithme de champ de forces virtuelles (*VFF*) pour l'évitement de nouveaux obstacles détectés.

Les résultats expérimentaux de l'algorithme *DVFF* implémenté sur le système mobile *ATRV2* avec des configurations réelles de l'environnement ont été acceptables, toutefois, les problèmes inhérents à l'utilisation des capteurs ultrasonores tels que les réflexions multiples et la diaphonie mènent à des situations de blocage ou à des réactions imprévues du système mobile.

Conclusion Générale et Perspectives.

Le travail présenté dans notre thèse concerne la localisation et la navigation du système mobile *ATRV2* dans un environnement qui peut être inconnu a priori. Notre motivation était de rendre ce système capable de percevoir et d'utiliser ses données perceptuelles pour augmenter son autonomie de déplacement, pouvoir agir et réagir face à son environnement et accomplir ses tâches de façon autonome tout en évitant les obstacles imprévus. Cet objectif requiert des capacités de traitement pour effectuer la mission demandée. En particulier, il est nécessaire de modéliser l'environnement à partir de ses données perceptuelles issues des capteurs embarqués. Cette étape est primordiale puisqu'elle est à la base des décisions futures sur laquelle reposera le traitement pour établir sa localisation et son déplacement dans son environnement.

Pour construire le modèle de l'environnement, il nous a fallu développer des méthodes d'analyse et d'affichage intégrant tous les algorithmes et procédures développés telles que la lecture des données en provenance des différents capteurs embarqués (US, odomètre et banc stéréoscopique), les calibrages fort et faible pour le banc stéréoscopique, les changements de repères entre les différents capteurs embarqués avec celui du système mobile, et la fusion des données capteurs US-Banc stéréoscopique appartenant à leur champ de vision commun.

Pour le mode de représentation des mesures et de l'environnement, nous avons utilisé les grilles de certitude, car elles sont utilisables dans tout type d'environnement sans nécessiter d'aménagements particuliers, et peuvent être utilisées à partir des mesures issues de plusieurs types de capteurs. Chaque capteur construit sa carte locale qui va ensuite être intégrée à une carte locale du système mobile servant à établir la carte globale de l'environnement. La précision et la robustesse du modèle de l'environnement ainsi obtenu ont été testées dans l'étape de localisation.

Nous avons ramené le problème de la localisation de notre système mobile à une combinaison entre le filtre de *Kalman* étendu et la mise en correspondance de cartes (locale et globale). L'algorithme de localisation obtenu a été testé et validé sur un autre système mobile disponible dans notre laboratoire (le système mobile *B21r*) ensuite utilisé sur le système mobile *ATRV2*. Cet algorithme fusionne les données odométriques (modèle du système odométrique) et les données du système de perception embarqué (mesure) pour corriger régulièrement l'estimation relative de la position du système mobile due à l'erreur cumulative engendrée par les capteurs odométriques. La mesure est obtenue par la mise en correspondance, dont la zone de recherche est estimée par l'algorithme autour de la position estimée.

L'algorithme de navigation autonome, appelé *DVFF*, que nous avons développé est un algorithme hybride qui combine à la fois deux méthodes de navigation

existantes, l'algorithme de planification de trajectoire globale basé sur l'algorithme D^* , et l'algorithme de navigation réactive non stop basé sur le principe des champs de forces virtuelles VFF .

Il utilise les valeurs courantes des capteurs ultrasonores embarqués (sur une petite fenêtre temporelle), et les données provenant de son modèle interne, pour décider de l'action à effectuer. Il permet au système mobile *ATRV2* de s'orienter vers sa position destination tout en évitant les collisions avec les obstacles de son environnement.

Malgré la simplicité conceptuelle de l'algorithme $DVFF$ et sa capacité à résoudre, dans la plupart des cas, le problème de la navigation autonome, un certain nombre d'améliorations restent à mener. La première amélioration concerne essentiellement son temps de calcul, car le système mobile est insensible aux différents changements dans la position des obstacles formant son environnement. Celle-ci dépend : du type de programmation utilisé (programmation séquentielle), l'utilisation des données visuelles dans le processus de perception-fusion et la mise en correspondance de cartes dans l'algorithme EKF développé. La deuxième amélioration est celle relative au comportement du système mobile dans son déplacement qui dépend de la commande du mouvement utilisée.

Enfin, lorsque ces améliorations seront achevées, les algorithmes développés dans le cadre de notre thèse seront utilisés dans le projet de recherche Robotique de surveillance lancé par le *CDTA*, incluant les systèmes mobiles comme partie intégrante de son réseau de capteurs de surveillance.

La première proposition dans ce projet de recherche est l'inclusion du système mobile *ATRV2* comme agent autonome dans le système de surveillance. La tâche de surveillance est restreinte à la détection des intrusions dans un environnement structuré. Le système mobile peut ajuster son mouvement en fonction de sa propre interprétation de son environnement proche et de recueillir les informations reçues de ses capteurs embarqués et les transmettre à un superviseur humain. Ce dernier reçoit ces informations sur une interface graphique et commande, si possible, le système mobile par le biais d'un lien RF.

Bibliographie.

- [Aya88] N. Ayache, C. Hansen, "Rectification of images for binocular and trinocular stereovision", *In Proceeding of International Conference on Pattern Recognition (ICPR)*, 1988, pp. 11-16.
- [Avn88] F. Avnaim, J.D. Boissonnat, B. Faverjon, "A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles", *In Proc. IEEE Int. Conf. on Robotics & Automation, Philadelphia, PA (USA)*, Apr. 1988, pp. 1656-1661.
- [App91] A. Appriou, "Probabilités et incertitude en fusion de données multisenseurs", *Revue Scientifique et Technique de la Défense*, 1er trim. , 1991, pp. 27-40.
- [Asc92] P. Aschwander, W. Guggenbul, "Experimental result from a comparative study on correlation-type registration algorithms", *In FORSTNER et RUWIEDEL, Robust computer vision, Wichmann*, 1992, pp. 268-282.
- [Arm93] José Armando Segovia de Los Rios, "Etude des déplacements d'un robot mobile dans un environnement peu contraint", *thèse de doctorat, Université de Technologie de Compiègne, France, Septembre 1993*.
- [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 finder and monocular vision", *3rd European Workshop on Advanced Mobile Robots (Eurobot'99), Zurich, Switzerland, september 6-8, 1999*.
- [Arl99] A. Arleo, del R.J. Millan, D. Floreano, "Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation", *In IEEE Transactions on Robotics and Automation*, 1999, volume 15, pp. 990-1000.
- [Bro83] R.A. Brooks, "Solving the find-path problem by representing free space as generalized cones". *IEEE Transactions on Systems, Man and Cybernetics*, 1983, 13(3), pp. 90-197.
- [Bor88] J. Borenstein and Y.Koren, "High-speed obstacle avoidance for mobile robots", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, n°5, août 1988, pp. 1179-1187.
- [Bor91a] Johann Borenstein, Ulrich Raschke, "Real-time Obstacle Avoidance for Non-Point Mobile Robots", *Fourth World Conference on Robotics Research, Pittsburgh, PA, Sept 1991*, pp. 2.1-2.9.
- [Bor91b] J. Borenstein, Y. Koren, "Histogramic in-motion mapping for mobile robot obstacle avoidance", *In: IEEE Journal of Robotics and Automation, Sscramento, California, 1991, Vol. 7, N° 4*, pp. 535-539.

- [Bor91c] J. Borenstein, Y. Koren, "The Vector Field Histogram- fast obstacle avoidance for mobile robots", *IEEE Transactions on Robotics and Automation*, 1991, 7(3), pp. 278-288.
- [Bar91] J. Barraquand, J.C. Latombe, "Robot motion planning : a distributed representation approach", *International Journal Robotics Research*, , Dec. 1991, 10(6), pp. 628-649.
- [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.
- [Bea92] Paul Beardsley, David Murray, Andrew Zisserman, "Camera Calibration Using Multiple Images", *European Conf. Comp. Vision 1992*, pp. 312--320.
- [Bor94] J. Borenstein, "Internal Correction of Dead-reckoning Errors with the Smart Encoder Trailer", *International Conference on Intelligent Robots and Systems, Munich, Germany, 1994, vol1*, pp. 127-134.
- [Blo94] I. Bloch, "Information Combination operators for data fusion : a comparative review with classification", *SPIE*, 1994, vol. 2315, pp. 148-159.
- [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.
- [Buh95] J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, S. Thrun, "The mobile robot rhino", *AI Magazine*, 1995, 16(1).
- [Bor95] J. Borenstein, "The CLAPPER: A dual-drive Mobile Robot with Internal Correction of Dead-reckoning Errors", *International Conference on Robotics and Automation, University of Michigan, Ann Arbor, USA, 1995, vol. 3*, pp. 3085-3090.
- [Bof95] J. Borenstein, L. Feng, "Correction of systematic odometry errors in mobile robots", *International Conference on Intelligent Robots and Systems, Pittsburgh, Pennsylvania, 1995, vol. 3*, pp. 569-574.
- [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.
- [Bor96a] J. Borenstein, L. Feng, "Measurement and correction of systematic odometry errors in mobile robots", *IEEE Transactions on Robotics and Automation, University of Michigan, Ann Arbor, USA, 1996, vol. 12*, pp. 869-880.
- [Bor96b] J. Borenstein, L. Feng, "Gyrodometry : a new method for combining data from gyros and odometry in mobile robots", *Proceedings of the*

- IEEE International Conférence on Robotics and Automation*, pp. 423-428, 1996.
- [Bor96c] J. Borenstein, H. R. Everett, L. Feng, "Where am I? Sensors and methods for mobile robot positioning". *Rapport technique, Université de Michigan, USA, 1996.*
- [Bor97a] J. Borenstein, H.R. Everett, L. Feng, D. Wehe, "Mobile robot positioning - sensors and techniques". *Journal of Robotic Systems, special issue on mobile robots, April 1997, Invited paper, 14(4), pp.231-249.*
- [Bor97b] J. Borenstein, J. Evans, "The OmniMate Mobile Robot - Design, Implementation, and Experimental Results", *Proceedings of the IEEE International Conférence on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 3505-3510.*
- [Bon97] Philippe Bonnifait, "Localisation précise en position et attitude des robots mobiles d'extérieur à évolutions lentes", *thèse, Ecole Doctorale Science pour l'ingénieur de Nantes, Spécialité : Automatique et Informatique Appliquée, le 24 Novembre 1997.*
- [Bas98a] J. Batista, H. Ara'ujo, A. Almeida, "A.. Iterative multistep explicit camera calibration", *In Proc. Of the 6th International Conference on Computer Vision, Bombay, India, January 1998.*
- [Bas98b] J. Batista, "Iterative Multi-Step Explicit Camera Calibration", *Computer Vision (ICCV98), Bombay, India, January 4-7, 1998.*
- [Bor98] J. Borenstein, "Experimental results from internal odometry error correction with the OmniMate mobile robot", *IEEE Transactions on Robotics and Automation, University of Michigan, Ann Arbor, USA, 1998, vol. 14, pp. 963- 969.*
- [Bou99] J.Y. Bouguet, "Visual methods for three-dimensional modelling", *Thesis California Institute of Technology Pasadena 1999.*
- [Ben02] Mohamed Bénallal, "Système de calibration de Caméra. Localisation de forme polyédrique par vision monoculaire", *Thèse de doctorat, ÉCOLE DES MINES DE PARIS, Novembre 2002.*
- [Bon03] David Bonnafous, "Execution réactive de trajectoires pour robots mobiles non-holonomes". *Thèse doctorat, Institut National Polytechnique de Toulouse, Systèmes Automatiques, décembre 2003.*
- [Bre04] F. Bretar, M. Pierrot-Desseilligny, M. Roux, "Fusion lidar/photogrammétrie : Une mise en cohérence globale". *In Actes du 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, Toulouse, France, Janvier 2004.*
- [Bay05] Bernard Bayle, "Robotique Mobile", *Ecole Nationale Supérieure de Physique de Strasbourg, année 2005-2006*

- [Bou07a] Bouraine Sara, "Contribution à la localisation dynamique du robot mobile d'intérieur B21r en utilisant la plateforme multi sensorielle", *Faculté des Sciences de l'Ingénieur, Département d'Electronique, Université Saad Dahlab, Blida, année universitaire 2006/2007*.
- [Bou07b] S. Bourain, A. O. Djekoune and O. Azouaoui, "Hybrid localisation approach of a bi-steerable mobile robot based on grids matching and Extended Kalman Filter", *The 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China, October 12-15, pp. 1136-1141, 2008*.
- [Cha85] R. Chatila, J. Laumond, "Position referencing and consistent world modelling for mobile robots". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA85), 1985, pp. 138-170*.
- [Can87] J. Canny, "A new algebraic method for robot motion planning and real geometry", In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (Los Angeles, Oct. 22-24), Washington D.C., USA. IEEE, 1987, pp. 39-48*.
- [Can88] J. Canny, "The Complexity of Robot Motion Planning". *MIT Press, Cambridge, MA, USA, 1988*.
- [Cro89] J. Crowley, "World modelling and position estimation for a mobile robot using ultrasonic ranging", *Proc. of IEEE International Conference on Robotics and Automation, Scottsdale, May 1989, pp. 674-680*.
- [Cap90] B. Caprile, V. Torre, "Using Vanishing Points for Camera Calibration", In *Int. Journal of Computer Vision, 1990, 4, pp. 127-140*.
- [Cox91] 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*.
- [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*.
- [Cho97] 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*.
- [Cro97] Alain Crouzil, "Perception du relief et du mouvement par analyse d'une séquence stéréoscopique d'images", *Thèse de doctorat, Université Paul Sabatier de Toulouse, Septembre 1997*.
- [Cip99] R. Cipolla, T. Drummond, D. Robertson, "Camera calibration from vanishing points in images of architectural scenes", In *BMVC, Sep 1999*.

- [Cou00] Alain COURCELLE, "Localisation d'un robot mobile : Application à l'aide à la mobilité des personnes handicapées moteur", *Doctorat de l'Université de Metz, France, janvier 2000*.
- [Cui01] Y.J. Cui, S.S. Ge, "Autonomous Vehicle Positioning with GPS in Urban Canyon Environments", *Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Séoul, Corée, 21-26 mai 2001*.
- [Chu01] H. Chung, L. Ojeda, J. Borenstein, "Sensor fusion for Mobile Robot Dead-reckoning With a Precision-calibrated Fiber Optic Gyroscope", *Proc. of the 2001 IEEE International Conference on Robotics & Automation, Séoul, Corée, 21-26 mai 2001, pp.3588-3593*.
- [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*.
- [Dec92] A. De Cegama, J. Smith, "Neural networks and genetic algorithms for combinatorial optimisation of sensor data fusion", *SPIE, 1992, vol. 1699, pp. 108-115*.
- [Dur95] H. Durrant-Whyte, "The design of a radar-based Navigation System for large outdoor vehicles", *Proc. IEEE Int. Conf. on Robotics and Automation, Nagoya, Aichi, Japan, May 1995, pp. 764-769*.
- [Dub96] Y. Dubé, M. Kaffel, "Safety systems for school buses", *Rapport technique, Département de génie mécanique, Université du Québec à Trois-Rivières, 1996*.
- [Dev97] F. Devernay, "Vision stéréoscopique et propriétés différentielles des surfaces", *Thèse de Doctorat, école Polytechnique de Grenoble, Février 1997*.
- [Del97] L. Delahoche, 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*.
- [Del98] L. Delahoche, C. Pégard, M. Mouaddib, P. Vasseur, "Incremental map building for mobile robot navigation in an indoor environment", *Proceedings of International Conf. on Robotics and Automation (ICRA'98), Belgium, May 1998*.
- [Dud00a] G. Dudek, M. Jenkin, "Computational Principles of Mobile Robotics", *Cambridge University Press, 2000*.
- [Dud00b] G. Dudek, D. Jugessur, "Robust Place Recognition using Local Appearance based Methods", *IEEE International Conference on Robotics and Automation, 2000, pp 1030-1035*.
- [Dea00] M. Deans, M. Hebert, "Invariant filtering for simultaneous localization and mapping", *Proc. IEEE International Conference on Robotics and Automation, April 2000, pp. 1042-1047*.

- [Doi02] L. Doitsidis, K.P. Valavanis, N.C. Tsourveloudis, "Fuzzy logic based autonomous skid steering vehicle navigation", *In Proceedings of the International Conference on Robotics and Automation, Washington DC, USA*.
- [Dro02] Cyril Drocourt, "Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels", *Thèse doctorat, université de technologie de Compiègne, février 2002*.
- [Dje05] A. O. Djekoune, K. Achour, R. Toumi, "Ultrasonic sensing based navigation for mobile robot with obstacles avoidance", *IEEE International Computer Systems & Information Technology Conference, IEEE CSIT'05, Hilton du 19 au 21 juillet 2005*.
- [Dje07] O. Djekoune, K. Achour and Redouane Toumi, "A new approach for mobile robot navigation ", *5^{ème} Conférence sur le Génie Electrique, CGE'05, le 16 et 17 avril 2007, École Militaire Polytechnique, Bordj El Bahri, Alger, Algérie*.
- [Dje09a] A. Oualid Djekoune, Karim Achour and Redouane Toumi, "DVFF navigation: A combined approach to real time mobile robot navigation without prior knowledge of the environment", *Journal of Electrical Engineering, Vol. 9, N°. 3, 2009, pp.69-76, ISSN 1582-4594*.
- [Dje09b] A. Oualid Djekoune, Karim Achour and Redouane Toumi, "A Sensor Based Navigation Algorithm for a Mobile Robot using the DVFF approach", *International Journal of Advanced Robotic Systems, Vol. 6, N°2, June 2009, ISSN 1729-8806*.
- [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 Conférence on Decision and Control, Los Angeles CA, December 1987*.
- [Eve95] H.R. Everett, "Sensors for Mobile Robots", *Theory and Application. AK Peters ltd, 1995*.
- [Fav84] B. Faverjon, "Obstacle avoidance using an octree in the configuration space of a manipulator", *In Proc. of the IEEE Int. Conf. on Robotics & Automation, Atlanta, GA (USA), 1984, pp. 504-512*.
- [Fau86] O. Faugeras, G. Toscani, "The calibration problem for stereo", *Proceedings of Computer Vision and Pattern Recognition, Miami Beach, Florida, USA, juin 1986, pp.15-20*.
- [Fra90] G. Frappier, "Système inertiels de navigation pour robots mobiles", *Séminaire Les robots mobiles, EC2, Paris, 1990*.

- [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*.
- [Fau92] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?", *In Proceedings of the 2nd European Conference on Computer Vision (ECCV), mai 1992, pp. 563–578*.
- [Fau93a] O. Faugeras, T. Vieville, E. Theron, J. Vuillemin, B. Hertz, Z. Zhang, L. Moll, P. Bettin, H. Mathieu, P. Fua, G. Berry, C. Proy, "Real time correlation based stereo: Algorithms, implementation and applications", *Rapport de recherche numéro 2013, INRIA, France, 1993*.
- [Fau93b] O. Faugeras, "Three-Dimensional Computer Vision: A Geometric Viewpoint", *MIT press, 1993*.
- [Fau95] O. Faugeras, S. Laveau, L. Robert, G. Csurka, C. Zeller, "3-D reconstruction of urban scenes from sequences of images", *Rapport technique numéro 2572, INRIA, juin 1995*.
- [Fuk96] Y. Fuke, 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*.
- [Fof97] D. Fofi, "Détection d'obstacles par perception visuelle pour un robot mobile", *Rapport de DEA, Ecole Nationale Supérieure de l'Electronique et de ses Applications, Université de Cergy-Pontoise, Août 1997*.
- [Fox97] D. Fox, W. Burgard, S. Thrun, S., "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine, 4(1), 1997, pp. 23-33*.
- [Faj03] B.R. Fajen, W.H. Warren, "A dynamical model of visuallyguided steering, obstacle avoidance, and route selection", *International Journal of Computer Vision, 123(54), 2003, pp.13-34*.
- [Fil04] David FILLIAT, "Robotique Mobile", *Cours C10-2, ENSTA, France, Octobre 2004*.
- [Fig05] Pedro Figueiredo Santana, "Survival Kit: A Bottom Layer for Robot Navigation", *Master degree in Applied Artificial Intelligence, Universidade Nova De Lisboa, 2005*.
- [Gho87] S. K. Ghosh, D. M. Mount, "An output sensitive algorithm for computing visibility graphs", *IEEE Symposium on Foundations of Computer Science, Los Angeles, États-Unis, 1987*.
- [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*.

- [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.
- [Gas99] J. Gasos, A. Saffiotti, "Integrating fuzzy geometric maps and topological maps for robot navigation", *Proc. of the 3rd International ICSC Symposium on Soft Computing (SOCO'99) Gène, Italie, Juin 1999*.
- [Gui00a] J. Guivant, E. Nebot, HF Durrant-Whyte, "Simultaneous localization and map bulding using natural features in outdoor environments", IAS-6 Intelligent Autonomous Systems 25-28 July 2000, Italy, vol 1, July 2000, pp 581-588.
- [Gui00b] J. Guivant, E. Nebot, S. Baiker, "Autonomous navigation and map building using laser range sensors in outdoor applications", *Journal of Robotic Systems*, October 2000, Vol 17, n° 10, pp 565-283.
- [Hri88] C. Harris, M. Stephens, "A combined corner and edge detector", *In Proceedings Alvey Conference, 1988*, pp. 189-192.
- [Hwa92] Y.K. Hwang, N. Ahuja, "Gross motion planning. A survey", *ACM Computing Surveys*, 24(3), 1992, pp.219-291.
- [Hor93] R. Horaud, O. Monga, "Vision par ordinateur, outils fondamentaux", *Traité des nouvelles technologies, série informatique. Hermes, 1993*.
- [Har94] R. Hartley, "Projective reconstruction and invariants from multiple images", *Pattern Analysis and Machine Intelligence (PAMI)*, 16 octobre 1994, pp. 1036-1040.
- [Hor94] O. Horn, G. Riemer, "Dynamic Localisation Method for the V.A.H.M", *SPRANN'94, IMACS-LILLE, France, avril 1994*, pp. 512-517.
- [Hon97] 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.
- [Hol00] Robert Holmberg, Oussama Khatib, "Development and Control of a Holonomic Mobile Robot for Mobile Manipulation Tasks", *International Journal of Robotics Research*, vol. 19, N°11, november 2000, pp. 1066-1074.
- [Har00] R. Hartley, A. Zisserman, "Multiple view geometry in computer Vision", *Cambridge University Press, Combridge, 2000*.
- [Hao04] Yongxing Hao, "A Practical Framework For Formation Planning And Control Of Multiple Unmanned Ground Vehicles", *Doctrat of Philosophy in Mechanical Engineering, University of Delaware, 2004*.
- [Iro01] *iRobot Corporation*, "ATRV All-Terrain Mobile Robot User's Guide", *Part Number: 3969, Rev.2, Copyright 2001*.

- [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.
- [Jou01] I.S. Joung, H.S. Cho, "Self localization for mobile robots by matching of two consecutive environmental range data", *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Séoul, Corée, 21-26 mai 2001, pp. 1603-1608.
- [Kha86] O. Khatib, "Real-time obstacle avoidance for robot manipulator and mobile robots", *The International Journal of Robotics Research*, 1986, 5(1), pp. 90-98.
- [Kor88] Y. Koren, J. Borenstein, "High-speed obstacle avoidance for mobile robotics", *In Proceedings of the IEEE Symposium on Intelligent Control*, Arlington, VA, pp. 382-384.
- [Kor90] Y. Koren, J. Borenstein, "Analysis of Control Methods for Mobile Robot Obstacle Avoidance", *in Proc. IEEE Int. Workshop Intelligent Motion Control*, 1990.
- [Kui91] B. Kuipers, Y.T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations", *Robotics and Autonomous Systems*, 8, 1991.
- [Kha93] O. Khatib, S. Quinlan, "Elastic bands: connecting, path planning and control", *In Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, GA.
- [Koo96] 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.
- [Kav96] L. Kavraki, M. N. Kolountzakis, J.C. Latombe, "Analysis of probabilistic roadmaps for path planning", *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 1996, pp.3020-3026.
- [Kar97] O. Karch, H. Noltemeier, "Robot Localization: Theory and Practice", *Proceedings of the International Conférence, IROS'97*, pp. 850-856.
- [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.
- [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.
- [Kko98] N.Y. Ko, R. Simmons, "The lane-curvature method for obstacle avoidance", *In Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 98*, 1998, Victoria, B.C., Canada.

- [Ken99] Simon Kent, "Evolutionary Approaches to Robot Path Planning", *Doctor thesis of Philosophy, Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, United Kingdom, March 1999.*
- [Lag87] Ch. Laugier, "Raisonnement géométrique et méthodes de décision en robotique. Application à la programmation automatique des robots", *Thèse d'état, Inst. Nat. Polytechnique de Grenoble, (F), Dec. 1987.*
- [Lum90] V. Lumelsky, T. Skewis, T. "Incorporating range sensing in the robot navigation function", *IEEE Transactions Systems, Man, Cybernetics, 1990, 20(5), pp. 1058-1069.*
- [Lat91] J.C. Latombe, "Robot motion planning", *Kluwer Academic Publishers. Cambridge, MA, USA, 1991.*
- [Lan92] B. Langlois, J. Barraquand, J.C. Latombe, "Numerical potential field techniques for robot path planning", *IEEE Transactions On Systems, Man, and Cybernetics, , 1992, 22(2), pp. 224-241.*
- [Leo92] J. Leonard, H.F. Durrant-Whyte, J. Cox, "Dynamic map building for an autonomous mobile robot", *The International Journal of Robotics Research, Vol. 11, N° 4, August 1992, pp. 286-298.*
- [Loz92] T. Lozano-Perez, J.L. Jones, E. Mazer, and P.A. O'Donnell, "HANDEY: A Robot Task Planner", *M.I.T. Press, Cambridge, MA, 1992.*
- [Lau98] J.P. Laumond, "Robot Motion Planning and Control", *Lectures Notes in Control and Information Sciences 229, Springer.*
- [Lal98] A. Lallement, M. Dufaut, R. Husson, "Sensor cooperation and world perception of the autonomous mobile robot Romane", *Third IFAC Symposium on Intelligent Autonomous Vehicles, Madrid, Espagne, Mars 1998.*
- [Loo99] C. Loop, Z. Zhang, "Computing Rectifying Homographies for Stereo Vision", *IEEE Conf. Computer Vision and Pattern Recognition (CVPR), juin 1999, Colorado.*
- [Lar00] Eric LAURIN, "Système intelligent d'assistance à la perception dans la conduite de véhicule", *Mémoire de maîtrise es sciences appliquées Spécialité: génie électrique, Université de SHERBROOKE, janvier 2000.*
- [Lau01] J. P. Laumond, "La robotique mobile", *Hermès Science Publications, 2001, Paris, ISBN 2-7462-0246-8.*
- [Low01] S. Se, D. Lowe, J. Little, "Vision-based Mobile Robot Localization And Mapping using Scale- Invariant Features", *Proc. IEEE International Conference on Robotics and Automation, Seoul, Korea, May 2001, pp. 2051-2058.*

- [Lem05] V. Lemonde, "Stéréovision Embarquée sur Véhicule : de l'Auto-Calibrage à la Détection d'Obstacles", *Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, 2005.*
- [Laf06] Stéphane Laforet, "Evaluation de la sûreté de techniques de navigation autonomes", *Examen Probatoire en Informatique, Conservatoire National Des Arts et Métiers, mars 2006.*
- [Mor79] H. Moravec, "visual mapping by robot rover", *In le Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI), 1979, pp. 598-600.*
- [Man80] American Society of Photogrammetry, "Manual of Photogrammetry", *Fourth Edition, Falls Church, Virginia, Chapters 2 and 4. 1980.*
- [Mor85] H.P. Moravec, A. Elfes, "High Resolution Maps from Wide Angle Sonar". *Proceeding of the IEEE International Conference on Robotics and Automation, St Louis, Missouri, USA, March 1985 , pp. 116-121.*
- [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.*
- [Mat95] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, T. Litwin, "Mars microrover navigation: performance evaluation and enhancement", *Autonomous Robot Journal, special issue on autonomous vehicle for planetary exploration, vol. 2, n° 4, 1995.*
- [Moz95] Moëz Cherif, "Planification De Mouvements Pour Un Robot Mobile Autonome Tout-Terrain: Une Approche Par Utilisation Des Modèles Physiques", *doctorat spécialité Informatique, L'institut national polytechnique de Grenoble, 16 Octobre 1995.*
- [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.*
- [Mar99a] 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.*
- [Mar99b] B. Marhic, "Localisation Absolue de Robots Mobiles par le Capteur de Vision Omnidirectionnelle SYCLOP", *Thèse de doctorat, Université de Picardie Jules Verne, Amiens, Décembre 1999.*
- [Mur00] R.R. Murphy, "Introduction to AI Robotics". *The MIT Press, 2000.*
- [Mez01] Youcef Mezouar, "Planification de trajectoires pour l'asservissement visuel", *Thèse de doctorat, Université de Rennes I, 2001.*
- [Min01] J. Minguez, L. Montano, T. Simeon et R Alami, "Global Nearness Diagram Navigation", *Proceedings of the IEEE International Conference*

- on *Robotics and Automation*, Seoul, Korea, May 21-26, 2001, pp. 33-39.
- [Min04] Javier Minguez, Luis Montano, “Global Nearness Diagram Navigation (GND)”, *IEEE Transactions on robotics and automation*. Février 2004, Vol. 20, N°1.
- [Nei99] J. Neira, J. Tardos, J. Horn, G. Schmidt, “Fusing range and intensity images for mobile robot localization”, *IEEE Transactions on Robotics and Automation*, vol. 15, N°1, Fev. 1999.
- [Ohn94] 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.
- [Ohy94b] A. Ohya, E. Shoji, S. Yuta, “3-D range sensor using fiber grating for recognition of autonomous mobile robots passage space”, *International Conference on Intelligent Robots and Systems*, 1994.
- [Pru88] Alain Pruski, “Robotique générale”, *Ellipses*, 1988, ISBN 2-7298-8838-1.
- [Pol91] POLAROID 1991, “Ultrasonic Ranging System”, *Product Literature*, Polaroid Corporation, 784 Memorial Drive, Cambridge, MA02139, 617-3863964.
- [Pre91] Arturo Preciado-Ruiz, “Sur la modélisation, la localisation et le contrôle d’un robot mobile”, *thèse de doctorat*, Université de Technologie de Compiègne, France, Juillet 1991.
- [Pap96] D. Papadimitriou, T. Dennis, “Epipolar line estimation and rectification for stereo image pairs”, *IEEE Trans. Image Processing*, 1996, Vol. 5, pp. 672-676.
- [Pap97] D. Papadopoulos Orfanos, “Numérisation géométrique automatique à l’aide d’un capteur 3D de précision à profondeur de champ réduite”, *Thèse de doctorat de l’ENST*, 1997.
- [Pol99] M. Pollefeys, R. Koch, L. Van Gool, “A simple and efficient rectification method for general motion”, *Proceeding International Conference on Computer Vision (ICCV)*, 1999, Grèce, pp. 496-501.
- [Roy97] S. Roy, J. Meunier, I. Cox, “Cylindrical Rectification to Minimize Epipolar Distortion”, *Proceeding IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 393-399.
- [Phi04] Roland Philippsen, “Motion Planning And Obstacle Avoidance For Mobile Robots In Highly Cluttered Dynamic Environments”, *Thèse n°3146*, École Polytechnique Fédérale De Lausanne, 2004.
- [Rem04] Anthony Remazeilles, “Navigation à partir d’une mémoire d’images”, *Thèse doctorat*, Université de Rennes I, décembre 2004.

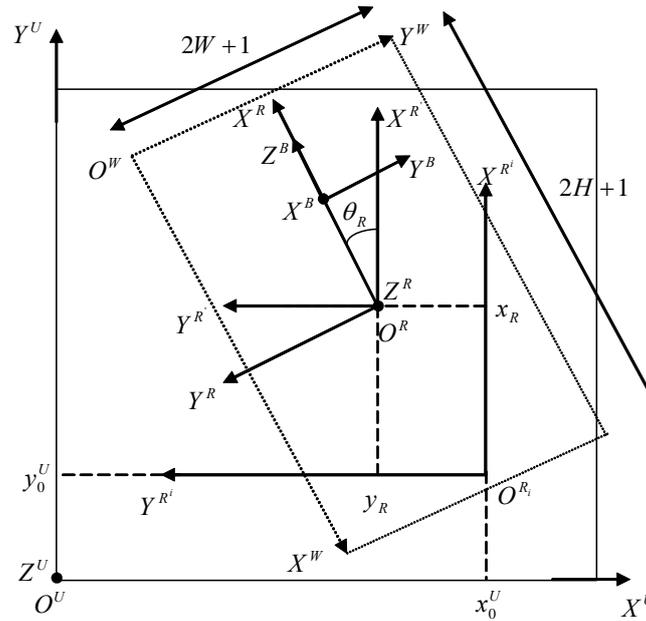
- [Ric04] Vincent Riquebourg, "Identification, Localisation et Suivi de personnes au sein d'un habitat intelligent ", *Diplôme d'études approfondies, université de Rouen et université de Picardie Jules Verne, septembre 2004.*
- [Scw83] J. T. Schwartz, M. Sharir, "On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifold", *Advances in Applied Mathematics, 1983, 4, pp. 298-351.*
- [Sim89] T. Siméon, "Génération automatique de trajectoires sans collision et planification de tâches de manipulation en robotique", *PhD thesis, Lab. d'Automatique et d'Analyse des Systèmes, Toulouse (F), Jan. 1989.*
- [Sab94] A.M. Sabatini, O.D. Benedetto, "Towards a Robust Methodology for Mobile Robot Localisation Using Sonar", *Proceedings of the IEEE International Conférence on Robotics and Automation, 1994, pp. 3142-3147.*
- [Ste94] A. Stenz, "Optimal and efficient path planning for partially known environment", *In Proceedings IEEE International Conference on Robotics and Automation, may 1994.*
- [Sch94] B. Schiele, J.L. Crowley, "A comparison of position estimation techniques using occupancy grids", *Proc. IEEE Int. Conf. on Robotics and Automation, San Diego, California, May 1994, pp 1628-1634.*
- [Sko95] T. Skordas, "Une revue des derniers progress en autocalibration de camera CCD", *Traitement du signal, Vol. 12, N°1, 1995.*
- [Smm96] R. Simmons, "The curvature velocity method for local obstacle avoidance", *In Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, 1996.*
- [Sma96] K. Simsarian, T.J. Olson, N. Nandhakumar, "View invariant regions and mobile robot selflocalization", *IEEE Transactions on Robotics and Automation, October 1996.*
- [Sie04] R. Siegwart, I.R. Nourbakhsh, "Introduction to Autonomous Mobile Robots", *A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England.*
- [Tec] "Traité Mesures et Contrôle", *Techniques de l'Ingénieur, R 7 226, pp.1-9.*
- [Tsa86] R.Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision", *IEEE Computer Vision and Pattern Recognition, Miami Beach Florida, June 1986, pp. 364 -374.*
- [Tos87] G. Toscani, "Système de Calibration et perception du mouvement en vision artificielle", *Thèse de Doctorat, Université Paris Sud, 15 déc. 1987.*

- [Tou88] Pierre Tournassoud, "Géométrie et Intelligence Artificielle pour les robots", *Edition Hermes, Paris 1988*.
- [Tsa89] R.Y. Tsai, "Synopsis of Recent Progress on Camera Calibration for 3D Machine Vision", In *Oussama Khatib, John J. Craig, and Tom'as Lozano-Perez, editors, The Robotics Review, MIT Press, 1989, pp. 147-159*.
- [Tar96] J.P. Tarel, "Estimation géométrique et appariement en modélisation automatique", *Thèse de Doctorat, Université Paris IX-Dauphine, 1996*.
- [Tza99] C.S. Tzafestas, S.G. Tzafestas, "Recent algorithms for fuzzy and neurofuzzy path planning and navigation of autonomous mobile robots", *Systems. Science, 1999, 25, pp. 25-39*.
- [Thn99] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation", *Artificial Intelligence, 1999, 99(1), pp. 21-71*.
- [Thr01] R. Thrapp, C. Westbrook, D. Subramanian, "Robust localization algorithms for an autonomous campus tour guide", *Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Séoul, Corée, 21-26 may 2001, pp. 2065-2071*.
- [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 Conférence on Intelligent Robots and Systems, IROS'98, Victoria, Canada, October 13-17, 1998, pp. 582-587*.
- [Ulr98] I. Ulrich, J. Borenstein, "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots", in *Proc. IEEE Int. Conf. on Robotics and Automation, Leuven, Belgium, May 1998*.
- [Ulr00a] I. Ulrich, I. Nourbaskhsh. "Appearance based Place Recognition for Topological Localization", *IEEE International Conference on Robotics and Automation, 2000, pp. 1023-1029*.
- [Ulr00b] Iwan Ulrich et Johann Borenstein, "VFH*: Local Obstacle Avoidance with Look-Ahead Verification", *Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000, pp. 2505-2511*.
- [Vag93] J. Vaganay, "Conception d'un système multisensoriel de localisation dynamique 3D pour robot mobile", *Thèse de doctorat, LIRMM, Montpellier, juillet 1993*.
- [Vis99] A. Visser, F. Groen, "Organisation and design of autonomous systems. Organisation and Design of Autonomous Systems Course", *M.Sc., Faculty of Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, August 1999*.
- [Van00] D. Van Zwynsvoorde, "Construction incrémentale de modèles topologiques pour la Navigation d'un robot mobile", *Thèse de l'Université Paul Sabatier, Décembre 2000*.

- [Vic00] A. Vicrorino, P. Rives, J.J. Borrelly, "Localisation d'un robot mobile et cartographie par télémétrie laser", *Journées des Jeunes Chercheurs en Robotique, Bourges, France, Février 2000*.
- [Wec98] P. Weckesser, R. Dillmann, "Modeling unknown environments with a mobile robot", *Robotics and Autonomous Systems, Vol. 23, 1998, pp. 293-300*.
- [Yap85] C.K. Yap. "An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments", *Rapport de recherche, Robotics Laboratory, Courant Institute, New-York University, 1985*.
- [Zui92] R.G. Zuidgeest, "Multi sensor data fusion and the use of artificial intelligence", *12^e International Conference Artificial Intelligence, Expert Systems, Natural Language, Nanterre, 1992, pp. 73-84*.
- [Zel96] C. Zeller, "Calibration Projective Affine et Euclidienne en Vision par Ordinateur", *Thèse de doctorat, école Polytechnique, 1996*.

Annexes.

A. Formules de changement de repères : Le système mobile ATRV2 et son environnement.



Repères utilisés

Passage du repère R^W au repère R

$$\begin{cases} x^R = H - x^W \\ y^R = W - y^W \end{cases}$$

Passage du repère R au repère R'

$$\begin{cases} x^{R'} = x^R \cos \theta_R - y^R \sin \theta_R \\ y^{R'} = x^R \sin \theta_R + y^R \cos \theta_R \end{cases}$$

Passage du repère R' au repère R^i

$$\begin{cases} x^{R^i} = x_R + x^{R'} \\ y^{R^i} = y_R + y^{R'} \end{cases}$$

Passage du repère R^i au repère R^U

$$\begin{cases} x^U = x_0^U - y^{R^i} \\ y^U = y_0^U + x^{R^i} \end{cases}$$

Passage du repère R au repère R^U

$$\begin{cases} x^U = x_0^U - y_R - x^R \sin \theta_R - y^R \cos \theta_R \\ y^U = y_0^U + x_R + x^R \cos \theta_R - y^R \sin \theta_R \end{cases}$$

avec :

$R^U(O^U, X^U, Y^U)$ repère univers associé au milieu d'évolution du système mobile,

$R(O^R, X^R, Y^R)$ repère local associé au système mobile. Son déplacement est mesuré par des capteurs odométriques,

$R^W(O^W, X^W, Y^W)$ repère associé à la grille locale centrée sur le système mobile et dont la taille est peut être ajustée sur la plus grande mesure ultrasonore détectée.

$R'(O^R, X^R, Y^R)$ repère intermédiaire utilisé pour le changement de repère de R à R^i ,

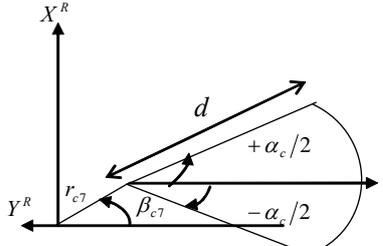
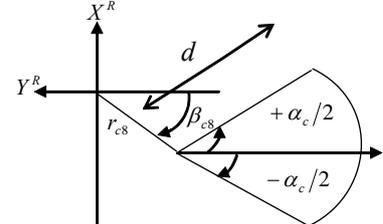
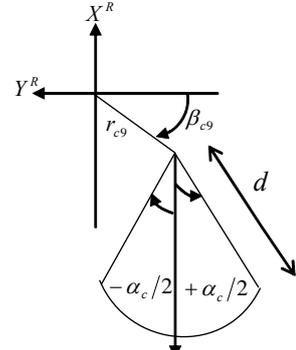
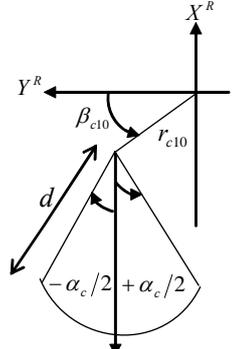
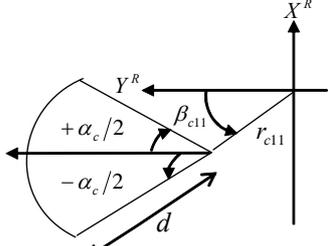
$R^i(O^{R^i}, X^{R^i}, Y^{R^i})$ repère initial associé au système mobile,

(x_0^U, y_0^U) position initiale du système mobile dans son environnement (R^U),

(x_R, y_R, θ_R) la position et l'orientation du système mobile par rapport à son repère initial (R^i).

B. Formules de changement de repères : Les 12 capteurs ultrasonores embarqués

Capteurs	Modèle	Equations
0		$\begin{cases} x_{c0}^R = r_{c0} \sin \beta_{c0} + d \cos \alpha \\ y_{c0}^R = r_{c0} \cos \beta_{c0} + d \sin \alpha \end{cases}$
1 et 2		$\begin{cases} x_{ci}^R = r_{ci} \sin \beta_{ci} + d \sin(\theta_{ci} - \alpha) \\ y_{ci}^R = r_{ci} \cos \beta_{ci} - d \cos(\theta_{ci} - \alpha) \end{cases}$
3		$\begin{cases} x_{c3}^R = r_{c3} \sin \beta_{c3} + d \sin \alpha \\ y_{c3}^R = r_{c3} \cos \beta_{c3} + d \cos \alpha \end{cases}$
4		$\begin{cases} x_{c4}^R = r_{c4} \sin \beta_{c4} + d \cos \alpha \\ y_{c4}^R = -r_{c4} \cos \beta_{c4} - d \sin \alpha \end{cases}$
5 et 6		$\begin{cases} x_{ci}^R = r_{ci} \sin \beta_{ci} + d \sin(\theta_{ci} + \alpha) \\ y_{ci}^R = -r_{ci} \cos \beta_{ci} - d \cos(\theta_{ci} + \alpha) \end{cases}$

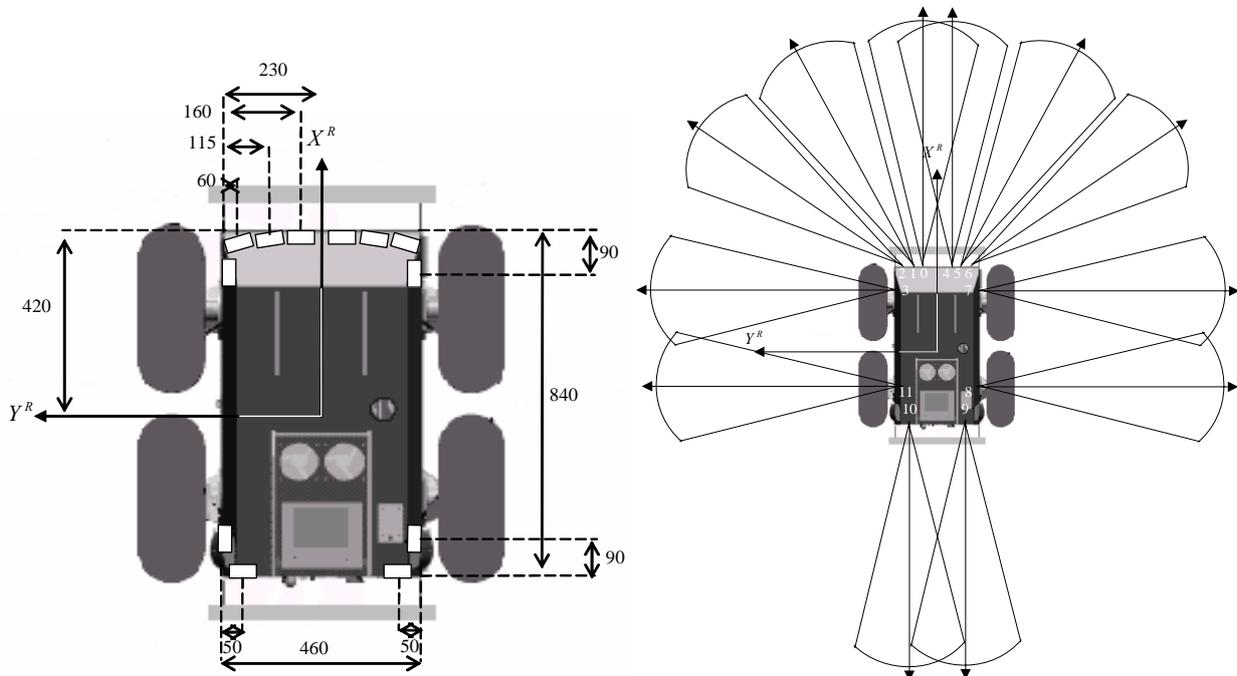
7		$\begin{cases} x_{c7}^R = r_{c7} \sin \beta_{c7} + d \sin \alpha \\ y_{c7}^R = -r_{c7} \cos \beta_{c7} - d \cos \alpha \end{cases}$
8		$\begin{cases} x_{c8}^R = r_{c8} \sin \beta_{c8} + d \sin \alpha \\ y_{c8}^R = -r_{c8} \cos \beta_{c8} - d \cos \alpha \end{cases}$
9		$\begin{cases} x_{c9}^R = r_{c9} \sin \beta_{c9} - d \cos \alpha \\ y_{c9}^R = -r_{c9} \cos \beta_{c9} - d \sin \alpha \end{cases}$
10		$\begin{cases} x_{c10}^R = -r_{c10} \sin \beta_{c10} - d \cos \alpha \\ y_{c10}^R = r_{c10} \cos \beta_{c10} - d \sin \alpha \end{cases}$
11		$\begin{cases} x_{c11}^R = -r_{c11} \sin \beta_{c11} + d \sin \alpha \\ y_{c11}^R = r_{c11} \cos \beta_{c11} + d \cos \alpha \end{cases}$

avec :

r_{ci} , β_{ci} et θ_{ci} paramètres définissant la position du capteur i par rapport au centre de gravité du système mobile,

$\alpha \in [-\alpha_c/2, +\alpha_c/2]$ où α_c est l'angle d'ouverture du cône d'émission du capteur ultrasonore,

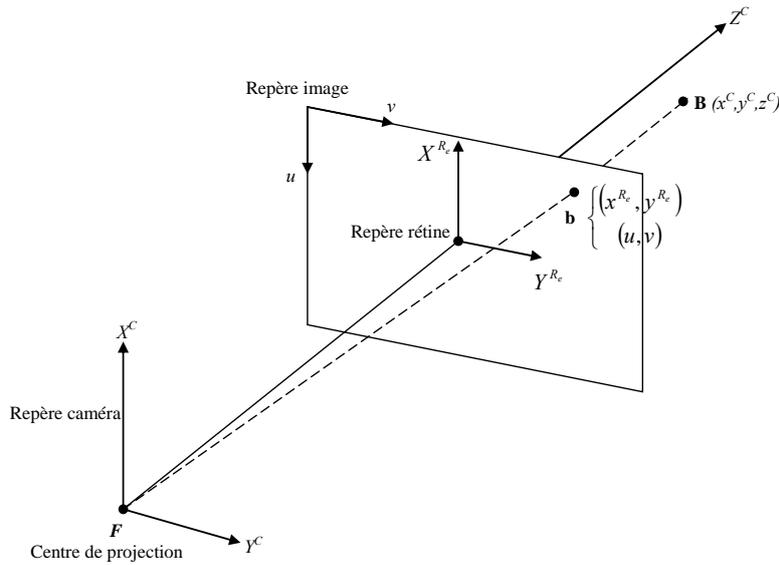
US	r_c (mm)	β_c (degrés)	θ_c (degrés)
0	425.79	80.54	90.0
1	435.46	74.69	105.0
2	453.10	67.96	120.0
3	402.24	55.12	180.0
4	425.79	80.54	90.0
5	435.46	74.69	75.0
6	453.10	67.96	60.0
7	402.24	55.12	0.0
8	402.24	-55.12	0.0
9	456.95	-66.80	270.0
10	456.95	66.80	270.0
11	402.24	55.12	180.0



Positions, orientations et zones couvertes des 12 capteurs ultrasonores de l'ATRV2.

C. Calibrage des caméras et du banc stéréoscopique.

C.1. La caméra: Le modèle *Pinole*.



Modèle de caméra utilisé

Passage du repère *Rétine* au repère *Image*.

$$\begin{cases} u = -k_u x^{R_e} + u_0 \\ v = k_v y^{R_e} + v_0 \end{cases}$$

Passage du repère *Rétine* au repère *Caméra*.

$$\begin{cases} x^C = x^{R_e} \frac{z^C}{f} \\ y^C = y^{R_e} \frac{z^C}{f} \end{cases}$$

avec :

$$\begin{cases} \alpha_u^G = -7.4997925 e + 02 \\ \alpha_v^G = 6.2731322 e + 02 \\ \alpha_u^D = -7.3052412 e + 02 \\ \alpha_v^D = 6.0016980 e + 02 \end{cases} \quad \text{et} \quad \begin{cases} u_0^G = 3.5145071 e + 02 \\ v_0^G = 4.0885937 e + 02 \\ u_0^D = 3.4582972 e + 02 \\ v_0^D = 3.9561875 e + 02 \end{cases}$$

Valeurs calculés à partir d'images acquises de taille 600X560.

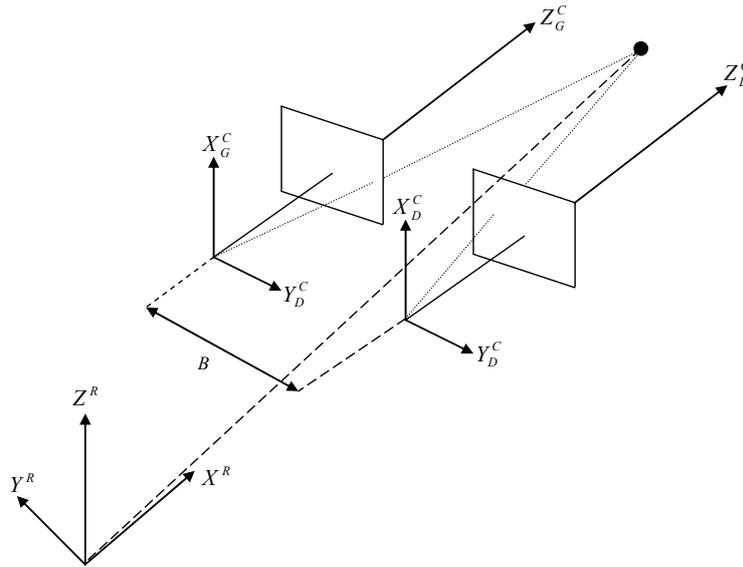
Passage du repère *Image* au repère *Rétine*.

$$\begin{cases} x^{R_e} = \frac{u_0 - u}{k_u} \\ y^{R_e} = \frac{v - v_0}{k_v} \end{cases}$$

Passage du repère *Caméra* (*gauche ou droite*) au repère *Image*

$$\begin{pmatrix} s u \\ s v \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u^i & 0 & u_0^i \\ 0 & \alpha_v^i & v_0^i \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^C \\ y^C \\ z^C \end{pmatrix} \quad (i = G \text{ ou } D)$$

C.2. Le banc stéréoscopique.



Modèle géométrique du banc stéréoscopique : vue de perspective.

Passage du repère *Robot* au repère *Caméra (gauche ou droite)*.

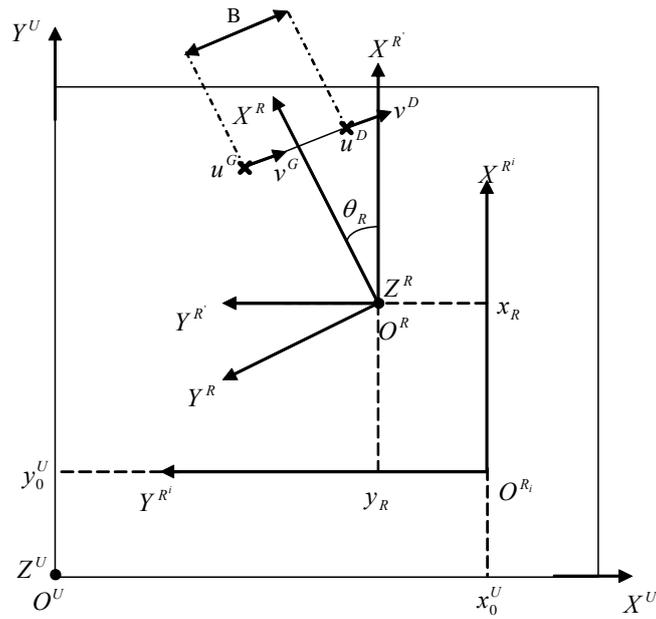
$$\begin{pmatrix} x_i^C \\ y_i^C \\ z_i^C \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11}^i & r_{12}^i & r_{13}^i & t_x^i \\ r_{21}^i & r_{22}^i & r_{23}^i & t_y^i \\ r_{31}^i & r_{32}^i & r_{33}^i & t_z^i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^R \\ y^R \\ z^R \\ 1 \end{pmatrix} \quad (i = G \text{ ou } D)$$

avec :

$$\left\{ \begin{array}{l} r_{11}^G = -0.08271184882285 \\ r_{12}^G = -0.00759219156692 \\ r_{13}^G = -0.97680910896757 \\ r_{21}^G = 0.20829899732750 \\ r_{22}^G = 0.95749299156391 \\ r_{23}^G = -0.03469120677768 \\ r_{31}^G = -1.00000000000000 \\ r_{32}^G = 0.19200474884143 \\ r_{33}^G = 0.06355917643143 \\ t_x^G = 8.131001172488038e+02 \\ t_y^G = -1.057477996328210e+02 \\ t_z^G = 1.531416816237290e+02 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} r_{11}^D = -0.07733514262240 \\ r_{12}^D = -0.02074146739449 \\ r_{13}^D = -0.98372638368125 \\ r_{21}^D = 0.16154101741209 \\ r_{22}^D = 0.97259664934943 \\ r_{23}^D = -0.04727718867043 \\ r_{31}^D = -1.00000000000000 \\ r_{32}^D = 0.15102256632406 \\ r_{33}^D = 0.06257036855085 \\ t_x^D = 8.191392271237748e+02 \\ t_y^D = 1.126011879737281e+02 \\ t_z^D = 1.980238691323546e+02 \end{array} \right.$$

Valeurs calculés à partir d'images acquises de taille 600X560.

C.3. Le système mobile ATRV2 et le banc stéréoscopique.



Repères utilisés

Passage du repère *Image gauche* au repère *R* du robot mobile.

$$\begin{pmatrix} su^G \\ su^G \\ s \end{pmatrix} = \begin{pmatrix} -2.8941854e+02 & 7.3174191e+01 & 7.5492448e+02 & -5.5598646e+05 \\ -2.7819066e+02 & 6.7915095e+02 & 4.2245125e+00 & -3.7235805e+03 \\ -1.0000000e+00 & 1.9200475e-01 & 6.3559176e-02 & 1.5314168e+02 \end{pmatrix} \begin{pmatrix} x^R \\ y^R \\ z^R \\ 1 \end{pmatrix}$$

Passage du repère *Image droite* au repère *R* du robot mobile.

$$\begin{pmatrix} su^D \\ su^D \\ s \end{pmatrix} = \begin{pmatrix} -2.8933454e+02 & 6.7380234e+01 & 7.4027454e+02 & -5.2991842e+05 \\ -2.9866671e+02 & 6.4347050e+02 & -3.6203301e+00 & 1.4592179e+05 \\ -1.0000000e+00 & 1.5102257e-01 & 6.2570369e-02 & 1.9802387e+02 \end{pmatrix} \begin{pmatrix} x^R \\ y^R \\ z^R \\ 1 \end{pmatrix}$$

C.4- Le calibrage du banc stéréoscopique : expérimentation.



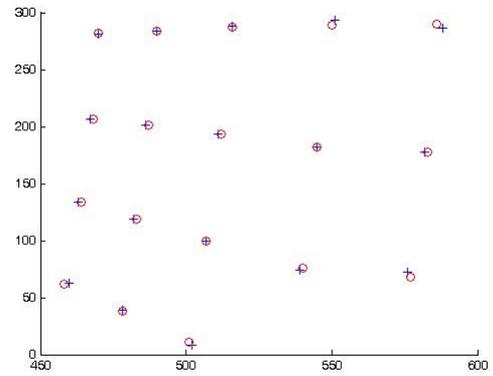
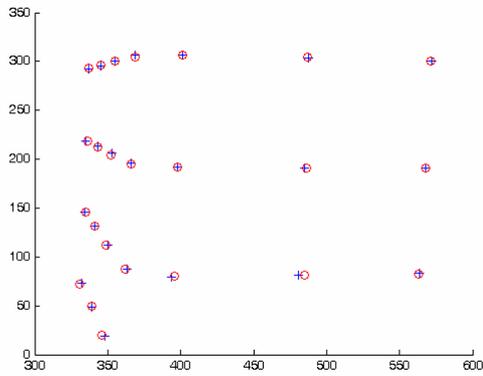
Scène de calibrage



(a)



(b)



(c)

Caméra Gauche (24 points)					Caméra Droite (18 points)				
X^R	Y^R	Z^R	u^G	v^G	X^R	Y^R	Z^R	u^D	v^D
530.00	30.00	830.00	253	401	530.00	-270.00	830.00	270	586
530.00	-20.00	830.00	256	487	530.00	-270.00	780.00	382	583
530.00	-70.00	830.00	260	571	530.00	-270.00	730.00	492	577
530.00	30.00	780.00	368	398	545.00	-255.00	830.00	271	550
530.00	-20.00	780.00	369	486	595.00	-255.00	830.00	273	516
530.00	-70.00	780.00	369	568	645.00	-255.00	830.00	276	490
530.00	30.00	730.00	480	396	695.00	-255.00	830.00	278	470
530.00	-20.00	730.00	479	485	545.00	-255.00	780.00	378	545
530.00	-70.00	730.00	478	563	595.00	-255.00	780.00	367	512
545.00	45.00	830.00	256	369	645.00	-255.00	780.00	359	487
595.00	45.00	830.00	260	355	695.00	-255.00	780.00	353	468
645.00	45.00	830.00	264	345	545.00	-255.00	730.00	484	540
695.00	45.00	830.00	267	337	595.00	-255.00	730.00	460	507
545.00	45.00	780.00	365	366	645.00	-255.00	730.00	441	483
595.00	45.00	780.00	356	352	695.00	-255.00	730.00	426	464
645.00	45.00	780.00	348	343	595.00	-255.00	680.00	549	501
695.00	45.00	780.00	342	336	645.00	-255.00	680.00	522	478
545.00	45.00	730.00	472	362	695.00	-255.00	680.00	498	458
595.00	45.00	730.00	448	349					
645.00	45.00	730.00	429	341					
695.00	45.00	730.00	414	335					
595.00	45.000	680.00	540	346					

645.00	45.00	680.00	510	339
695.00	45.00	680.00	488	331

(d)

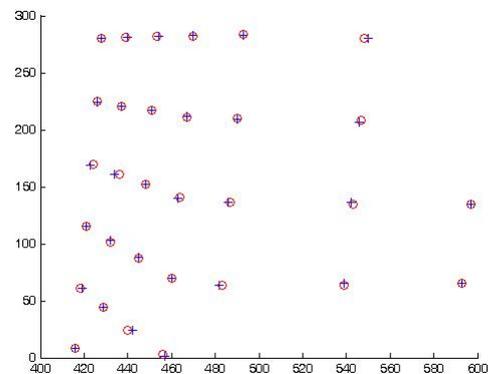
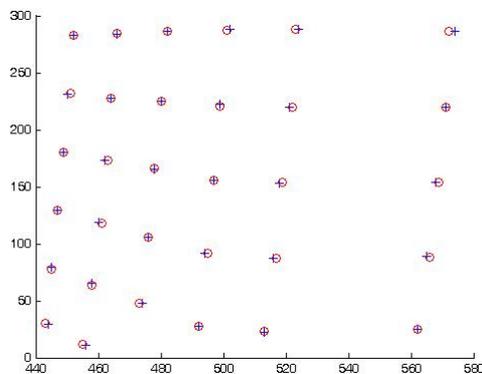
Paire stéréoscopique 1 de la mire de calibration : (a) Paires d'images acquises de la mire. (b) et (c) Validation 2D des résultats du calibrage (les cercles rouges désignent les points 2D extraits manuellement et les croix bleues désignent les points 2D calculés à partir de la matrice de projection perspective M). (d) Points extraits et utilisés lors du calibrage.



(a)



(b)



(c)

Caméra Gauche (32 points)					Caméra Droite (31 points)				
X^R	Y^R	Z^R	u^G	v^G	X^R	Y^R	Z^R	u^D	v^D
730.00	-120.00	830.00	272	523	695.00	-270.00	830.00	276	493
730.00	-170.00	830.00	274	572	695.00	-320.00	830.00	280	548
730.00	-120.00	780.00	340	522	695.00	-270.00	780.00	350	490
730.00	-170.00	780.00	340	571	695.00	-320.00	780.00	352	547
730.00	-120.00	730.00	406	519	695.00	-270.00	730.00	424	487
730.00	-170.00	730.00	406	569	695.00	-320.00	730.00	425	543
730.00	-120.00	680.00	473	517	695.00	-370.00	730.00	425	597
730.00	-170.00	680.00	472	566	695.00	-270.00	680.00	496	483
730.00	-120.00	630.00	537	513	695.00	-320.00	680.00	496	539
730.00	-170.00	630.00	535	562	695.00	-370.00	680.00	495	593
745.00	-105.00	830.00	273	501	710.00	-255.00	830.00	278	470
795.00	-105.00	830.00	274	482	760.00	-255.00	830.00	278	453

845.00	-105.00	830.00	275	466	810.00	-255.00	830.00	279	439
895.00	-105.00	830.00	277	452	860.00	-255.00	830.00	280	428
745.00	-105.00	780.00	339	499	710.00	-255.00	780.00	349	467
795.00	-105.00	780.00	335	480	760.00	-255.00	780.00	343	451
845.00	-105.00	780.00	332	464	810.00	-255.00	780.00	339	437
895.00	-105.00	780.00	328	451	860.00	-255.00	780.00	335	426
745.00	-105.00	730.00	404	497	710.00	-255.00	730.00	419	464
795.00	-105.00	730.00	394	478	760.00	-255.00	730.00	408	448
845.00	-105.00	730.00	387	463	810.00	-255.00	730.00	399	436
895.00	-105.00	730.00	380	449	860.00	-255.00	730.00	390	424
745.00	-105.00	680.00	468	495	710.00	-255.00	680.00	490	460
795.00	-105.00	680.00	454	476	760.00	-255.00	680.00	473	445
845.00	-105.00	680.00	442	461	810.00	-255.00	680.00	459	432
895.00	-105.00	680.00	431	447	860.00	-255.00	680.00	445	421
745.00	-105.00	630.00	532	492	710.00	-255.00	630.00	557	456
795.00	-105.00	630.00	512	473	760.00	-255.00	630.00	536	440
845.00	-105.00	630.00	496	458	810.00	-255.00	630.00	516	429
895.00	-105.00	630.00	482	445	860.00	-255.00	630.00	499	418
845.00	-105.00	580.00	548	455	860.00	-255.00	580.00	552	416
895.00	-105.00	580.00	530	443					

(d)

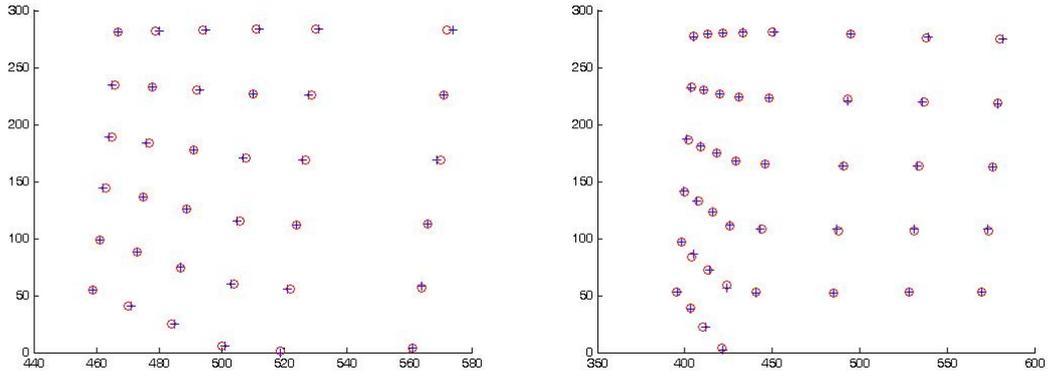
Paire stéréoscopique 2 de la mire de calibration : (a) Paires d'images acquises de la mire. (b) et (c) Validation 2D des résultats du calibration (les cercles rouges désignent les points 2D extraits manuellement et les croix bleues désignent les points 2D calculés à partir de la matrice de projection perspective M). (d) Points extraits et utilisés lors du calibration.



(a)



(b)



(c)

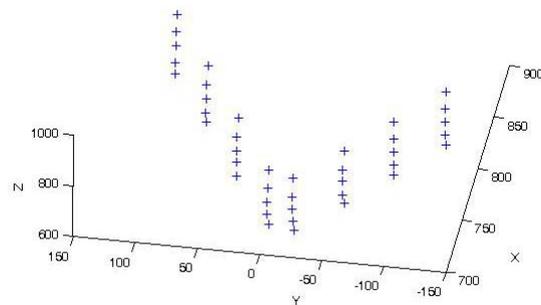
Caméra Gauche (36 points)					Caméra Droite (44 points)				
X^R	Y^R	Z^R	u^G	v^G	X^R	Y^R	Z^R	u^D	v^D
810.00	-165.00	830.00	276	530	830.00	-270.00	830.00	279	450
810.00	-215.00	830.00	277	572	830.00	-320.00	830.00	281	495
810.00	-165.00	780.00	334	529	830.00	-370.00	830.00	284	538
810.00	-215.00	780.00	334	571	830.00	-420.00	830.00	285	580
810.00	-165.00	730.00	391	527	830.00	-270.00	780.00	337	448
810.00	-215.00	730.00	391	570	830.00	-320.00	780.00	338	493
810.00	-165.00	680.00	448	524	830.00	-370.00	780.00	340	537
810.00	-215.00	680.00	447	566	830.00	-420.00	780.00	341	579
810.00	-165.00	630.00	504	522	830.00	-270.00	730.00	395	446
810.00	-215.00	630.00	503	564	830.00	-320.00	730.00	396	491
810.00	-165.00	580.00	559	519	830.00	-370.00	730.00	396	534
810.00	-215.00	580.00	556	561	830.00	-420.00	730.00	397	576
825.00	-150.00	830.00	276	511	830.00	-270.00	680.00	452	444
875.00	-150.00	830.00	277	494	830.00	-320.00	680.00	453	488
925.00	-150.00	830.00	278	479	830.00	-370.00	680.00	453	531
975.00	-150.00	830.00	279	467	830.00	-420.00	680.00	453	574
825.00	-150.00	780.00	333	510	830.00	-270.00	630.00	507	441
875.00	-150.00	780.00	330	492	830.00	-320.00	630.00	508	485
925.00	-150.00	780.00	327	478	830.00	-370.00	630.00	507	528
975.00	-150.00	780.00	325	466	830.00	-420.00	630.00	507	570
825.00	-150.00	730.00	389	508	845.00	-255.00	830.00	280	433
875.00	-150.00	730.00	382	491	895.00	-255.00	830.00	280	422
925.00	-150.00	730.00	376	477	945.00	-255.00	830.00	281	413
975.00	-150.00	730.00	371	465	995.00	-255.00	830.00	282	405
825.00	-150.00	680.00	445	506	845.00	-255.00	780.00	336	431
875.00	-150.00	680.00	434	489	895.00	-255.00	780.00	333	420
925.00	-150.00	680.00	424	475	945.00	-255.00	780.00	330	411
975.00	-150.00	680.00	416	463	995.00	-255.00	780.00	327	404
825.00	-150.00	630.00	500	504	845.00	-255.00	730.00	392	429
875.00	-150.00	630.00	486	487	895.00	-255.00	730.00	385	418
925.00	-150.00	630.00	472	473	945.00	-255.00	730.00	380	409
975.00	-150.00	630.00	461	461	995.00	-255.00	730.00	374	402
825.00	-150.00	580.00	554	500	845.00	-255.00	680.00	449	426
875.00	-150.00	580.00	535	484	895.00	-255.00	680.00	437	416
925.00	-150.00	580.00	519	470	945.00	-255.00	680.00	427	408
975.00	-150.00	580.00	505	459	995.00	-255.00	680.00	419	400
					845.00	-255.00	630.00	501	424
					895.00	-255.00	630.00	488	413
					945.00	-255.00	630.00	476	404
					995.00	-255.00	630.00	463	398
					845.00	-255.00	580.00	556	421
					895.00	-255.00	580.00	538	410
					945.00	-255.00	580.00	521	403
					995.00	-255.00	580.00	507	395

(d)

Paire stéréoscopique 3 de la mire de calibration : (a) Paires d'images acquises de la mire. (b) et (c) Validation 2D des résultats du calibrage (les cercles rouges désignent les points 2D extraits manuellement et les croix bleus désignent les points 2D calculés à partir de la matrice de projection perspective M). (d) Points extraits et utilisés lors du calibrage.



(a)



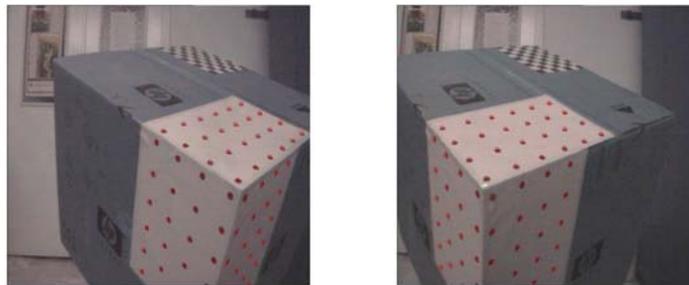
(b)

Caméra Gauche (40 points)		Caméra Droite (40 points)	
u^G	v^G	u^D	v^D
255	392	250	157
255	343	251	126
256	300	251	99
257	263	252	76
330	391	326	154
326	342	321	123
323	299	317	96
319	262	313	74
405	389	401	152
396	340	391	121
388	298	381	95
381	261	373	73
478	387	474	151
465	338	459	120
453	297	445	94
442	260	433	71
549	385	546	151
532	337	526	119
516	295	508	94
501	259	492	71

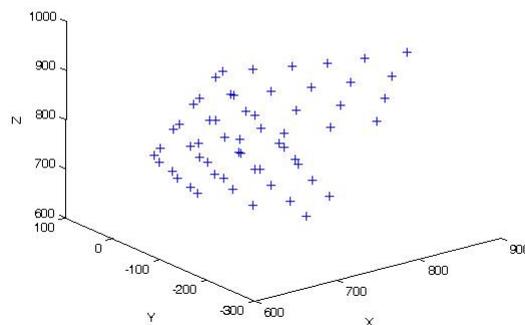
254	418	250	183
258	446	255	230
259	469	258	270
262	490	262	306
330	418	327	181
327	446	325	228
324	469	325	269
322	490	324	305
404	415	402	179
395	443	397	226
387	467	390	267
382	488	386	303
477	413	476	178
463	441	465	224
451	465	455	265
440	486	446	301
546	410	547	177
528	438	532	223
512	462	518	263
497	483	505	299

(c)

Reconstruction 3D de la paire stéréoscopique 4 de la mire de calibrage. (a) Paires d'images acquises de la mire, les cercles rouges désignent les points 2D extraits manuellement. (b) Reconstruction 3D dans le repère du robot mobile *ATRV2* des points extraits. (c) Points extraits et utilisés lors de la reconstruction.



(a)



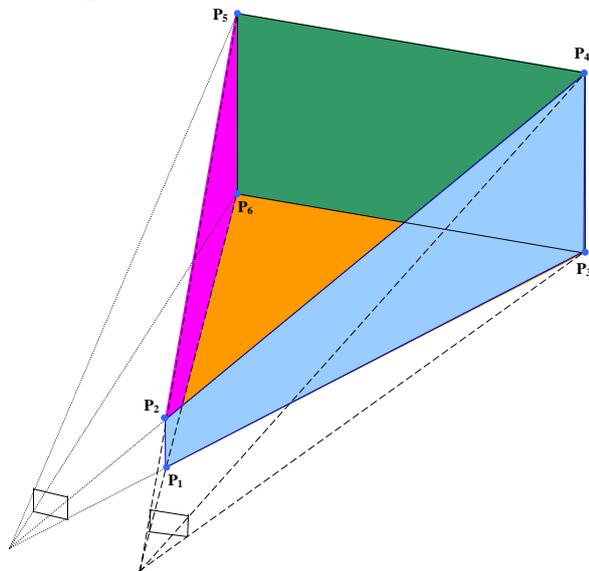
(b)

Caméra Gauche (58 points)		Caméra Droite (58 points)	
u^G	v^G	u^D	v^D
380	442	375	159
339	386	334	125
305	338	299	97
274	296	268	74
448	424	445	158
407	373	401	126
370	328	363	99
336	289	329	77
509	409	506	158
467	361	460	127
427	320	420	101
392	282	385	79
519	351	514	128
480	312	472	103
444	277	436	82
526	305	519	106
489	271	481	85
531	267	522	88
384	472	382	189
357	505	357	248
334	533	335	298
314	557	317	343
453	454	452	188
422	488	424	243
396	516	400	291
373	540	378	334
511	436	512	186
479	470	483	238
451	499	457	284
425	524	433	325
530	454	535	234
500	482	508	277
473	508	482	317
545	468	553	271
349	457	345	177
323	492	321	236
303	521	303	287
286	545	286	332
271	566	272	371
257	584	258	405
313	402	308	143
293	439	289	199
276	470	274	249
262	497	261	293
249	521	248	333

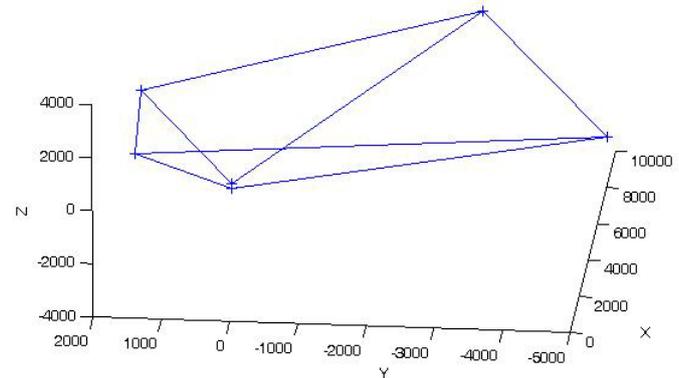
236	541	236	367
281	353	275	114
265	392	260	168
251	425	248	216
239	453	237	259
227	479	226	298
217	501	215	332
252	310	246	90
238	350	234	141
228	383	225	188
218	413	215	229
208	440	205	267
198	464	196	302

(c)

Reconstruction 3D de la paire stéréoscopique 5 de la mire de calibration. (a) Paires d'images acquises de la mire, les cercles rouges désignent les points 2D extraits manuellement. (b) Reconstruction 3D dans le repère du robot mobile ATRV2 des points extraits. (c) Points extraits et utilisés lors de la reconstruction.



(a)



(b)

	X^R	Y^R	Z^R
P ₁	422	-27	739
P ₂	430	-22	895
P ₃	8 711	-4 972	-2 589
P ₄	6 479	-3 315	3 546
P ₅	3 751	1 528	2 034
P ₆	3 986	1 631	-555

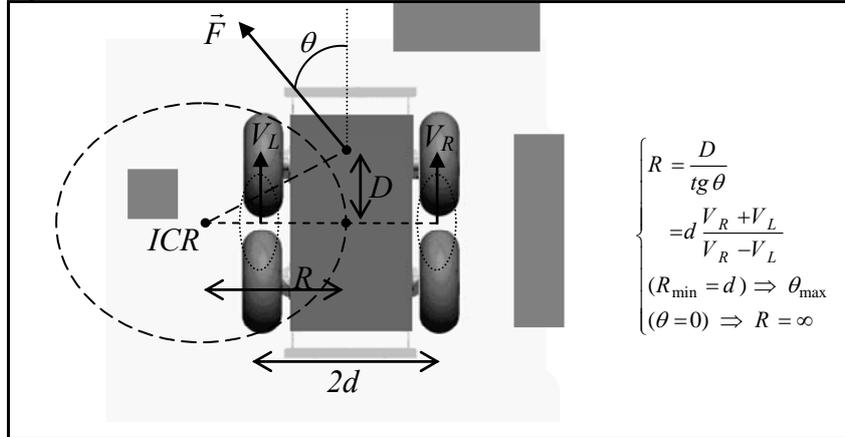
(c)

Champ de vision commun des deux caméras du banc stéréoscopique. (a) Le champ théorique. (b) Le champ calculé après calibration. (c) Points formant le champ de vision.

D. Commande du mouvement.

D.1. Déplacement du système mobile ATRV2 dans son environnement.

- ◆ Calcul du rayon de courbure ICR .



- ◆ Calcul des valeurs de v et w :

$$v = \frac{V_R + V_L}{2} = \begin{cases} \frac{V_R}{2} \left(1 + \frac{R-d}{R+d} \right) & \left\{ \begin{array}{l} d \leq R < +\infty \\ \frac{V_R}{2} \leq v \leq V_R \end{array} \right. \\ 0 & \left\{ \begin{array}{l} 0 \leq R < d \\ \frac{V_R}{2} \leq v \leq V_R \end{array} \right. \\ \frac{V_L}{2} \left(1 + \frac{R+d}{R-d} \right) & \left\{ \begin{array}{l} -d \geq R > -\infty \\ \frac{V_L}{2} \leq v \leq V_L \end{array} \right. \end{cases}$$

$$\omega = \frac{V_R - V_L}{2d} = \begin{cases} \frac{V_R}{2d} \left(1 - \frac{R-d}{R+d} \right) & \left\{ \begin{array}{l} d \leq R < +\infty \\ \frac{V_R}{2d} \leq \omega \leq 0 \end{array} \right. \\ \frac{V_L}{2d} \left(\frac{R+d}{R-d} - 1 \right) & \left\{ \begin{array}{l} -d \geq R > -\infty \\ -\frac{V_L}{2d} \leq \omega \leq 0 \end{array} \right. \end{cases}$$

E. Algorithme de planification de chemin optimal basé sur l'algorithme D^* .

Les fonctions $Init_D^*$ et $Case_D^*$:

Nous avons présenté les deux fonctions de même façon que celles utilisées dans [Ste94] pour montrer leurs différences et similitudes.

Function : $Init_D^*$.

```

L1 Do {
L2    $X = MinState()$ .
L3   if ( $X == Null$ ) break.
L4    $k_{old} = GetKmin()$ .
L5   Delete( $X$ ).
L6    $wc = WorkableCase(X)$ .
L7   if ( $wc$ ) {
L8     if ( $k_{old} < h(X)$ )
L9       ProcessI( $X, k_{old}$ ).
L10    else if ( $k_{old} == h(X)$ )
L11      ProcessII( $X, k_{old}$ ).
L12    else ProcessIII( $X, k_{old}$ ).
L13  }
L14   $k_p = GetKmin()$ .
L15 }while( $k_p != -1$ ).
```

bool $WorkableCase(X)$.

```

L1  $wc = true$ .
L2 for each neighbor  $Y$  of  $X$  :
L3   if ( $Y$  is not walkable or
L4      $Y$ 's CV  $> 0$ )  $wc = false$ .
L5 return( $wc$ ).
```

Function : $ProcessII(X, k_{old})$.

```

L1 for each neighbor  $Y$  of  $X$  :
L2   if ( $(t(Y) = New)$  or
L3     ( $b(Y) == X$ ) and ( $h(Y) != (h(X) + c(X, Y))$ ) or
L4     ( $b(Y) != X$ ) and ( $h(Y) > (h(X) + c(X, Y))$ )) {
L5      $b(Y) = X$ .
L6     Insert( $Y, h(X) + c(X, Y)$ ).
L7   }
```

Function : $Case_D^*(X_{Rob})$.

```

L1 The tag value of all states is set to New.
L2 The backpointer of all states is set to  $-1$ .
L3 The goal  $G$  in the OpenList.
L4 Do {
L5    $X = MinState()$ .
L6   if ( $(X == Null)$  or ( $X == X_{Rob}$ )) break.
L7    $k_{old} = GetKmin()$ .
L8   Delete( $X$ ).
L9    $wc = WorkableCase(X)$ .
L10  if ( $wc$ ) {
L11   if ( $k_{old} < h(X)$ )
L12     ProcessI( $X, k_{old}$ ).
L13   else if ( $k_{old} == h(X)$ )
L14     ProcessII( $X, k_{old}$ ).
L15   else ProcessIII( $X, k_{old}$ ).
L16  }
L17   $k_p = GetKmin()$ .
L18 }while( $k_p != -1$ ).
```

Function : $ProcessI(X, k_{old})$.

```

L1 for each neighbor  $Y$  of  $X$  :
L2   if ( $(h(Y) \leq k_{old})$  and ( $h(X) > (h(X) + c(Y, X))$ )) {
L3      $b(X) = Y$ ;
L4      $h(X) = h(Y) + c(Y, X)$ .
L5   }
```

Function : $ProcessIII(X, k_{old})$.

```

L1 for each neighbor  $Y$  of  $X$  :
L2   if ( $(t(Y) = New)$  or
L3     ( $b(Y) == X$ ) and ( $h(Y) != (h(X) + c(X, Y))$ )) {
L4      $b(Y) = X$ .
L5     Insert( $Y, h(X) + c(X, Y)$ ).
L6   }
L7   else if ( $(b(Y) != X)$  and ( $h(Y) > (h(X) + c(X, Y))$ ))
L8     Insert( $X, h(X)$ ).
L9   else if ( $(b(Y) != X)$  and ( $h(X) > (h(Y) + c(Y, X))$ )
L10    and ( $t(Y) = Closed$ ) and ( $h(Y) > k_{old}$ )) {
L11     Insert( $Y, h(Y)$ ).
L12   }
```

✓ La fonction $MinState$, renvoie la cellule dans $OpenList$ dont la valeur de k est minimale, et elle renvoie $NULL$ si la liste est vide.

- ✓ La fonction *GetKmin*, retourne la valeur k_{min} dans *Openlist*, la valeur -1 est retournée si la liste est vide.
- ✓ La fonction *Delete(X)* supprime la cellule X du *OpenList* et initialise $t(X)=Closed$.
- ✓ La fonction *Insert(X, h_{new})*, calcule :
 - $k(X)=h_{new}$ si $t(X)=New$,
 - $k(X)=\min(k(X), h_{new})$ si $t(X)=Open$,
 - $k(X)=\min(h(X), h_{new})$ si $t(X)=Closed$,prend $h(X)=h_{new}$ et $t(X)=Open$, et place ou repositionne suivant la valeur de k la cellule X dans *OpenList*.