



AVERTISSEMENT

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

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

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

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

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

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

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

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

BIBLIOTHEQUE UNIVERSITAIRE - METZ	
N° inv.	20040505
Cote	S/M304/17
Loc	

THESE

Perception Multisensorielle Pour le Positionnement d'un Véhicule Autonome dédié aux Personnes Handicapés Moteurs

Présentée dans le cadre de l'école doctorale IAEM Lorraine pour l'obtention
du grade de Docteur de l'Université de Metz
mention Automatique

MICHEL KREUTNER

Soutenue publiquement le 21 septembre 2004,
devant la commission d'examen :

Rapporteurs :

M. Etienne COLLE
M. Michel DEVY

Examineurs :

Mme. Edwige PISSALOUX
M. Guy BOURHIS
Mme. Odile HORN

BIBLIOTHEQUE UNIVERSITAIRE DE METZ



031 536054 3

Table des matières

Introduction	17
Organisation du mémoire	18
1 Capteurs et méthodes pour la localisation	21
1.1 Les capteurs en robotique mobile	21
1.1.1 Les capteurs proprioceptifs	22
1.1.1.1 Codeurs optiques	22
(a) Codeurs optiques incrémentaux	22
(b) Codeurs optiques absolus	23
1.1.1.2 Capteurs inertiels	23
(a) Accéléromètre	24
(b) Gyroscope	24
1.1.2 Capteurs extéroceptifs	25
1.1.2.1 Télémètres à ultrasons	25
(a) Physique de base d'un capteur	25
(b) Principes de la mesure de distance par sonar	26
(c) Avantages et inconvénients du capteur à ultrasons	27
(d) Modèles des capteurs à ultrasons	28
1.1.2.2 Lasers	29
(a) Techniques de mesures	29
(a).1 Technique par mesure du temps de vol	29
(a).2 Mesure de la différence de phase	30
(b) Lasers à balayage	30
(c) Avantages et inconvénient des lasers	30
1.1.2.3 Capteurs infrarouges	30
1.1.2.4 Capteur de vision	32
(a) Modèle trou d'épingle	32
(b) Système de vision monoculaire	34
(c) Système de stéréovision	34
(d) Différents types de systèmes de vision omnidirectionnelle	34
1.2 Représentation de l'environnement et interprétation des mesures	35
1.2.1 Modèles géométriques	35
1.2.2 Grilles d'occupation	36
1.2.3 Cartes topologique et vecteurs d'informations	36
1.3 Méthodes de calcul de la position	37
1.3.1 Méthodes d'intégration	37
1.3.1.1 Modèles odométriques	37
(a) Plate-forme à mode différentiel	38
(b) Plate-forme à mode synchrone	38
1.3.1.2 Conclusion sur l'odomètre	39
1.3.2 Méthodes de classification	39
1.3.3 Méthodes fondées sur le filtre de Kalman	42

1.3.3.1	Principe	42
1.3.3.2	Exemples d'application du filtre de Kalman	43
1.3.4	Méthodes fondées sur les moindres carrés	45
1.3.4.1	Principe	46
1.3.4.2	Exemples d'application des moindres carrés	46
1.3.5	Méthodes probabilistes	48
1.3.6	Autres méthodes	50
1.3.7	Conclusion	51
1.4	Robotique mobile pour personnes handicapées : méthodes de localisation	51
1.4.1	Introduction	51
1.4.2	Projets	53
1.4.2.1	Le projet "A.R.P.H." – Assistance Robotique aux Personnes Handicapées	53
(a)	Capteurs de la plate-forme	53
(b)	Localisation de la base mobile	53
(c)	Coopération Homme Machine	53
1.4.2.2	Le projet "NavChair"	54
(a)	Prototype du NavChair	54
(b)	Module d'assistance à la navigation	54
1.4.2.3	Le projet "RobChair"	55
(a)	Principe du contrôle réactif partagé à l'aide d'une IHM vocale	55
(b)	Le fauteuil et les capteurs	55
(c)	Interface Homme Machine Vocale	55
(d)	Module de navigation	56
1.4.2.4	Le projet "INRO"	56
(a)	Objectifs du projet INRO	56
(b)	Le système	56
(c)	Utilisation des capteurs	57
(c).1	Système à ultrasons	57
(c).2	Marqueur laser	57
(c).3	GPS	57
(c).4	Fusion des données	57
1.4.2.5	Le projet "Rolland"	58
1.4.2.6	Le projet "OMNI"	58
(a)	La base mobile et le système de perception	58
(b)	L'interface Homme-Machine et l'assistance proposée	59
1.4.3	Conclusion sur la robotique mobile appliquée à l'assistance aux personnes handicapées	60
1.5	Conclusion	60
2	Méthode de localisation dynamique du VAHM	63
2.1	Introduction	63
2.2	Le projet VAHM	63
2.2.1	Architecture matérielle	63
2.2.2	Architecture logicielle	65
2.3	Evaluation de l'erreur odométrique	66
2.4	Contexte et principe	67
2.4.1	Contexte	67
2.4.2	Principe de la méthode	67
2.5	Modèle de l'environnement et des capteurs à ultrasons	68
2.5.1	Modèle global de l'environnement	68
2.5.2	Construction du modèle local	69
2.5.2.1	Modèle d'un capteur à ultrasons	69
2.5.2.2	Création de la carte locale	70
2.6	Méthode de correction de la position odométrique	72

2.6.1	Mise en correspondance des modèles	72
2.6.1.1	Affectation des points d'impact du modèle local	72
2.6.1.2	Calcul de la distance entre le point et le segment	73
2.6.1.3	Estimation de l'angle d'incidence	74
(a)	Présentation du problème	74
(b)	Mise en œuvre	75
2.6.1.4	Illustration d'une mise en correspondance	75
2.6.1.5	Problème lié à la diaphonie	75
(a)	Présentation du problème	75
(b)	Solution et mise en œuvre	76
2.6.2	Calcul de la transformation	78
2.7	Mise en œuvre en simulation	80
2.7.1	Contexte	80
2.7.2	Exemple de simulation dans un environnement simple	80
2.8	Mise en œuvre sur le prototype	81
2.8.1	Conditions d'évaluations	81
2.8.2	Mode de présentation des résultats	82
2.8.2.1	Représentation des erreurs de localisation en fonction de la position odométrique	82
2.8.2.2	Représentation des erreurs de localisation en fonction de l'orientation odométrique	82
2.8.2.3	Histogrammes des valeurs corrigées	84
2.8.3	Présentation de quelques résultats	84
2.8.3.1	Etude de la position 00	84
2.8.3.2	Etude de la position 03	85
2.8.3.3	Etude de la position 05	87
2.8.3.4	Etude de la position 09	88
2.8.3.5	Etude de la position 15	91
2.8.4	Résumé des résultats obtenus pour le reste des essais	91
2.9	Conclusion sur la méthode	92
3	Ajustement de l'orientation par vision monoculaire	95
3.1	Introduction	95
3.2	Principe de la méthode	95
3.3	Modèles utilisés par la méthode d'ajustement	96
3.3.1	Modélisation de l'environnement	96
3.3.2	Informations extraites de l'image	96
3.4	Mise en œuvre de l'extraction des contours de l'image	97
3.4.1	Origine des contours	97
3.4.2	Amélioration du contraste de l'image	98
3.4.3	Détection des contours dans l'image	100
3.4.3.1	Calcul du gradient	101
(a)	Résumé des équations de filtrage	101
(b)	Principe de calcul du gradient	102
3.4.3.2	Suppression des non maxima locaux	102
3.4.3.3	Seuillage par hystérésis	103
3.4.3.4	Suivi de contour	103
3.4.3.5	Mise en œuvre	104
3.4.4	Paramètres du capteur de vision	104
3.4.4.1	Présentation des paramètres utilisés	104
3.5	Détermination de l'orientation du robot à partir de l'observation d'un unique segment	106
3.5.1	Calcul de l'orientation	106
3.5.2	Ajustement de l'orientation	107

3.6	Ajustement en orientation à partir de plusieurs observations	108
3.6.1	L'arbre d'interprétation	109
3.6.1.1	Présentation de l'arbre	109
3.6.1.2	Construction de l'arbre	110
3.6.1.3	Interprétation de l'arbre	111
3.6.1.4	Limitation de l'espace des segments	112
3.6.1.5	Elagage de l'arbre	113
3.7	Etude de la complexité de l'algorithme	114
3.7.1	Nombre de nœuds d'un arbre total	114
3.7.2	Nombre de nœuds de l'arbre élagué	115
3.7.3	Comparaison et conclusion	115
3.8	Influence d'une erreur en position	116
3.9	Etude en simulation	118
3.9.1	Objectif et contexte de la simulation	118
3.9.2	Etude qualitative de la méthode d'ajustement	119
3.9.3	Synthèse	120
3.9.4	Estimation de la plage d'erreur admissible sur l'orientation estimée	121
3.9.5	Evaluation de l'influence d'une erreur en position	123
3.9.6	Synthèse	125
3.10	Ajustement à partir d'images réelles	125
3.10.1	Etudes des résultats obtenus pour différentes configurations	126
3.10.2	Estimation de la plage d'erreur admissible	127
3.10.3	Influence d'une erreur en position	129
3.10.4	Résumé	130
3.11	L'arbre d'interprétation à nœuds nuls	130
3.11.1	Problème lié à la structure de l'arbre	130
3.11.2	Solution proposée	130
3.11.3	Evaluation avec des images réelles	131
3.11.3.1	Evaluation des ajustements pour des erreurs initiales nulles	131
3.11.3.2	Evaluation de la plage d'erreur initiale admissible	132
3.11.3.3	Influence d'une erreur en position sur l'ajustement	132
3.11.4	Résumé	134
3.12	Conclusion sur l'ajustement en orientation	134
Conclusion et Perspectives		135
	Conclusion	135
	Perspectives	136
A	Algorithmes utilisés par la localisation dynamique	139
A.1	Version utilisant la distance point segment	139
A.2	Version utilisant la distance point segment et l'angle d'incidence	139
A.3	Version éliminant les appariements incohérents	140
B	Intersection de deux segments	143
B.1	Position d'un point par rapport à une droite	143
B.2	Intersection de deux segments	143
C	Méthode de Gréville	145
C.1	Présentation	145
C.2	Mise en œuvre	145
D	Détermination des coefficients du filtre de Deriche	147
D.1	Filtre de lissage	147
D.2	Filtre de dérivation	149

E Résultats d'un traitement d'image	153
F Résultats de l'influence d'une erreur en conditions réelles	155
Bibliographie	161

Cette page est laissée blanche intentionnellement

Table des figures

1	Le système "Car Parking Aid".	17
2	Le robot tondeuse Automower de chez Electrolux.	17
1.1	Principe d'un codeur incrémental : (a) simple canal et (b) double canaux.	22
1.2	Exemple du disque d'un codeur absolu utilisant un code de gray sur 8 bits.	23
1.3	Principe de fonctionnement d'un accéléromètre.	24
1.4	Configuration d'un gyroscope mécanique à deux axes [Eve95].	24
1.5	Exemples de configuration de ceintures de capteurs à ultrasons.	25
1.6	Cône d'émission de l'énergie acoustique.	26
1.7	Principe de la fusion des mesures provenant d'une même cible, d'après [Wij98].	27
1.8	Principe de la diaphonie	27
1.9	Modèles de capteur à ultrasons	28
1.10	Modèle de capteur à ultrasons utilisé dans [Mor85].	29
1.11	Représentation de l'espace devant le capteur à ultrasons	29
1.12	Paramètres pris en compte pour déterminer la confiance de l'occupation de la cellule dans [Lim92].	29
1.13	Paramètres utiles à la conception d'un capteur infrarouge.	31
1.14	Géométrie du modèle de caméra sténopé	32
1.15	Vue de côté de la projection au travers d'un modèle de caméra sténopé	32
1.16	Schéma de principe d'une caméra avec lentille sphérique.	34
1.17	Schéma de principe d'une caméra avec miroir conique.	34
1.18	Schéma de principe d'une caméra avec miroir à double courbure.	34
1.19	Exemple de Régions de Profondeur Constance (RCD) d'après [Leo91].	35
1.20	Exemple de grille d'occupation bayésienne extrait [Wij01].	36
1.21	Exemple de carte topologique extraite de [Alt03]	37
1.22	Plateforme à conduite différentielle.	38
1.23	Configuration d'une plate-forme à conduite synchrone.	39
1.24	Configuration d'une roue d'une plate-forme à conduite synchrone.	39
1.25	Exemple d'échec d'appariement par histogramme des couleurs dans [Zho03].	40
1.26	Résultats obtenus avec la méthode des histogrammes multi-dimensionnels dans [Zho03].	41
1.27	Informations utilisées dans [Lee00]	41
1.28	Modélisation d'un amer d'après [Hay01]	42
1.29	Exemple de correction extrait de [Arr99].	44
1.30	Principe de fonctionnement du système de localisation présenté dans [Nei99].	45
1.31	Les trois zones définies pour un segment.	47
1.32	Représentation des paramètres des caractéristiques dans le plan 2D, d'après [Bor02b]	48
1.33	Principe du système de localisation combinant une localisation Markovienne et un filtre de Kalman, d'après [Gut02b]	49
1.34	Modélisation de l'environnement dans [Lan03].	50
1.35	Système décrit dans [Her03].	51
1.36	Prototype du robot ARPH	53
1.37	Prototype du projet NavChair.	54
1.38	Vue d'ensemble du système de commande de RobChair.	55

1.39	Le prototype de RobChair avec ses équipements	56
1.40	Système du fauteuil INRO.	57
1.41	Marqueur Laser installé sur le prototype INRO.	58
1.42	Prototype du fauteuil Rolland	58
1.43	Prototype du fauteuil OMNI.	59
1.44	Répartition de la charge de travail entre l'utilisateur et le système.	59
2.1	Prototype du VAHM	64
2.2	Architecture du Bus DX	64
2.3	Positionnement des capteurs à ultrasons tout autour du prototype du VAHM.	64
2.4	Grille globale construite à partir d'un plan d'occupation de l'environnement.	66
2.5	Grille locale construite autour du robot à partir des données des capteurs à ultrasons.	66
2.6	Test de l'odomètre du prototype du VAHM	67
2.7	Principe de la méthode de correction de la localisation par capteurs à ultrasons.	68
2.8	Modélisation des frontières de l'environnement.	69
2.9	Modélisation d'une mesure de distance fournie par un capteur à ultrasons.	70
2.10	Représentation des variables utilisées pour la projection des mesures des capteurs à ultrasons dans le référentiel global - projection de la carte locale dans la carte globale.	71
2.11	Exemple d'une carte locale après acquisition de 16 capteurs à ultrasons.	71
2.12	Algorithme d'appariement des points de la carte locale avec les segments de la carte globale.	73
2.13	Exemples d'appariements multiples possibles.	73
2.14	Incidence inférieure à l'incidence limite de l'onde acoustique sur une surface.	74
2.15	Incidence supérieure à l'incidence limite de l'onde acoustique sur une surface.	74
2.16	Exemple de projection d'une carte locale dans une carte globale.	75
2.17	Problème des réflexions multiples.	76
2.18	Exemple réel de réflexions multiples provoquant une ambiguïté d'appariement.	76
2.19	Organigramme de la version complète de l'algorithme d'appariement des point du modèle local avec les segments du modèle global.	77
2.20	Exemples d'appariements ambigus.	78
2.21	Représentation des Transformations appliquées au point P	78
2.22	Position réelle du robot dans l'environnement - Position (250, 170, 0)	80
2.23	Position odométrique du robot dans l'environnement - Position (260, 160, 5)	80
2.24	Position corrigée du robot dans l'environnement - Position (249, 171, 2)	80
2.25	Environnement utilisé pour les essais de la méthode de correction dynamique avec les différentes positions du véhicule.	81
2.26	Région d'initialisation odométrique.	82
2.27	Résultats de correction en fonction de la localisation odométrique.	83
2.28	Histogrammes des résultats de la correction pour une localisation réelle.	83
2.29	Position réelle 00 (359, 87, -2) : Histogrammes des résultats de la correction.	84
2.30	Position réelle 00 (359, 87, -2) : Résultats de correction en fonction de la localisation odométrique.	85
2.31	Position réelle 03 (279, 196, 63) : Histogrammes des résultats de la correction.	85
2.32	Position réelle 03 (279, 196, 63) : Résultats de correction en fonction de la localisation odométrique.	86
2.33	Position réelle 05 (200, 221, 87) : Histogrammes des résultats de la correction.	87
2.34	Position réelle 05 (200, 221, 87) : Résultats de correction en fonction de la localisation odométrique.	88
2.35	Position réelle 05 (200, 221, 87) : Résultats de correction en fonction de la localisation odométrique.	89
2.36	Position réelle 09 (168, 267, 180).	90
2.37	Position réelle 09 (168, 267, 180) : Résultats de correction en fonction de la localisation odométrique.	90
2.38	Position réelle 15 (302, 137, 268) : Histogrammes des résultats de la correction.	93

2.39	Position réelle 15 (302, 137, 268) : Résultats de correction en fonction de la localisation odométrique.	93
3.1	Principe de la méthode d'ajustement de l'orientation par vision monoculaire.	95
3.2	Modèle d'un segment vertical de l'environnement.	96
3.3	Informations extraites de l'Image.	97
3.4	Exemple de détections de contours dans une image	97
3.5	Les origines des contours dans une image.	98
3.6	Egalisation d'histogramme.	99
3.7	Schéma de principe du calcul de la norme et de la direction du gradient de l'image.	100
3.8	Profil d'un contour idéal.	101
3.9	Profil utilisé pour le contour.	101
3.10	Schéma de principe de mise en œuvre du calcul des composantes en i et en j du gradient.	102
3.11	Interpolation du gradient en un point du voisinage du point central.	103
3.12	Principe du seuillage par hystérésis.	103
3.13	Repères et paramètres utilisés pour définir la caméra	104
3.14	Repère du plan image et paramètres utilisés pour définir l'image	104
3.15	Observation de la mire de calibration.	105
3.16	Vue de dessus du système de calibration.	105
3.17	Calcul de l'orientation de la caméra à partir d'une observation d'une seule verticale du modèle.	107
3.18	Ajustement de l'orientation estimée de la caméra.	108
3.19	Orientation estimée de la caméra après ajustement.	109
3.20	Exemple d'arbre pour 4 contours (C_1 à C_4) et 6 verticales (V_1 à V_6).	110
3.21	Procédure : Creer_Noeuds_Enfants.	110
3.22	Exemple où l'ajustement τ est identique pour différents appariements.	111
3.23	Exemple de sélection de verticales du modèle – définition de la zone de visibilité en fonction des paramètres connus de la caméra, de sa position et de son orientation estimées dans le repère global.	112
3.24	Exemple d'incohérence d'un appariement.	113
3.25	Principe de la recherche de la cohérence de deux appariements successifs d'un chemin.	113
3.26	Résultat sur la création de l'arbre.	114
3.27	Evolution du nombre de nœuds pour 5 contours dans l'image avec un arbre total et un arbre élagué.	116
3.28	Evolution de l'erreur en fonction de l'erreur en position pour différentes positions relatives caméra/segment vertical.	118
3.29	Positions du robot dans l'environnement utilisé pour l'étude en simulation de la méthode d'ajustement.	119
3.30	Simulation 00.	120
3.31	Simulation 01.	120
3.32	Simulation 02.	121
3.33	Simulation 03.	121
3.34	Simulation 03 : Limite de la plage de variation de l'erreur initiale en orientation.	122
3.35	Simulation 00 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.	123
3.36	Simulation 01 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.	123
3.37	Simulation 02 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.	124
3.38	Simulation 03 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.	124
3.39	Position des segments verticaux et trajectoire du fauteuil dans l'environnement de test.	125
3.40	Résultat de ajustement pour le cas 05 en conditions réelles.	127
3.41	Vue de dessus de la position 11 du véhicule avant ajustement de l'orientation.	128
3.42	Image acquise par la caméra avec mise en évidence des contours.	128
3.43	Position 11 du véhicule pour une erreur en orientation de $+18^\circ$	128
3.44	Essai 00 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	130

3.45	Essai 01 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée avec l'utilisation d'un nœud nul.	133
3.46	Essai 03 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée avec l'utilisation d'un nœud nul.	133
3.47	Essai 07 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée avec l'utilisation d'un nœud nul.	133
A.1	Algorithme d'appariement avec test de la distance point/segment	139
A.2	Algorithme d'appariement avec test de la distance point/segment et test de l'angle d'incidence	140
A.3	Algorithme d'appariement avec test de la distance point/segment et test de l'angle d'incidence et prise en compte des réflexions multiples	141
B.1	Position d'un point par rapport à une droite.	143
B.2	Intersection de deux segments.	144
E.1	Image fournie par la caméra.	153
E.2	Image égalisée.	153
E.3	Image de la norme du gradient en lut inverse.	153
E.4	Image de la direction du gradient.	153
E.5	Image du gradient avec suppression des non maxima locaux en lut inverse.	154
E.6	Image issue du chaînage de contour.	154
E.7	Représentation des contours détectés.	154
E.8	Contours détectés en surimpression de l'image originale.	154
E.9	Schéma du traitement de l'image fournie par la caméra.	154
F.1	Essai 00 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	155
F.2	Essai 01 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	156
F.3	Essai 02 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	156
F.4	Essai 03 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	156
F.5	Essai 04 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	157
F.6	Essai 05 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	157
F.7	Essai 06 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	157
F.8	Essai 07 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	158
F.9	Essai 09 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	158
F.10	Essai 11 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	158
F.11	Essai 13 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	159
F.12	Essai 14 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	159
F.13	Essai 15 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.	159

Liste des tableaux

1.1	Table des états d'un codeur incrémental à double canaux.	22
2.1	Valeur moyenne et écart-type sur la localisation estimée pour trois distances (50 essais par distances).	67
2.2	Valeur moyenne et écart-type de l'erreur de localisation pour les essais réels.	91
2.3	Pourcentage de réussite de correction de chaque paramètre définissant la localisation du véhicule.	91
3.1	Nombre de nœuds de l'arbre total en fonction du nombre m de contours et du nombre n de verticales	115
3.2	Nombre de nœuds de l'arbre élagué en fonction du nombre m de contours et du nombre n de verticales	116
3.3	Localisations réelles et estimées utilisées pour l'évaluation en simulation de l'ajustement de l'orientation.	119
3.4	Evaluation de la plage de variation de l'erreur admissible sur l'orientation estimée avant l'ajustement par vision monoculaire.	122
3.5	Résultats des ajustements réalisés.	126
3.6	Evaluation de la plage de variation de l'erreur admissible sur l'orientation estimée avant l'ajustement par vision monoculaire en conditions réelles.	129
3.7	Résultats des ajustements réalisés avec l'utilisation d'un nœud nul.	131
3.8	Evaluation de la plage de variation de l'erreur d'orientation avant correction - arbre élagué et utilisation de nœuds nuls.	132

Cette page est laissée blanche intentionnellement

Ce mémoire de thèse résume les travaux que j'ai effectués au Laboratoire d'Automatique des Systèmes Coopératifs (LASC) de l'Université de METZ.

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

Je remercie Monsieur Guy BOURHIS, Maître de Conférences à l'Université de METZ, Habilité à diriger des Recherches, mon directeur de thèse.

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

Je tiens à remercier Madame Edwige PISSALOUX, Professeur à l'Université de Rouen, Monsieur Etienne COLLE, Professeur à l'Université d'Evry-Val-d'Essonne, ainsi que Monsieur Michel DEVY Directeur de Recherche CNRS au LAAS de Toulouse, d'avoir accepté d'examiner mon mémoire.

Mes remerciements s'adressent également à l'ensemble du personnel du laboratoire pour leurs conseils, leur disponibilité et leur amitié.

Cette page est laissée blanche intentionnellement

Introduction

Les avancées récentes en robotique mobile ont permis l'utilisation de véhicules sans homme à bord dans de nombreuses applications industrielles ou scientifiques en particulier en milieux hostiles : navigation sous-marine (exploration à des profondeurs importantes ou visite d'installations pétrolières), exploration spatiale (mission Mars Pathfinder), domaine du nucléaire... Ces diverses applications se déroulent le plus souvent en environnement peu structuré et évolutif nécessitant des moyens de perception et d'interprétation évolués pour fonctionner de manière autonome. Des missions comme la visite d'installations nucléaires ou l'exploration d'environnements inconnus ne permettent pas une autonomie totale, les robots sont dans ces cas des systèmes téléopérés.

La robotique mobile commence également à s'intégrer dans notre vie de tous les jours. Par exemple le système "Car Parking Aid"¹ (figure 1) d'assistance au stationnement d'une voiture, développé au "Roke Manor Research", combine des informations fournies par des capteurs embarqués (sonar et vision) afin d'exécuter une manœuvre de stationnement de façon rapide et précise. Le robot "Automower"² (figure 2), développé par "Electrolux", est une tondeuse qui ne nécessite pas de surveillance. Il part de sa station de charge et navigue de façon aléatoire dans l'environnement qu'il doit traiter. Il est également doté de capteurs de contact afin de détecter les collisions éventuelles avec des obstacles. Ces systèmes évoluent en environnements non structurés et inconnus et de ce fait n'ont pas recours à une procédure de localisation.

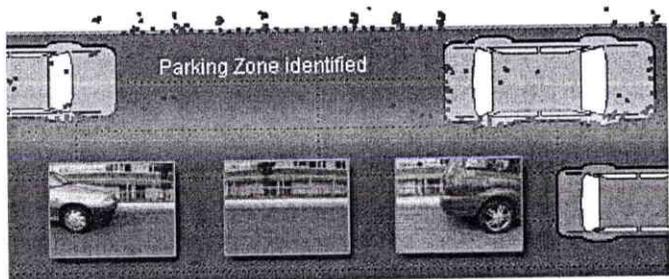


Figure 1 – Le système "Car Parking Aid".

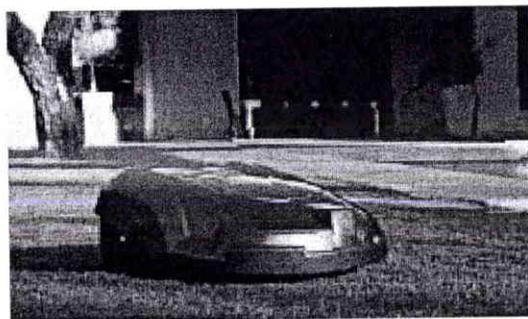


Figure 2 – Le robot tondeuse Automower de chez Electrolux.

Une autre application des robots mobiles concerne les domaines sociaux et médicaux. Dans ce cadre, un secteur de recherche porte sur le développement de fauteuils roulants électriques intelligents [Tra97, Pru03]. L'intégration d'un système intelligent sur un fauteuil électrique permet à l'utilisateur de se rendre d'un endroit à un autre sans assumer seul la tâche de conduite. De nombreuses études ont été menées dans ce domaine afin de développer des systèmes d'aide à la mobilité des personnes handicapées. La plupart des projets décrits dans la littérature sont relatifs à la détection et à l'évitement des obstacles, à la construction d'une carte de l'environnement proche, à la planification d'itinéraires sûrs, et au passage de porte. Un fauteuil roulant proposant de telles capacités requiert l'emploi de capteurs tels qu'un odomètre pour évaluer les mouvements du véhicule, des caméras, des capteurs à ultrasons ou infrarouges, ou encore des télémètres laser à balayage afin de percevoir l'environnement. Il établit parfois un modèle interne de

¹http://www.roke.co.uk/sensors/radar/parking_assistant.asp

²<http://www.automower.fr/>

l'environnement dans le but de résoudre divers problèmes de navigation.

Le travail développé dans ce mémoire s'intègre au projet VAHM³, conduit par le LASC⁴ depuis le début des années 90 qui vise à développer un nouveau type de véhicule intelligent destiné à l'assistance au pilotage des fauteuils roulants électriques. Cette assistance est réalisée par l'ajout de fonctionnalités facilitant la conduite du fauteuil sans pour autant accroître de façon excessive le prix global du système. Le bon déroulement de certaines de ces fonctionnalités, comme la navigation autonome, nécessite une connaissance de la localisation du véhicule par rapport à son environnement. Le prototype du VAHM doit intégrer deux modes de localisation :

- une localisation statique qui utilise uniquement les capteurs à ultrasons et une carte de l'environnement. Elle permet au véhicule de se localiser lorsqu'il démarre ou bien lorsqu'il a perdu toute information sur sa localisation actuelle.
- une localisation en cours de mouvement qui maintient une estimation correcte de la position du véhicule lorsqu'il se déplace. L'approche adoptée pour cette procédure fait l'objet de cette thèse.

L'objectif du travail présenté dans ce mémoire est de proposer une méthode dédiée à l'application qui se base sur l'utilisation des capteurs embarqués sur le fauteuil afin de fournir une bonne estimation de la localisation du robot. Cette méthode, devant être réalisée en cours de mouvement, devra être la moins coûteuse possible en temps de calcul. Elle se base sur une information odométrique corrigée en position et en orientation à l'aide des distances mesurées par les capteurs à ultrasons. L'utilisation d'un système de vision monoculaire vient préciser l'orientation.

Le fauteuil est destiné à naviguer dans des environnements d'intérieur comme un appartement ou des locaux hospitaliers. Ils sont généralement bien structurés et de configurations fixes, ce qui nous permet d'utiliser une carte de l'environnement établie a priori, générée à partir d'un plan du bâtiment et enregistrée sur la mémoire de masse du calculateur embarqué sur le véhicule.

Organisation du mémoire

Le mémoire se compose de trois chapitres.

Dans le chapitre un, nous présentons un état de l'art des capteurs et méthodes utilisés pour la localisation des robots mobiles à roues. Dans une première partie nous présentons les capteurs les plus couramment rencontrés en robotique mobile, nous y décrivons leurs caractéristiques ainsi que leurs avantages et inconvénients. Une seconde partie est réservée à la description des modèles de représentation de l'environnement. Enfin nous présentons un ensemble de méthodes de localisation que nous avons regroupées par thème en fonction du mode de représentation des mesures ou des formalismes utilisés pour les interpréter. La fin du chapitre est consacrée à une présentation de quelques projets ayant trait à l'assistance à la navigation de fauteuils intelligents.

Dans le chapitre deux, nous décrivons tout d'abord le projet VAHM, ses contraintes et ses objectifs. Cette présentation est suivie d'une évaluation de l'odomètre équipant le véhicule qui nous mène à la nécessité d'une correction régulière de la localisation qu'il délivre. La suite du chapitre est destinée à la description de la méthode de correction de la localisation odométrique du fauteuil par l'utilisation de capteurs à ultrasons. Nous présentons les modèles que nous avons choisis pour la représentation de l'environnement et des mesures fournies par les capteurs à ultrasons. Nous détaillons ensuite comment sont mises en correspondance ces données avant de procéder à la correction de l'estimation fournie par l'odomètre. Nous donnons enfin des résultats obtenus en simulation ainsi que des résultats obtenus à partir de données réelles. Une discussion sur les performances de la méthode conclut le chapitre.

Le chapitre trois est consacré à l'apport de la vision monoculaire pour la perception de l'environnement. Nous décrivons les modèles utilisés pour l'environnement, la caméra et les informations extraites de l'image. L'objectif de la méthode est de préciser l'estimation de l'orientation à partir d'une observation fournie par une caméra. Après une description des calculs nécessaires à la détermination de l'ajustement en orientation, nous étudions la complexité de la méthode ainsi que l'influence d'une erreur en position sur le calcul de

³Véhicule Autonome pour Handicapés Moteur

⁴Laboratoire d'Automatique des Systèmes Coopératifs

l'ajustement. Puis nous présentons des résultats obtenus en simulation et à partir d'images réelles. Nous concluons ensuite sur les apports de la méthode d'ajustement en orientation.

La conclusion dresse le bilan de la méthode de correction de la localisation odométrique par capteurs à ultrasons et vision monoculaire et considère les perspectives en terme d'amélioration de la méthode ou d'emploi des informations extraites des données capteurs dans un autre contexte.

Cette page est laissée blanche intentionnellement

Chapitre 1

Capteurs et méthodes pour la localisation

Chaque système de localisation pour robot mobile est une solution particulière au problème d'estimation de la position et de l'orientation du véhicule. Ils sont caractérisés par les informations disponibles et par la manière de les exploiter. Bien qu'il en existe une grande diversité, nous pouvons regrouper les méthodes en deux grandes familles :

- Les localisations relatives utilisent des capteurs proprioceptifs. Ces capteurs informent le robot sur ses mouvements relatifs par rapport à une position antérieure connue. En intégrant les informations capteurs, obtenues au cours de ses déplacements successifs, le robot peut estimer sa localisation actuelle. Etant donné le caractère intégrateur de ces méthodes, les erreurs de mesures sont également intégrées conduisant à une erreur d'estimation de la localisation non bornée.
- Les localisations absolues utilisent des capteurs extéroceptifs comme des capteurs à ultrasons, ou à infrarouges, des télémètres laser, ou encore des systèmes de vision. Ces capteurs permettent au robot de percevoir son environnement proche. Puis, en comparant cette perception locale avec une connaissance globale du monde dans lequel il évolue, il lui est possible de déterminer ou de corriger une estimation de sa localisation à l'intérieur de son espace de travail.

Ce chapitre s'organise de la manière suivante : dans une première partie nous dressons une liste des principaux capteurs rencontrés en robotique mobile, car une connaissance des caractéristiques des capteurs utilisés est nécessaire à la compréhension du problème de la localisation. Une seconde partie sera réservée à la description des méthodes de localisation en les regroupant en fonction de leurs similarités. Enfin, nous présenterons les modes de localisation choisis dans divers projets traitant du développement de fauteuils roulants électriques autonomes destinés à l'assistance de personnes à mobilité réduite.

1.1 Les capteurs en robotique mobile

Les capteurs utilisés en robotique mobile peuvent être répertoriés en deux familles suivant la nature des informations qu'ils fournissent au robot :

- les capteurs proprioceptifs renseignent le robot sur ses déplacements relatifs. Comme nous le verrons dans la suite, il existe différents types de capteurs proprioceptifs. Le plus communément utilisé en robotique mobile est le codeur incrémental qui mesure la rotation de la roue avec laquelle il est couplé.
- les capteurs extéroceptifs permettent au robot de percevoir l'environnement proche. Il en existe de nombreux modèles (ultrasons, infrarouges, laser, caméra) avec leurs avantages et inconvénients. La combinaison de différents capteurs permet d'accroître les capacités du robot à percevoir son environnement.

Pour bien appréhender le problème de la localisation, il est important de connaître les caractéristiques des divers capteurs disponibles. Dans cette partie nous décrivons quelques uns des différents capteurs utilisés en robotique mobile. L'odomètre, les capteurs à ultrasons et la vision seront plus particulièrement détaillés en raison de leur utilisation dans le travail présenté dans ce mémoire.

1.1.1 Les capteurs proprioceptifs

1.1.1.1 Codeurs optiques

Il existe deux catégories de codeurs optiques : les codeurs incrémentaux et les codeurs absolus. Les modèles incrémentaux mesurent des variations instantanées de la position angulaire et permettent de déterminer la position de l'axe de rotation par rapport à une position antérieure connue. Les modèles absolus permettent, quant à eux, directement une lecture de la position angulaire de l'axe.

Ces deux catégories de codeurs fonctionnent selon un principe similaire ; un faisceau de lumière partant d'un émetteur vers un récepteur est périodiquement interrompu par des raies sombres situées sur un disque translucide. En utilisant plusieurs couples émetteur/récepteur, il est possible de modifier la précision du capteur et également de déterminer le sens de rotation de l'axe couplé au disque.

(a) **Codeurs optiques incrémentaux** Le système d'encodage le plus simple est celui utilisant un seul canal optique, c'est-à-dire un seul couple émetteur / récepteur. Ils sont souvent appelés tachymètres (figure 1.1). La précision de ce capteur est directement liée au nombre de raies situées sur le disque. Ces capteurs, relativement peu coûteux, permettent des mesures aisées à des vitesses normales ou élevées, mais souffrent d'instabilité à des vitesses de rotation faibles, en raison des erreurs d'échantillonnage [Nic85]. De plus, le capteur ne fournit une information que si l'axe est en rotation.

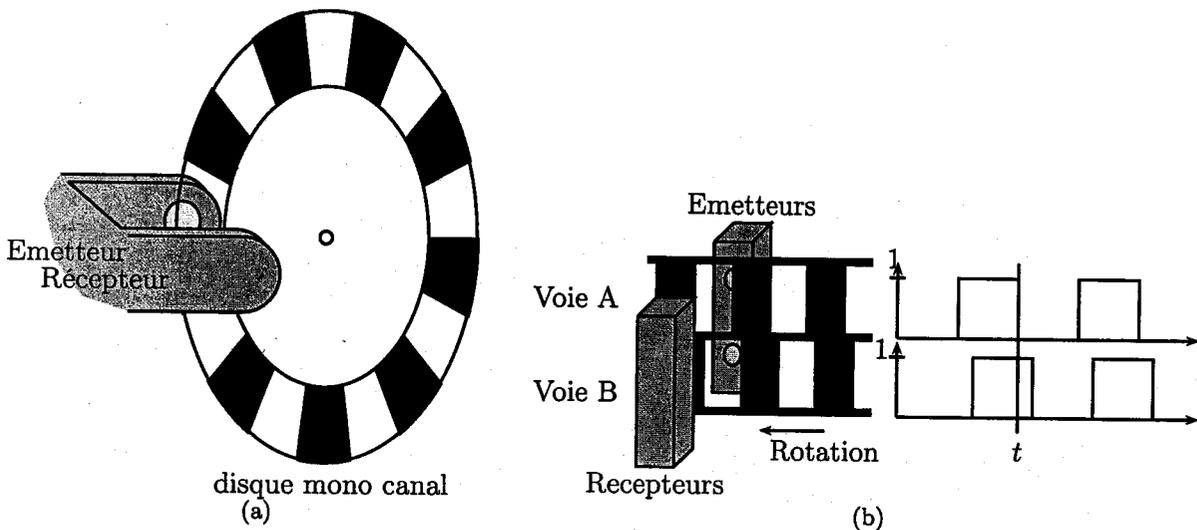


Figure 1.1 – Principe d'un codeur incrémental : (a) simple canal et (b) double canaux.

Etat	Voie A	Voie B
S_1	Haut	Bas
S_2	Haut	Haut
S_3	Bas	Haut
S_4	Bas	Bas

Tableau 1.1 – Table des états d'un codeur incrémental à double canaux.

Les capteurs incrémentaux mono canal ne fournissent pas d'information sur le sens de rotation de l'axe. Il n'est donc pas possible de déterminer le sens de rotation de la roue. Pour résoudre ce problème, un deuxième canal optique est ajouté 1.1(b). Il émet un signal qui est en quadrature de phase par rapport au premier. Ceci permet de détecter le sens de rotation suivant la séquence des états qui sont détectés par l'électronique embarquée, le tableau 1.1 donne un exemple de codage des états. La séquence $S_1-S_2-S_3-S_4$ indique une rotation dans le sens trigonométrique comme indiqué sur la figure 1.1, alors que la séquence $S_2-S_1-S_4-S_3$ indique une rotation dans le sens horaire.

La plupart des codeurs utilisent également un troisième canal, ne comportant qu'une raie de passage de la lumière, afin de détecter une révolution complète de l'axe lors d'une utilisation sur une roue par exemple.

(b) Codeurs optiques absolus Contrairement au codeur précédent, le codeur absolu est habituellement utilisé pour des applications fonctionnant à des vitesses de rotation faibles. L'élément de détection se compose d'un ensemble de détecteurs photosensibles. Le disque est codé suivant un code particulier qui permet de connaître de façon absolue la position angulaire de l'axe sans nécessiter une rotation de ce dernier. La précision du capteur est directement liée au nombre de bits utilisés pour coder les positions angulaires. La figure 1.2 montre un exemple de disque utilisant un code de Gray sur 8 bits.

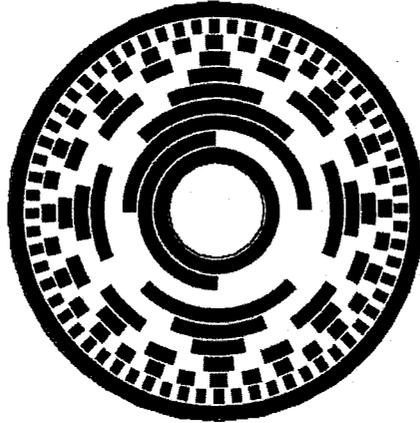


Figure 1.2 – Exemple du disque d'un codeur absolu utilisant un code de gray sur 8 bits.

Un disque codé en code Gray sur N bits comporte N pistes concentriques. La première piste en partant du centre du disque est composée d'une moitié opaque et d'une moitié transparente. La lecture de cette piste permet de déterminer dans quel demi-tour du disque on se situe. C'est la piste MSB (Most Significant Bit = bit de poids le plus fort). La piste suivante, en allant vers l'extérieur, est déphasée de 90° par rapport à la première. La piste suivante est divisée en quatre quarts alternativement opaques et transparents. La lecture de cette piste, combinée avec la lecture des deux pistes précédentes, permet de déterminer dans quel quart de tour du disque on se situe. La piste suivante permet de déterminer dans quel huitième de tour on se situe, la suivante dans quel seizième de tour, et ainsi de suite. La dernière piste, la plus extérieure, est la piste LSB (Least Significant Bit = bit de poids le plus faible). C'est elle qui donne la précision finale du codeur appelée résolution.

- Si on appelle N le nombre de pistes, cette dernière piste comportera 2^{N-2} points. Le nombre de positions codées sur un tour du disque sera 2^N , on dira alors que le disque codé a une résolution de 2^N points par tour.
- La lecture simultanée de toutes les pistes (informations Gray) nous donne un code Gray représentatif de la position du disque du capteur dans le tour.

Les codeurs absolus sont bien adaptés aux mouvements lents ou pour des rotations peu fréquentes, tel que le réglage d'un angle d'orientation, plutôt que pour la mesure de rotation rapide et continue comme le mouvement de roues. Un inconvénient de ce type de codeur est le nombre de données à placer sur le disque, qui requiert une fabrication soignée et donc coûteuse. Un autre inconvénient est la complexité de l'électronique embarquée qui est accrue en raison du nombre d'informations à lire sur le disque pour déterminer sa position angulaire.

1.1.1.2 Capteurs inertiels

C'est un capteur qui est depuis longtemps utilisé pour la navigation dans le domaine de l'aéronautique. Couplé à des balises radio et à un GPS, ils constituent les instruments de base pour la navigation d'un avion. Les capteurs inertiels peuvent être utilisés pour la navigation à l'estime (dead-reckoning navigation). Il en existe différents types, comme l'accéléromètre ou le gyroscope.

(a) **Accéléromètre** Ce capteur mesure, comme son nom l'indique, l'accélération. L'accélération linéaire et l'accélération angulaire peuvent être toutes les deux mesurées. La mesure à l'aide d'une jauge de contrainte de la force \vec{F} appliquée à un mobile de masse m permet de déterminer l'accélération $\gamma = F/m$ que subit ce mobile et donc le véhicule (figure 1.3). Les mesures fournies par ces capteurs doivent être intégrées deux fois pour donner une information de position ce qui induit une faible immunité au bruit pour les faibles accélérations.

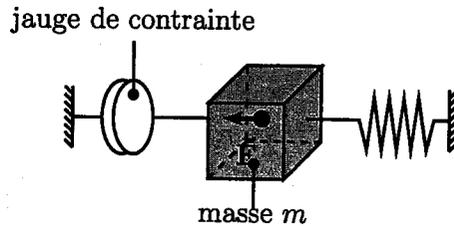


Figure 1.3 – Principe de fonctionnement d'un accéléromètre.

(b) **Gyroscope** Ce capteur peut mesurer des variations angulaires et linéaires. Son fonctionnement repose sur des principes mécaniques ou optiques

- gyroscopes mécaniques : ils sont basés sur l'observation des changements des moments linéaires ou angulaires. Ils se basent sur la loi de la conservation des forces. Ce sont ces forces de rappel que l'on ressent lorsque l'on tient à bout de bras une roue de vélo tournant à grande vitesse et que l'on tente de l'incliner. Elles tendent à ramener la roue verticalement. En mesurant ce type de force il est possible de quantifier la variation de la direction d'un véhicule. La figure 1.4 donne un exemple de configuration mécanique d'un gyroscope mécanique.

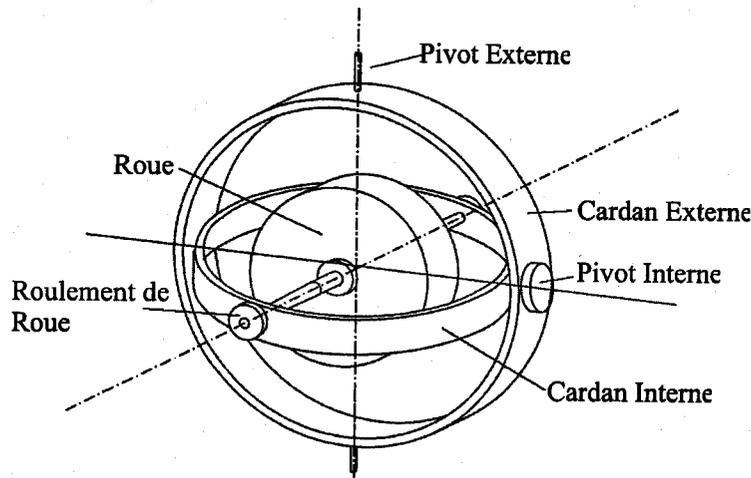


Figure 1.4 – Configuration d'un gyroscope mécanique à deux axes [Eve95].

La roue, mue électriquement, est suspendue par une paire de roulements de précision à faible frottements à chacune des extrémités de son axe. Les roulements du rotor en mouvement sont soutenus par le cardan interne qui pivote sur un deuxième ensemble de roulements qui le lie au cardan externe. Cette action de rotation du cardan interne définit l'axe horizontal du compas gyroscopique, qui est perpendiculaire à l'axe de rotation du rotor comme représenté sur le schéma de la figure 1.4. Le cardan externe est attaché à l'armature de l'instrument par un troisième ensemble de roulements qui définit l'axe vertical du compas gyroscopique. L'axe vertical est perpendiculaire à l'axe horizontal et à l'axe de rotation. En positionnant des jauges de contraintes sur les axes de rotation, les forces de rappel peuvent être mesurées sur ces axes. Lorsque le robot est en mouvement, des forces de rappel se créent sur les différents axes. Les mesures de ces forces sont utilisées pour déterminer la vitesse des mouvements. Ce type de gyroscope est sensible à la rotation de la terre si l'axe de rotation du capteur n'est pas orienté dans la direction nord-sud, ce qui ne peut pas être garanti en robotique mobile.

- gyroscopes optiques : ils utilisent la lumière pour mesurer les mouvements. L'idée est de laisser deux faisceaux laser identiques parcourir en sens inverse une fibre optique formant un anneau. Les deux faisceaux se superposent créant des interférences qui sont utilisées pour déterminer la rotation de l'anneau autour de son axe. En effet, le faisceau qui parcourt le cercle dans le sens de rotation de ce dernier aura une vitesse relative plus faible que l'autre, cela se mesure sur les interférences. Les changements intervenant sur les interférences donnent accès à la vitesse angulaire et au sens de rotation de la fibre circulaire. Par rapport aux systèmes mécaniques, ils présentent l'avantage de ne pas posséder de parties en mouvement ce qui rend leur maintenance plus facile.

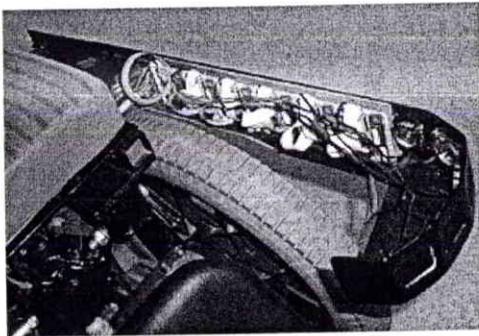
Les capteurs inertiels peuvent être utilisés pour mesurer à la fois la rotation et la translation du mobile. Dans certains systèmes robotiques, ils sont un complément intéressant pour un système odométrique. Le capteur inertiel est également sujet aux dérives mais, combiné avec un système odométrique, les performances peuvent être grandement améliorées [Goe99].

1.1.2 Capteurs extéroceptifs

1.1.2.1 Télémètres à ultrasons

Le capteur à ultrasons, appelé également sonar, est le capteur extéroceptif le plus largement utilisé en robotique mobile [Eve95]. Il a été originellement utilisé dans les applications sous-marines et dans les systèmes auto-focus d'appareils photographiques [Cro85]. Ce capteur est largement disponible, peu coûteux et facile à mettre en œuvre [Ras90], ce qui explique sa popularité ainsi que la grande quantité de travaux basés sur son utilisation [Mor85, Cro85, Dru87, Ayc97, Thr02, Kre03].

De nombreuses plate-formes commerciales ou non commerciales sont équipées de ceinture de capteurs à ultrasons. Le nombre de capteurs utilisés varie de 12 à 30. Les figures 1.5(a) et 1.5(b) montrent respectivement un exemple de ceinture de capteurs conçue pour le prototype du VAHM et une ceinture intégrée à un modèle de robot commercial.



(a) Partie de la ceinture de capteurs à ultrasons du prototype du VAHM



(b) Ceinture de capteurs à ultrasons sur le mobile Magellan (I-Robots, Real World Interface, Div U.S.A.)

Figure 1.5 – Exemples de configuration de ceintures de capteurs à ultrasons.

(a) Physique de base d'un capteur Nous décrivons ici rapidement la partie physique de ces capteurs, [Kuc87, Eve95, Bor96, Pol91] en donnant une description plus complète. Le sonar se base sur l'énergie acoustique pour effectuer ses mesures. L'onde ultrasonore est créée en utilisant un émetteur qui peut être de deux types : piézoélectrique ou électrostatique.

Piézoélectrique : Ce capteur est basé sur un cristal piézoélectrique qui oscille lorsqu'une tension est appliquée à ses bornes. Les oscillations sont de faibles amplitudes, mais la puissance peut être importante. L'air étant un milieu à faible densité, la transmission des vibrations n'est pas aussi bonne que dans l'eau qui, plus dense, permet une meilleure transmission. Le capteur piézoélectrique

peut opérer à de hautes fréquences, de l'ordre du MHz . Il nécessite peu de cycles pour son démarrage et son arrêt en raison de sa faible inertie.

Electroacoustique : Il se compose d'une fine membrane que l'on fait vibrer. Les vibrations sont d'amplitudes plus importantes que le capteur précédent, ce qui lui confère de meilleures propriétés dans des milieux comme l'air. La puissance est cependant plus faible. La plage des fréquences utilisables est plus importante que pour un capteur piézoélectrique, mais la limite supérieure n'est que de l'ordre d'une centaine de kHz .

(b) Principes de la mesure de distance par sonar L'idée de base de la mesure de distance par sonar est la suivante. L'émetteur à ultrasons produit un court train d'ondes ultrasonores, et un récepteur reçoit l'écho de ce train d'ondes. Le signal de retour peut être traité de différentes façons, mais la manière la plus couramment rencontrée reste la mesure du temps de vol (time-of-flight en anglais (TOF)), c'est-à-dire la mesure du temps écoulé entre le début de l'émission et la fin de la réception du signal acoustique. Le signal reçu est intégré jusqu'à valider un seuil¹, à cet instant, le temps est mesuré. Connaissant la célérité du son dans l'air ($\approx 340 m/s$) et le temps de vol, il est facile de calculer la distance parcourue par l'onde. Par contre, il est plus délicat de connaître les paramètres comme la température, l'humidité, et la pression atmosphérique, alors que ceux-ci modifient la vitesse du son. Ces paramètres n'étant pas connus exactement, ils contribueront à l'incertitude sur la mesure de la distance. En environnement d'intérieur, ces paramètres sont plus ou moins constants. La distance à l'objet ayant renvoyé l'onde est donnée par la relation 1.1 :

$$D = \frac{1}{2}cT \quad (1.1)$$

où c est la célérité du son, et T le temps d'aller-retour mesuré. Il existe plusieurs facteurs qui ont un effet sur la propagation de l'énergie acoustique :

- L'intensité décroît proportionnellement à l'inverse de la distance parcourue au carré, car l'émission se fait dans un cône et non en ligne droite.
- La réflectivité de la cible conditionne la quantité énergétique renvoyée.
- Le vent et la température affectent la propagation de l'onde dans l'air. La vitesse du son est proportionnelle à la racine carrée de la température.
- L'onde acoustique est diffractée par le capteur et se propage à l'intérieur d'un cône (figure 1.6). Le capteur le plus fréquemment utilisé, l'émetteur/récepteur Polaroid, possède une ouverture à $3dB$ de 25° et une plage de mesure² de $30 cm$ à $10 m$ [Fly88]. En raison du phénomène de diffraction, il n'est pas possible de déterminer la direction de détection de l'objet à partir d'une seule position du robot. L'objet est situé quelque part sur la surface sphérique terminale du cône, noté S sur la figure 1.6.

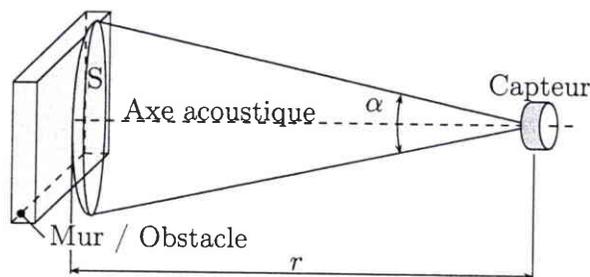


Figure 1.6 – Cône d'émission de l'énergie acoustique.

La mesure de la distance à un objet (r sur la figure 1.6) est déterminée par la mesure du temps de vol, mais elle ne permet pas de déterminer la direction de détection de l'objet à partir d'une seule position du capteur.

¹Nombre d'impulsions ultrasonores reçues.

²Il est possible de réduire la limite inférieure en agissant sur l'électronique de contrôle du capteur.

(c) **Avantages et inconvénients du capteur à ultrasons** Suivant l'utilisation du capteur à ultrasons, certaines de ses caractéristiques physiques peuvent apparaître comme un avantage ou un inconvénient. Ces caractéristiques sont :

- La largeur du faisceau, qui rend impossible la détermination de la direction exacte de détection à l'intérieur du cône. Cette caractéristique n'est pas seulement un inconvénient, puisqu'elle confère au capteur un grand champ de perception, ce qui peut être intéressant dans certains cas comme pour l'évitement d'obstacle. Par ailleurs pour lever l'incertitude sur l'axe de détection plusieurs lectures successives peuvent être interprétées. Ainsi dans [Wij98] les auteurs supposent qu'à la suite d'un petit déplacement du robot, le capteur détecte toujours la même caractéristique naturelle de l'environnement. Et au fur et à mesure des déplacements du robot, celui-ci peut affiner la position de la cible par rapport au robot. La figure 1.7 donne un exemple de l'affinement de l'estimation de la position d'une cible au cours des déplacements du robot ; nous pouvons constater la précision croissante de l'estimation de la position relative de la cible par rapport au robot lorsque n_t croît.

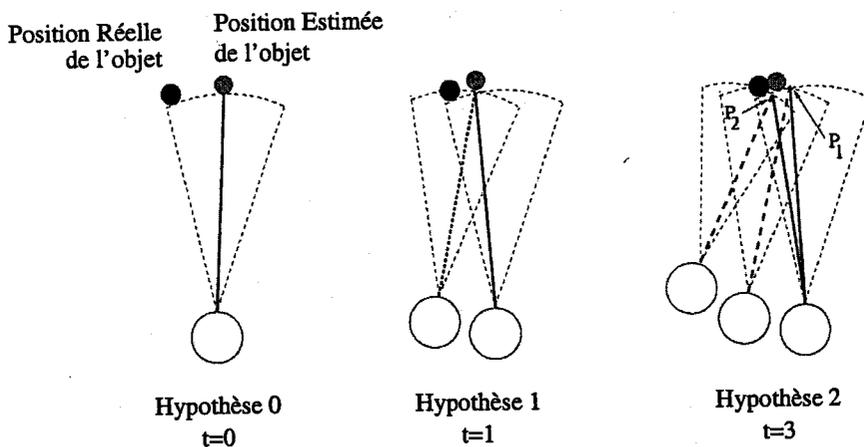


Figure 1.7 – Principe de la fusion des mesures provenant d'une même cible, d'après [Wij98].

- Le temps d'acquisition non négligeable, en raison de la célérité du son. Lorsque plusieurs capteurs à ultrasons sont utilisés sur un même système et que ceux-ci travaillent à la même fréquence, il se peut qu'un capteur perçoive l'écho provenant d'un autre capteur. Ce phénomène est connu sous le nom de diaphonie (crosstalk en anglais) (figure 1.8). Sur cet exemple, le capteur 1 émet une onde qui se réfléchit et recouvre le capteur 2. Au moment où l'onde réfléchie atteint le capteur 2, si celui-ci est en réception alors il mesure une distance erronée. Une solution destinée à contrer ce problème consiste à laisser un temps d'acquisition suffisamment long avant d'activer un autre capteur, généralement 300 à 600 ms [Bor92] ce qui ralentit la prise de mesure. Une méthode, nommée EERUF (Error Eliminating Rapid Ultrasonic Firing), a été développée pour éliminer la diaphonie tout en augmentant la vitesse de tir par un facteur de cinq à dix [Bor92]. Elle nécessite la possibilité de commande des capteurs à un bas niveau électronique, ce que la plupart des ceintures ne permettent pas. Des projets récents portent sur des robots communicants [Bal95, How03]. Si différents robots proches les uns des autres se déplacent, le problème de diaphonie doit être pris en compte.

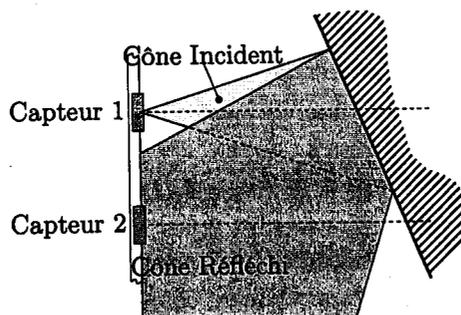


Figure 1.8 – Principe de la diaphonie

- La sensibilité à la nature de la surface perçue qui intervient lorsque le capteur n'est pas normal à la surface. Si la rugosité de la surface est petite par rapport à la longueur d'onde de l'onde émise par le capteur alors celle-ci se réfléchit comme un rayon lumineux sur un miroir avec un angle égal à l'angle d'incidence et peu de diffusion, la réflexion est dite spéculaire. L'énergie renvoyée vers le capteur peut devenir très faible, voire indétectable si la surface est trop inclinée par rapport à la direction de tir du capteur. Seul l'angle d'ouverture du faisceau limite ce phénomène. Dans [Wal87], l'auteur indique qu'une surface est considérée comme spéculaire pour un type d'onde lorsque la rugosité de la surface est inférieure à 5% de la longueur d'onde. Pour une onde ultrasonore la longueur d'onde typique est de $\lambda = cT = \frac{c}{f} = 8.5 \text{ mm}$, où c est la célérité du son dans l'air et f la fréquence de vibration du capteur (40kHz pour un capteur Polaroid), ce qui donne une rugosité minimale de 0.4 mm pour que la surface puisse être perçue.

Différentes techniques peuvent être mises en œuvre pour améliorer la validité des mesures issues des capteurs à ultrasons. Dans [Fly88], l'auteur propose un système qui utilise quatre fréquences distinctes afin de limiter l'effet de la diaphonie. On peut aussi gérer l'activation des acquisitions des modules ultrasons Polaroid, comme cela est fait dans [Bor92] avec la méthode EERUF. Par ailleurs des capteurs spécifiques peuvent être conçus comme dans [Cho97] où les auteurs utilisent une matrice de capteurs composée de trois récepteurs et trois émetteurs placés dans une certaine configuration, ce qui permet de localiser et classifier les caractéristiques de l'environnement détectées comme étant des plans, des coins, des intersections de couloirs, ou non identifiées.

(d) **Modèles des capteurs à ultrasons** Afin de représenter les mesures, de nombreux modèles de capteurs à ultrasons ont été proposés dans la littérature. Par exemple, celui proposé dans [Cro85] suppose que la mesure retournée par le capteur se situe soit sur l'axe acoustique de ce dernier, soit sur l'extrémité droite ou gauche de l'arc de cercle (figure 1.9(a)). Puis dans [Cro89] l'auteur affine le modèle du capteur et introduit l'idée que l'écho provient d'un obstacle situé sur un arc de cercle à la distance renvoyée par le capteur (figure 1.9(b)). Moravec et Elfes [Mor85] introduisent l'utilisation d'un profil probabiliste pour le capteur à ultrasons (figure 1.10) qui définit la probabilité d'occupation et de non occupation de l'espace situé devant le capteur (figure 1.11). Ce profil est projeté sur une carte discrète, indiquant ainsi la probabilité d'occupation d'une cellule située dans l'empreinte discrète du faisceau du capteur à ultrasons. Dans [Lim92], les auteurs se basent sur un modèle modifié de celui proposé dans [Mor85] auquel ils ajoutent une mesure de confiance, basée sur la distance (r) retournée par le capteur à ultrasons (la mesure de confiance décroît lorsque la distance augmente) et l'angle (θ) entre l'axe acoustique du capteur et la droite partant du capteur et passant par la cellule i (figure 1.12).

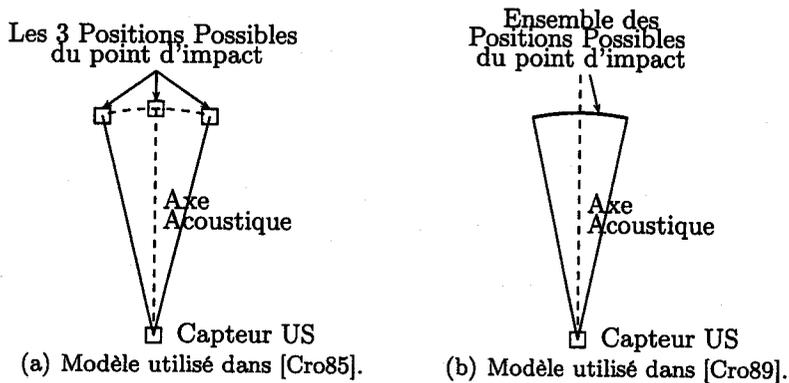


Figure 1.9 – Modèles de capteur à ultrasons

Afin de limiter la charge de calcul nécessaire à la manipulation de carte discrète, Borenstein et Koren proposent un modèle simplifié du modèle de Moravec et Elfes qui ne considère que l'axe acoustique du capteur. Dans un premier temps leur modèle n'est constitué que du point situé à la distance renvoyée par le capteur [Bor89]. Puis dans [Bor90a] ils affinent le modèle en considérant l'espace situé sur l'axe acoustique entre le capteur et le point comme étant probablement libre. Dans [Leo91], les auteurs introduisent le concept de régions de profondeur constante (RCD : Regions of Constant Depth). Ils décrivent un modèle

physique d'un capteur à ultrasons – système Polaroid – qui prend en compte les échos pouvant être renvoyés par des coins, murs et bords de murs présents dans un environnement, et établissent la nécessité de la perpendicularité entre l'axe acoustique du capteur et la surface détectée afin d'obtenir une mesure de distance précise.

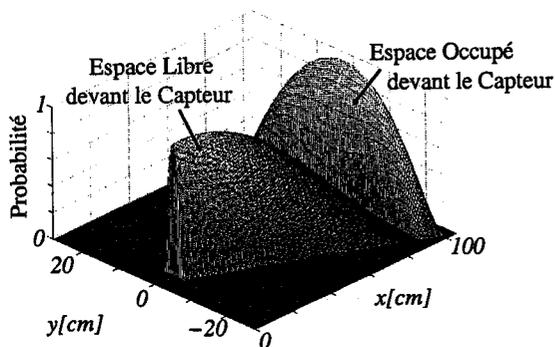


Figure 1.10 – Modèle de capteur à ultrasons utilisé dans [Mor85].

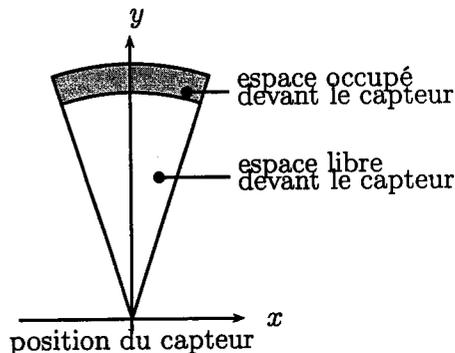


Figure 1.11 – Représentation de l'espace devant le capteur à ultrasons .

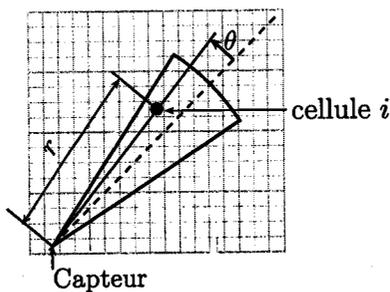


Figure 1.12 – Paramètres pris en compte pour déterminer la confiance de l'occupation de la cellule dans [Lim92].

1.1.2.2 Lasers

(a) **Techniques de mesures** Les deux techniques dominantes pour la mesure de distance par laser sont les techniques de mesure du temps de vol (Time of Flight – TOF) et les techniques de mesure de différence de phase.

(a).1 **Technique par mesure du temps de vol** Dans cette technique, un court train d'ondes est émis et le temps écoulé jusqu'à sa réception après réflexion est mesuré. La distance mesurée est donc :

$$D = \frac{1}{2}cT \tag{1.2}$$

où c est la vitesse de la lumière et T est le temps écoulé entre l'instant d'émission du début du train d'ondes et la fin de réception du train d'ondes. Afin de réaliser ce type de système, une mesure précise du temps est nécessaire. Comme en robotique mobile une résolution de l'ordre de quelques centimètres est généralement requise et que la vitesse de la lumière est $c = 3 \cdot 10^8 \text{ ms}^{-1}$, il est nécessaire d'avoir une mesure du temps d'une précision de l'ordre de 100 ps , ce qui correspond à une fréquence de 10 GHz . Un avantage des courts trains d'onde est qu'une forte puissance peut être utilisée, donnant ainsi une meilleure couverture pour la distance, sans pour autant consommer beaucoup d'énergie, ce qui est d'un grand intérêt pour les systèmes autonomes. Les systèmes commerciaux actuels ont une précision inférieure au centimètre, mais pour un prix nettement supérieur aux capteurs à ultrasons.

(a).2 **Mesure de la différence de phase** Pour les systèmes à mesure de la différence de phase, l'onde est transmise en permanence. L'idée est de comparer la phase de l'onde renvoyée avec celle de l'onde émise. En utilisant l'effet Doppler, la vitesse relative de l'obstacle peut être mesurée ainsi que sa distance au capteur.

Un des problèmes du laser à mesure de différence de phase est qu'il est incapable de détecter des différences de phases supérieures à la longueur d'onde en raison de la périodicité de celle-ci [And92]. Les distances supérieures à la longueur d'onde ne peuvent pas être distinguées des distances inférieures à cette dernière.

(b) **Lasers à balayage** Un télémètre laser mesure la distance dans une seule direction. En le montant sur un système rotatif, il est possible de réaliser un effet de balayage, ce qui permet d'effectuer des mesures dans des directions multiples. Afin de simplifier le mécanisme de rotation les modèles commerciaux de laser utilisent un miroir rotatif qui permet de couvrir un champ important. Ce capteur est couramment utilisé en robotique [Hop90, Buc93, Bor93, For93, Wei95].

Comparé au capteur à ultrasons le laser à balayage reste cher. Dans certaines applications comme les fauteuils roulants électriques, l'utilisation d'un laser à balayage génère une augmentation relative du coût du prototype plus importante que pour d'autres applications, comme la conduite de véhicule de chantier [Bou00].

Il est également possible d'ajouter plus d'un degré de liberté au laser et ainsi obtenir des lasers à balayage 3D [Nas92].

(c) **Avantages et inconvénient des lasers** Les avantages d'un système utilisant un laser sont les suivants :

- La rapidité : la mesure peut dans la plupart des cas être considérée comme instantanée. Cela permet de ne pas avoir à tenir compte du mouvement du capteur durant la mesure. Ceci est vrai pour le laser fixe, mais dans le cas d'un laser tournant il faut tenir compte du temps de balayage.
- La bonne précision en distance : Dans [For93], les auteurs rapportent un écart-type de 2.0 cm et les lasers à balayage de dernière génération de SICK Electro-Optics sont donnés pour une précision de l'ordre des 10 mm et une résolution angulaire de 0.5°.
- L'ouverture angulaire est plus faible pour un laser que pour un capteur à ultrasons, ce qui permet de connaître plus précisément la direction dans le repère capteur de l'obstacle ayant renvoyé le signal. La résolution du PLS Laser Scanner de chez SICK Electro-Optics est de 0.5°.
- Tout comme le capteur à ultrasons, l'information fournie par une mesure laser peut être directement exploitée comme la distance à un obstacle dans une certaine direction. Ce qui n'est pas le cas pour une caméra seule qui nécessite un traitement complémentaire important pour extraire les informations utiles de l'image.

Les désavantages de ce type de systèmes sont les suivants :

- Le capteur fournit uniquement une information dans un plan, alors que le capteur à ultrasons en raison de son cône d'émission permet de détecter des obstacles situés plus bas, ou plus haut, que le capteur, ce qui peut présenter un avantage en robotique d'intérieur.
- C'est un capteur encore très cher pour des applications domestiques comme les fauteuils roulants autonomes.
- Certains matériaux ne sont pas détectables par le laser comme par exemple les miroirs. Il reste cependant la possibilité de combiner le laser avec un autre capteur pour lequel ces matériaux ne sont pas invisibles.

1.1.2.3 Capteurs infrarouges

Ce capteur se compose d'un couple émetteur/récepteur. L'émission s'effectue par diode électroluminescente infrarouge (longueur d'onde $\lambda = 950 \text{ nm}$). Le récepteur est une photodiode ou un phototransistor, sensible au flux lumineux rétrodiffusé par la cible. Trois paramètres sont à prendre en compte pour la réalisation d'un tel capteur (figure 1.13):

- la distance entre émetteur et récepteur l ,

- l'inclinaison des axes émetteur et récepteur,
- l'ouverture du faisceau à l'émission et à la réception.

Ces paramètres déterminent les performances du capteur, comme la courbe de réponse dans l'étendue de mesure. Cette réponse est non linéaire et varie en fonction de l'inclinaison de l'objet visé ainsi que du rapport entre le flux lumineux reçu et le flux lumineux rétrodiffusé, lié à la couleur et à l'état de la surface perçue (ce rapport est souvent nommé albedo). Un bon étalonnage permet néanmoins d'atteindre une précision de l'ordre du centimètre.

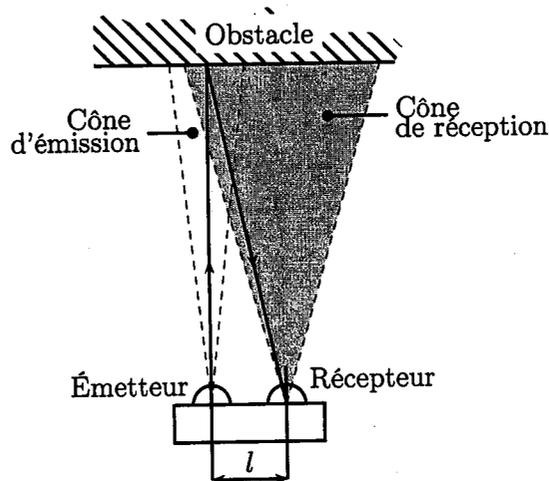


Figure 1.13 – Paramètres utiles à la conception d'un capteur infrarouge.

Cette technique présente quelques défauts :

- L'énergie n'est pas fortement concentrée dans une direction bien définie comme pour un laser, ce qui limite la portée de ces capteurs. Les obstacles éloignés ne sont donc plus détectés. La distance limite est en général de l'ordre de 1 m.
- Comme la mesure dépend de l'intensité retournée, le matériau sur lequel se réfléchit la lumière influence la mesure de la distance.

Ce capteur est d'un coût actuellement inférieur à celui d'un capteur à ultrasons et sa mise en œuvre est du même ordre de difficulté. Il est également sensible au problème de diaphonie lorsque plusieurs de ces capteurs fonctionnent à la même fréquence. Il présente un désavantage supplémentaire qui est la difficulté à détecter des surfaces comme les miroirs et les vitres, ce qui n'est pas le cas des capteurs à ultrasons. Il est employé dans de nombreuses applications de robotique mobile ludique (coupe de robotique) pour les raisons suivantes :

- son faible encombrement, car il ne nécessite pas de module additionnel contrairement au capteur à ultrasons Polaroid,
- sa consommation très faible : seulement 25 mA pour le modèle analogique de SHARP alors que le capteur à ultrasons Polaroid consomme 100 mA en veille et 2 A lors de l'émission du train d'ondes,
- son taux d'acquisition constant : quelle que soit la distance mesurée, le capteur donne information toutes les 40 ms, pour les capteurs à ultrasons ce temps dépend de la distance maximale que l'on souhaite mesurer (pour 11 m le temps d'acquisition est de l'ordre de 74 ms),
- sa directivité meilleure que celle d'un capteur à ultrasons par l'utilisation de lentilles additionnelles.

Dans [LoP02], les auteurs présentent différents capteurs utilisables pour un fauteuil roulant intelligent. Ils mettent en évidence la différence de qualité de perception entre différents capteurs infrarouges et ultrasons en fonction du matériau sur lequel le signal se réfléchit. Dans cette étude, les capteurs à infrarouges utilisés possèdent un angle d'incidence limite plus grand que ceux des capteurs à ultrasons utilisés. Cette étude met en évidence que la distance de perception maximale d'un capteur à ultrasons est supérieure à celle d'un capteur à infrarouges. Les auteurs concluent que ces différences entre les capteurs peuvent être utilisées avantageusement en combinant ces deux types de capteurs (un modèle à infrarouge avec un modèle à ultrasons).

1.1.2.4 Capteur de vision

L'élément de base d'un système de vision est la caméra. Une caméra est caractérisée par deux groupes de paramètres ; les paramètres intrinsèques et les paramètres extrinsèques. Alors que les premiers définissent la géométrie interne de la caméra, les seconds permettent de définir la localisation (position et orientation) tridimensionnelle de la caméra.

La caméra permet la projection d'un espace à trois dimensions vers une image en deux dimensions. Quel que soit le modèle de caméra, les éléments suivants sont toujours présents (figure 1.14):

- le centre de la caméra C (également nommé centre optique),
- le point principal P confondu avec le centre de l'image,
- l'axe optique : droite passant par les points C et P ,
- la distance focale : distance entre le point C et le point P ,
- le plan image : plan sur lequel se projettent les points 3D de l'espace, il est normal à l'axe optique et est situé en P .

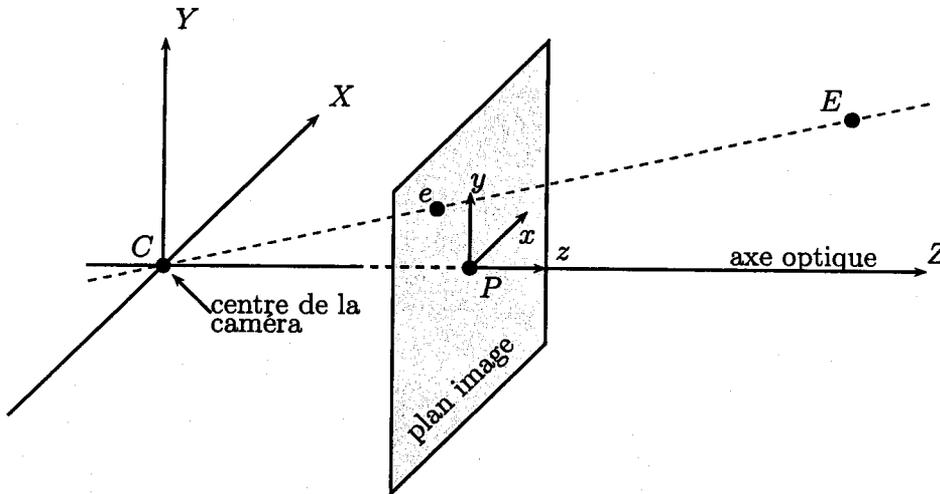


Figure 1.14 – Géométrie du modèle de caméra sténopé

Le modèle de caméra le plus simple est le modèle "trou d'épingle" (ou modèle sténopé) qui définit la projection perspective de centre C de l'espace en trois dimensions sur l'image en deux dimensions. Voyons à présent les divers paramètres qui caractérisent ce modèle.

(a) **Modèle trou d'épingle** Le référentiel de la caméra est placé sur le centre C de la caméra avec l'axe des z confondu avec l'axe optique et orienté positivement vers P . Dans le cas du modèle sténopé, un point de l'espace E , de coordonnées $(X_E, Y_E, Z_E)^T$ dans le référentiel global, est projeté sur le plan image à l'endroit où la droite joignant le point E au point C traverse le plan image. Ce point est le point e de coordonnées (x_e, y_e) dans le référentiel de plan image (figure 1.15).

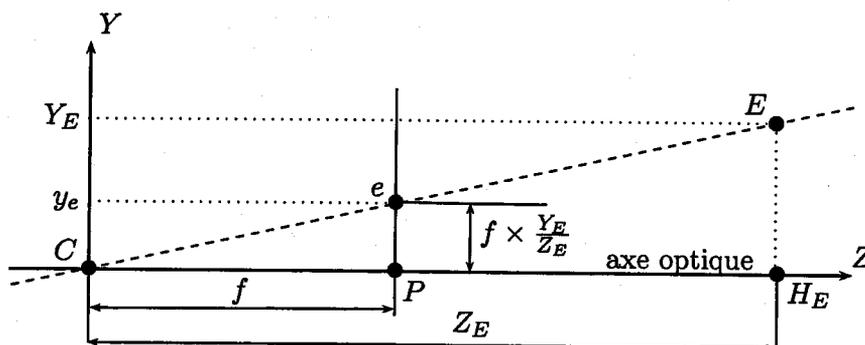


Figure 1.15 – Vue de côté de la projection au travers d'un modèle de caméra sténopé

Dans les triangles C, P, e et C, H_E, E , le rapport constant des distances donne :

$$\frac{CP}{CH_E} = \frac{Pe}{H_E E} \quad (1.3)$$

ce qui s'écrit également :

$$\frac{f}{Z_E} = \frac{y_e}{Y_E} \quad (1.4)$$

soit :

$$y_e = f \times \frac{Y_E}{Z_E} \quad (1.5)$$

il en est de même pour :

$$x_e = f \times \frac{X_E}{Z_E} \quad (1.6)$$

Ce qui peut s'écrire :

$$\begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f}{Z_E} & & \\ & \frac{f}{Z_E} & \\ & & \frac{1}{Z_E} \end{pmatrix} \begin{pmatrix} X_E \\ Y_E \\ Z_E \end{pmatrix} \quad (1.7)$$

Ce modèle suppose que la caméra utilisée a son axe optique qui passe effectivement par le centre du plan image, ce qui peut ne pas être le cas. Alors deux paramètres supplémentaires sont intégrés dans la matrice de projection perspective \mathcal{P} :

$$\mathcal{P} = \begin{pmatrix} \frac{f}{Z_E} & & \frac{x_0}{Z_E} \\ & \frac{f}{Z_E} & \frac{y_0}{Z_E} \\ & & \frac{1}{Z_E} \end{pmatrix} \quad (1.8)$$

où (x_0, y_0) définit le décalage entre le centre du plan image et le centre de l'image. De plus, si les pixels ne sont pas carrés mais rectangulaires, alors la matrice de projection perspective devient :

$$\mathcal{P} = \frac{1}{Z_E} \begin{pmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{pmatrix} \quad (1.9)$$

où $\alpha_x = f \times k_x$ et $\alpha_y = f \times k_y$. Le rapport $\frac{k_x}{k_y}$ généralement donné par le constructeur est de l'ordre de 0.7 pour les caméras courantes.

Enfin, pour tenir compte du défaut d'orthogonalité du plan des cellules CCD par rapport à l'axe optique, le paramètre γ est introduit. Si ce paramètre n'est pas nul, cela signifie que les pixels n'ont pas une forme de rectangle mais de parallélogramme. En pratique, sa valeur est très faible. Ainsi, la projection perspective est de façon la plus générale représentée par la relation suivante :

$$Z_E \mathbf{e} = \begin{pmatrix} \alpha_x & \gamma & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{E} \quad (1.10)$$

Le modèle perspectif est le plus employé mais d'autres variantes existent [Chr98], notamment :

- le modèle de projection para-perspective,
- le modèle de projection affine, appelée également orthographique.

La projection para-perspective a été initialement utilisée en reconstruction 3D, elle est bien adaptée à la focalisation d'attention autour d'un angle de vue particulier. Elle nécessite que l'objet soit proche de l'axe optique ou loin de la caméra, la projection est alors correctement approximée par la composée d'une projection parallèle sur un plan orthogonal à l'axe optique et d'une projection perspective. La projection orthographique est un cas particulier de la projection para-perspective pour laquelle la direction de vue est la même que celle de l'axe optique de la caméra.

(b) **Système de vision monoculaire** Il utilise une caméra modélisée à l'aide du modèle de projection perspectif le plus simple. Des composantes sont recherchées dans l'image qui peuvent être des contours [Lee00]. Mais des primitives géométriques ne sont pas les seules à être utilisées. Certains auteurs utilisent des textures ou parties de l'image dont l'emplacement est connu dans l'environnement, par exemple dans [Thr00a], les auteurs utilisent la texture du plafond, i.e. la position des lampes. Leurs positions dans l'image donnent une indication sur la position de la caméra.

(c) **Système de stéréovision** Ce système se compose d'au moins deux caméras. En mettant en correspondance les points d'intérêts présents dans les deux images et en utilisant la géométrie projective, affine ou euclidienne [Har01], il est alors possible de retrouver la position relative des objets observés.

(d) **Différents types de systèmes de vision omnidirectionnelle** Ils sont composés d'une caméra surmontée d'un dispositif optique complémentaire permettant d'obtenir en une seule image une vision panoramique de l'environnement. Les systèmes existants sont :

les lentilles demi-sphériques : (figure 1.16) elles ont été introduites en 1986. Elles offrent une vue panoramique pour une surveillance sur un grand angle, par exemple pour la surveillance du fond d'une piscine. Par contre, elles ne permettent pas la détection d'obstacles car ceux-ci apparaissent en périphérie des images qui est de faible résolution.

le miroir conique : (figure 1.17) son utilisation a été proposée par Jarvis et Byrne en 1988 [Jar88], puis mis en œuvre par Yagi et son équipe en 1994 [Yag94]. Ce système est bien adapté à la détection de composantes verticales. La valeur de l'angle d'ouverture du cône de 90° offre le champ de vue le plus large possible. Dans ce cas la distance de l'objet à l'axe du cône n'influe pas sur sa représentation, ce qui permet des systèmes de localisation sans calibration de caméra.

le miroir à double courbure (figure 1.18) a été introduit plus récemment par Southwell et Basu en 1996, ce miroir est dédié à la reconstruction tridimensionnelle de scène et évite de projeter les objets sur de trop petites portions d'image [Sou96].

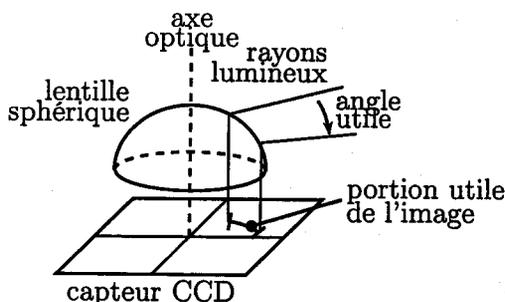


Figure 1.16 – Schéma de principe d'une caméra avec lentille sphérique.

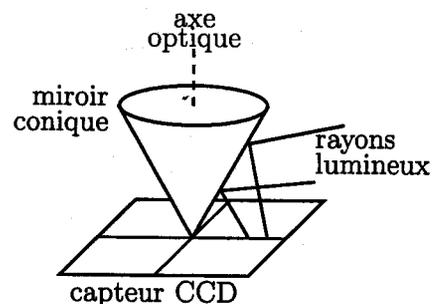


Figure 1.17 – Schéma de principe d'une caméra avec miroir conique.

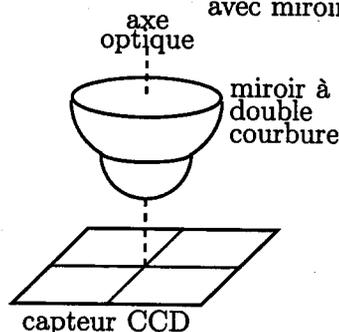


Figure 1.18 – Schéma de principe d'une caméra avec miroir à double courbure.

1.2 Représentation de l'environnement et interprétation des mesures

1.2.1 Modèles géométriques

La modélisation géométrique d'un environnement consiste à répertorier la position de composantes géométriques facilement identifiables à partir de données capteur acquises. Ces composantes sont décrites à l'aide de primitives géométriques qui peuvent être de différentes natures (droites, points, arcs de cercle,...) et représenter des informations différentes (limites de l'espace libre ou caractéristiques de l'environnement,...).

Ce type de modélisation n'est pas limité à une catégorie de capteurs. Elle convient aussi bien à des capteurs télémétriques qu'à des capteurs de vision. Dans [Cro89], l'auteur effectue une acquisition de l'environnement proche du robot à l'aide d'un télémètre à ultrasons rotatif et ensuite extrait du nuage de point acquis des segments de droite par une technique dite de regroupement et découpage itérative. Cette carte locale est ensuite comparée à un modèle global de l'environnement afin d'estimer la position du robot. Dans [Leo91] les auteurs effectuent également un balayage avec un système à ultrasons mais utilisent des régions de profondeur constante (RCD : Regions of Constant Depth, figure 1.19) pour construire une carte locale à partir des observations. Cette carte locale est ensuite comparée avec une prédiction de cette RCD locale afin d'estimer la position du robot.

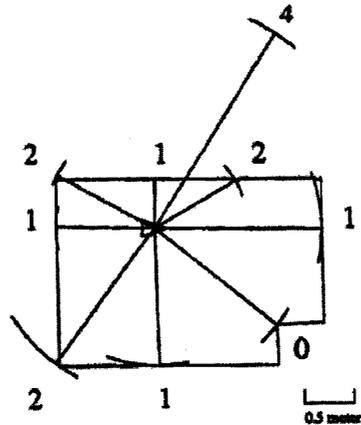


Figure 1.19 – Exemple de Régions de Profondeur Constante (RCD) d'après [Leo91].

Dans [Arr00], les auteurs utilisent un télémètre laser afin de détecter les limites de l'espace libre autour du robot qu'ils modélisent sous forme de droites [Arr97]. Ils utilisent également l'image fournie par une caméra CCD dont ils extraient les contours à l'aide d'un filtre de Sobel. Ils sélectionnent les lignes verticales de l'image du gradient qui sont caractérisées, dans le repère caméra, par leur angle de vue référencé par rapport à l'axe optique de la caméra. Le modèle local de l'environnement du robot est alors composé de lignes horizontales qui sont les frontières entre l'espace libre et l'espace occupé et des lignes verticales qui sont généralement situées aux intersections des murs. Cette carte locale est comparée avec une prédiction des observations et l'estimation de la position du robot est corrigée via un filtre de Kalman.

Dans [Mar99], les auteurs utilisent uniquement un capteur de vision. Ce capteur est un système omnidirectionnel, leur permettant d'avoir une image de l'environnement proche du robot sur 360°. Dans ce cas c'est un filtre de Deriche qui est utilisé afin d'extraire le gradient de l'image. Les droites sont ensuite caractérisées par leur angle dans l'image par rapport à la direction du robot. Les auteurs calculent ensuite le birapport (crossratio) pour chaque quadruplet de droites de l'image et identifient quatre droites avec leur position cartésienne respective dans le modèle global de l'environnement afin de calculer la position du robot sans estimation a priori de cette dernière.

L'intérêt des modes de représentation géométrique de l'information est la faible quantité de mémoire nécessaire à leur mémorisation. Ils requièrent cependant un traitement des données brutes qui peut être coûteux en temps d'exécution. Dans [Bor02b] les auteurs utilisent une représentation commune pour les lignes et les points. Ceci permet d'intégrer simultanément les deux types de primitives lors de l'estimation de la position du véhicule.

1.2.2 Grilles d'occupation

Une autre façon de représenter l'environnement est de le découper sous forme de grille où chaque cellule représente une portion de l'espace. A chaque cellule est associée une valeur qui représente sa probabilité d'occupation. Différents outils mathématiques peuvent être utilisés pour mettre à jour la grille, par exemple des méthodes bayésiennes dans [Mor85, Cho90] ou de la logique floue pour [Ori95, Fab00]. La figure 1.20 donne un exemple de grille d'occupation construite en utilisant une approche bayésienne. Cette figure, extraite de [Wij01], représente la trajectoire suivie par le robot ainsi que les contours réels de l'environnement (un salon réel reconstitué au CAS³) et la grille d'occupation construite à partir des mesures fournies par les capteurs à ultrasons. Les cellules occupées sont en sombre les cellules vides en blanc.

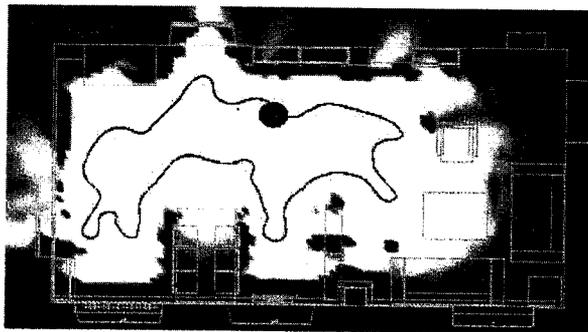


Figure 1.20 – Exemple de grille d'occupation bayésienne extrait [Wij01].

Un avantage de l'approche par grille d'occupation comparé à une représentation géométrique est énoncé dans [Hag90] : *les méthodes par grilles sont seulement une représentation approximative mais sont moins sensibles aux hypothèses faites sur le système de perception*, c'est-à-dire que le fait de discrétiser l'environnement permet d'utiliser un modèle du capteur qui sera susceptible d'intégrer l'incertitude relative aux mesures. Dans [Ras90] il est rapporté que les méthodes utilisant des modèles de grilles sont mieux adaptées à l'interprétation des mesures que les modèles utilisant des représentations géométriques lorsque le capteur utilisé est un capteur à ultrasons.

Moravec et Elfes, [Mor85], ont été parmi les premiers à populariser les techniques de grilles d'occupation. Comme les auteurs le font remarquer : *une mesure de distance contient peu d'information*, ce qui signifie que la seule façon d'être capable de construire une carte de l'environnement ou de trouver la position du robot est de combiner plusieurs lectures.

Un des inconvénients des grilles d'occupations présentées par Moravec et Elfes est le coût calculatoire des routines de mise à jour de l'information. Dans [Bor89, Bor91a] les auteurs utilisent un principe plus simple de mise à jour de la grille qui consiste à ne considérer que l'axe acoustique du capteur à ultrasons. Cela permet de réduire considérablement le temps de traitement, mais diminue l'acuité de la représentation.

1.2.3 Cartes topologique et vecteurs d'informations

La carte topologique peut être construite sans aucune connaissance des caractéristiques métriques de la pièce. Elle dépend uniquement de la structure de l'environnement. Elle se compose de nœuds qui représentent des pièces comme par exemple la cuisine, la chambre, ou même des objets comme le lit, l'ordinateur, etc.. Ces nœuds sont reliés entre eux par des arcs. Les arcs définissent la structure de l'environnement et la position relative entre les différents éléments qui le composent. Ils permettent de déterminer comment atteindre un lieu (nœud) en partant d'un autre. Un des avantages principaux des approches topologiques est qu'elles sont compactes et nécessitent peu de mémoire pour les sauvegarder.

Dans [Cha85], les auteurs définissent le concept de place comme une unité qui peut être soit topologique soit fonctionnelle. Par unité topologique les auteurs entendent des lieux comme les couloirs, les pièces, alors que par unité fonctionnelle ils déterminent des appareils situés dans des pièces, comme par exemple une imprimante. Les différentes places sont liées entre elles par des connecteurs comme des portes, escaliers,

³Centre for Autonomous Systems, Stockholm, Sweden

ascenseurs. Le modèle topologique de l'environnement est un graphe connecté où les nœuds sont les places et les liens entre les nœuds sont les connecteurs.

Dans [Bro84], Brooks construit une carte topologique à partir d'informations visuelles. Il définit des ensembles qu'il nomme "freeways" (pour les espaces navigables tels que des couloirs) et "meadows" (pour les pièces). Comme précédemment les "freeways" correspondent aux liens et les "meadows" aux nœuds .

Le cas typique d'environnement intérieur est celui d'un environnement de bureau. Ce type d'environnement se compose de longs couloirs, quelquefois d'atriums, et de pièces. Les couloirs forment fréquemment une structure croisée (intersections orthogonales des couloirs) qui ont une orientation soit de haut en bas ou de gauche à droite, lorsque l'on considère une représentation en vue de dessus.

La figure 1.21, extraite de [Alt03], donne un exemple de carte topologique. Les lignes en tirets représentent les couloirs, les cercles situés à l'intersection des lignes en tirets et des lignes continues sont des points de passage entre les couloirs et les pièces qui sont représentées par les cercles situés sur l'extrémité du segment en trait plein.

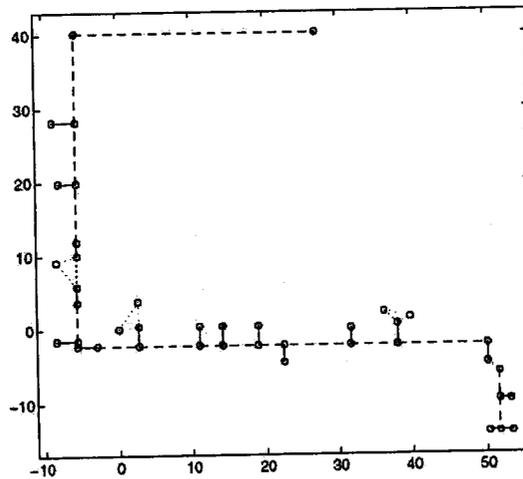


Figure 1.21 – Exemple de carte topologique extraite de [Alt03]

Dans [Wer00], puis dans [Lan03] les auteurs ajoutent des informations métriques à la représentation topologique de l'environnement et nomment cette variante Route Graph. Ces informations peuvent être la longueur des couloirs et la variation de direction lors d'un passage d'un couloir à un autre.

1.3 Méthodes de calcul de la position

1.3.1 Méthodes d'intégration

Ces méthodes intègrent au cours du temps les informations délivrées par les capteurs proprioceptifs. Le déplacement de la plupart des plate-formes de robotique mobile étant réalisé grâce à des roues motorisées, l'utilisation de codeurs pour mesurer la rotation de ces dernières est devenue plus ou moins un standard [Elf87, Pfi02]. Le terme communément utilisé pour nommer ce système est : odomètre. Habituellement, les codeurs ont une résolution élevée, et les plus couramment utilisés, sont les capteurs optiques incrémentaux. Comme nous l'avons vu à la section 1.1.1, ces codeurs optiques utilisent une source de lumière combinée à un photo-détecteur afin de percevoir les rotations d'un disque perforé. La précision de ce capteur est fonction du nombre de perforations sur le disque. Comme la plupart du temps ce nombre est important, la discrétisation est un problème seulement pour la détection de mouvements lents.

1.3.1.1 Modèles odométriques

Il existe différents modèles cinématiques selon le type de la plate-forme mobile. Ces différences influencent les performances de l'odomètre à une large échelle. Pour concevoir une plate-forme avec un odomètre performant, il est important de faire attention à certaines caractéristiques lors de la conception.

Nous présentons ici les deux types de plate-formes les plus courantes, avec leur odomètre correspondant et une étude de leur performance.

(a) **Plate-forme à mode différentiel** Ce type de plate-forme, [Leo92, Mar99, Cau02], utilise deux roues motrices indépendantes placées de chaque côté de cette dernière (figure 1.22). La rotation et le déplacement de la plate-forme sont réalisés en appliquant une vitesse de rotation différente sur les deux roues.

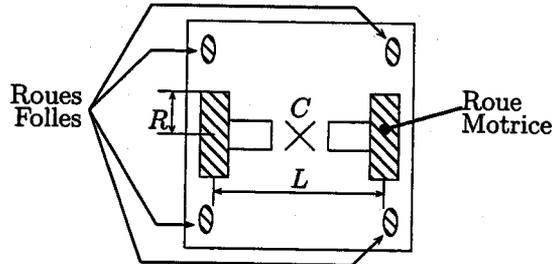


Figure 1.22 – Plateforme à conduite différentielle.

Généralement des roues folles sont également utilisées pour garantir l'équilibre de la plate-forme. Sur la figure 1.22 elles sont au nombre de quatre mais peuvent être moins nombreuses suivant la position des roues motrices. Si les roues motrices sont situées sur l'avant du robot, alors deux roues folles situées à l'arrière du robot sont suffisantes pour la stabilité mécanique du système.

Les amplitudes respectives des rotations des roues droite et gauche sont mesurées en positionnant des codeurs sur les axes des roues motrices ou des moteurs. Les distances parcourues par chaque roue, notées respectivement D_D et D_G pour la roue droite et gauche sont calculées connaissant leur diamètre R . La distance parcourue par le centre C du robot est alors donnée par la relation 1.11 :

$$D_C = \frac{D_D + D_G}{2} \quad (1.11)$$

Et la variation $\Delta\theta$ de l'orientation de la plate-forme est calculée par la formule 1.12 :

$$\Delta\theta = \frac{D_D - D_G}{L} \quad (1.12)$$

où L est la distance entre les deux points d'appui des deux roues motrices. L'orientation de la plate-forme est déterminée en cumulant les petites variations $\Delta\theta$. En supposant que les mouvements de la plate-forme peuvent être décomposés en une translation suivie d'une rotation, alors la position de cette dernière peut être calculée dans le repère global par :

$$x(k+1) = x(k) + D_C(k) \cos(\theta(k)), \quad (1.13)$$

$$y(k+1) = y(k) + D_C(k) \sin(\theta(k)), \quad (1.14)$$

$$\theta(k+1) = \theta(k) + \Delta\theta(k). \quad (1.15)$$

(b) **Plate-forme à mode synchrone** Ce type de véhicule, comme par exemple la plate-forme Nomade 200 [Nom03], est équipée de roues motrices tournant de façon synchrone. Chaque roue possède deux axes de rotation ; un axe permet le mouvement de translation alors que l'autre est utilisé afin d'orienter la roue. Le mouvement des roues est synchronisé par l'utilisation de chaînes ou de courroies. La figure 1.23 donne un exemple de configuration d'une plate-forme synchrone. La figure 1.24 montre comment s'oriente la roue en faisant varier l'angle φ . La translation du véhicule est réalisée par la rotation de l'axe intérieur (angle α). Les roues peuvent être orientées sans que la plate-forme ne tourne. Cela permet de garder idéalement constante l'orientation de la plate-forme dans le référentiel global (angle θ).

Le modèle odométrique d'une plate-forme synchrone est relativement simple car la direction de déplacement est directement donnée par l'orientation des roues dans le repère global, φ .

$$x(k+1) = x(k) + D(k) \cos(\theta(k) + \varphi(k)), \quad (1.16)$$

$$y(k+1) = y(k) + D(k) \sin(\theta(k) + \varphi(k)), \quad (1.17)$$

$$\theta(k+1) = \varphi(k) \quad (1.18)$$

Dans les équations 1.16 et 1.17, $D(k)$ est la distance parcourue au cours du k^{ieme} échantillon et $\varphi(k)$ est l'angle correspondant. En utilisant les notations introduites à la figure 1.24, nous obtenons :

$$D(k) = r\Delta\alpha(k) \tag{1.19}$$

La possibilité de modifier la direction de déplacement de la plate-forme sans effectuer de translation rend les plate-formes synchrones holonomes (ou quasi-holonomes).

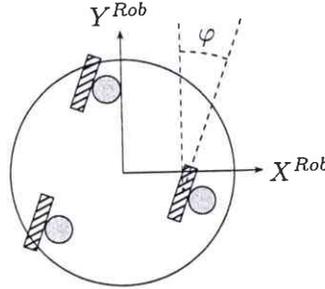


Figure 1.23 – Configuration d'une plate-forme à conduite synchrone.

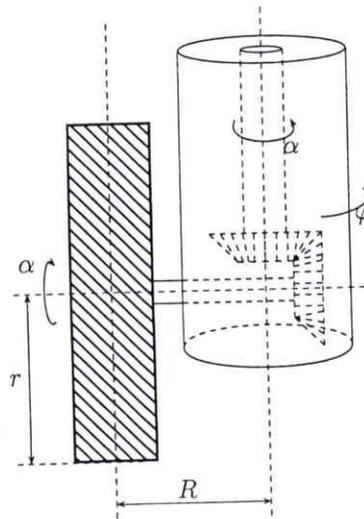


Figure 1.24 – Configuration d'une roue d'une plate-forme à conduite synchrone.

1.3.1.2 Conclusion sur l'odomètre

L'odomètre, quelque soit le type de plate-forme utilisé, présente l'avantage d'être peu coûteux et simple à mettre en œuvre. Les modèles odométriques présentés ci-dessus, qui supposent un comportement composé d'une translation suivie d'une rotation, ne sont que des approximations des cinématiques réelles. Mais, étant donné que la fréquence d'échantillonnage est élevée, l'approximation est assez bonne. Par contre, une petite erreur en orientation θ conduira à de fortes erreurs en position. Ces erreurs en orientation apparaîtront après avoir appliqué l'équation 1.15 plusieurs fois, en raison de l'imperfection du modèle (incertitude sur le rayon des roues, largeur de voie, etc.). En étant précis lors de la modélisation, ces erreurs peuvent être minimisées. Par contre, les erreurs non-systématiques ne sont pas prises en compte par le modèle et conduiront à des divergences irréversibles. Les sources de telles erreurs sont le glissement des roues ou les irrégularités du sol ; ces perturbations vont engendrer un écart entre la rotation des roues et son effet sur le changement de la localisation réelle du robot ce qui induit une erreur sur les variables $(\hat{x}, \hat{y}, \hat{\theta})$ estimant la position du robot.

1.3.2 Méthodes de classification

Ces méthodes consistent à mettre en correspondance des informations locales avec celles issues de positions connues de l'environnement. Ces méthodes sont fréquemment liées à des représentations de

l'environnement sous forme topologique, mais des représentations géométriques peuvent également être utilisées.

Dans [Ulr00], les auteurs proposent une méthode dans laquelle le robot est conduit à travers l'environnement, au cours d'une phase d'apprentissage, et prend des images avec une caméra omnidirectionnelle couleur toutes les secondes. L'utilisateur référence les images avec la position d'où elles ont été respectivement prises. L'indexation des images est relativement facile en raison de la séquentialité des acquisitions. Au lieu d'utiliser les images brutes, 6 histogrammes en une dimension sont extraits des images (1 par composantes HSV (Hue, Saturation and Value) et RGB (Red, Green and Blue)). Les histogrammes ont l'avantage d'occuper moins de mémoire que les images brutes et sont de plus invariants en rotation. Ces histogrammes composent ce que nous pouvons appeler un vecteur d'information permettant de caractériser l'image. Chaque image est associée à un lieu connu de l'environnement qui est un nœud de la carte topologique. Lorsque le robot parcourt en mode autonome l'environnement, et lorsqu'il veut corriger l'estimation de sa position, il effectue l'acquisition d'une image, puis la caractérise suivant le même principe que lors de la phase d'apprentissage. Les histogrammes de l'image courante sont mis en correspondance avec les histogrammes de référence. Chaque histogramme propose un vote. Les sorties de l'algorithme sont une classification de l'image et une mesure de confiance de la classification. La connaissance de l'estimation odométrique de la position du robot limite la plage des histogrammes de référence qu'il est nécessaire de prendre en compte lors de la classification.

Dans [Zho03] nous retrouvons un travail similaire à celui développé dans [Ulr00] qui est une décomposition de l'image en six histogrammes, trois selon les composantes RGB et trois autres selon HSV. Contrairement à la méthode précédente les auteurs utilisent une image issue d'une caméra simple, et non pas panoramique. Ici encore, il est nécessaire de faire parcourir une première fois l'environnement au robot, afin qu'il puisse acquérir des images de référence et ainsi constituer une base de connaissance des positions dans l'environnement qui est représenté par une carte topologique (phase d'apprentissage).

Les auteurs font remarquer que la méthode souffre de plusieurs problèmes. L'histogramme couleur peut juste exprimer l'information globale sur une image sans décrire les rapports existant entre les pixels, c'est-à-dire qu'il donne peu d'information spatiale sur les pixels : il donne la présence d'un niveau de couleur sans donner sa répartition dans l'image. Ainsi, le plus grand inconvénient de l'histogramme des couleurs est le risque d'associer deux images qui sont très différentes mais qui partagent un histogramme de couleurs similaire. La figure 1.25 montre un exemple de mauvais appariement de l'image courante (input image) et de l'une des images de la phase d'apprentissage.

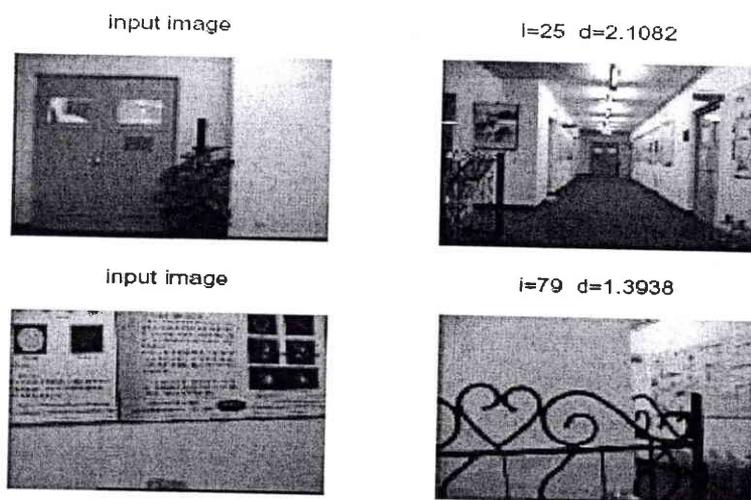


Figure 1.25 – Exemple d'échec d'appariement par histogramme des couleurs dans [Zho03].

Ce manque d'informations est compensé par l'intégration de données supplémentaires dans l'histogramme de l'image. C'est à dire que pour chaque pixel les auteurs considèrent en plus de la couleur la dynamique de l'image (Gradient et nombre de pixels d'intensité inférieure dans un voisinage donné) et la texture (nombre de contours, texture au voisinage du pixel). Ils obtiennent ainsi un histogramme à plusieurs dimensions pour chaque image de la base de connaissance.

Un exemple en situation réelle est donné dans cet article. Les auteurs utilisent un environnement composé de 14 pièces modélisées à l'aide d'une carte topologique. Chaque nœud représente une pièce d'environ 4 à 6 m^2 . Ils utilisent 810 images pour la base de connaissance. Ils ont fait l'acquisition de 340 images de test et ont testé 5 combinaisons différentes d'utilisation des fonctions d'apparence. Le taux de localisation correcte avec l'utilisation de l'histogramme des couleurs est de 36.7%, il passe à 61.1% dès l'utilisation d'un histogramme à deux dimensions pour atteindre 82.9% lorsque l'ensemble des informations supplémentaires sont utilisées. La figure 1.26 donne les résultats obtenus avec la méthode des histogrammes multidimensionnels comparée aux résultats obtenus avec la première méthode qui sont présentés sur la figure 1.25.

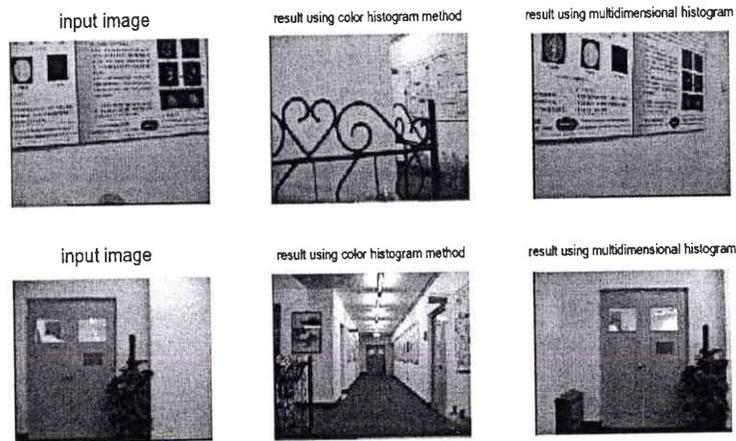
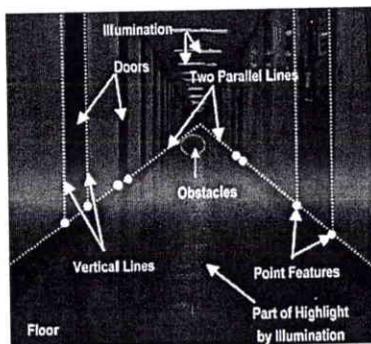
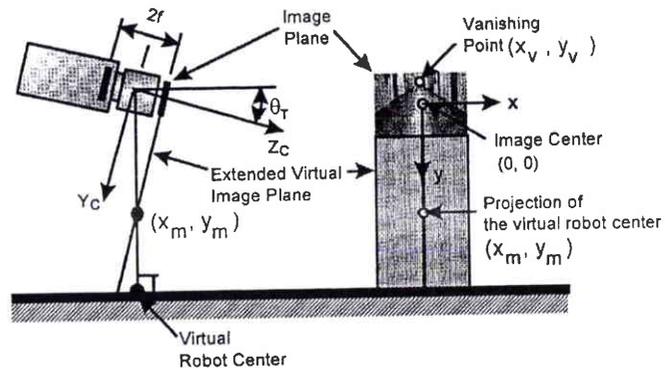


Figure 1.26 – Résultats obtenus avec la méthode des histogrammes multi-dimensionnels dans [Zho03].



(a) Acquisition dans un couloir



(b) Centre Virtuel du robot et sa projection sur le sol

Figure 1.27 – Informations utilisées dans [Lee00]

Dans [Roh97] et [Lee00] les auteurs proposent un système de localisation utilisant les invariants de projection [Mun92] afin de déterminer la position et l'orientation du robot sans connaissance a priori de celle-ci. Leur méthode ne nécessite pas la connaissance des paramètres de la caméra comme la distance focale et le facteur d'échelle. Leur système est principalement destiné à la navigation dans des couloirs. La figure 1.27(a) montre l'image d'un couloir qui peut être observé dans un environnement d'intérieur. On y repère deux lignes 'parallèles' (dans l'environnement réel) qui se croisent (dans l'image) en un point appelé point de fuite (vanishing point : caractéristique représentative de la géométrie projective) ainsi que des lignes verticales formées par des montants de portes. Les auteurs utilisent les intersections entre les deux parallèles et les verticales comme caractéristiques ponctuelles pour la localisation (marques blanches présentes sur la figure 1.27(a)). Le système d'auto-localisation nécessite la construction d'une base de données faisant office de modèle, cette base est construite hors-ligne. C'est une table indexée par l'invariant de projection calculé pour quatre points. Lors de la mise en correspondance des données

courantes avec le modèle, le système utilise l'index pour appairier les données image avec celles du modèle. Lors de l'auto-localisation, le centre virtuel du robot est défini par la projection verticale du centre optique de la caméra, montée sur le robot, sur le sol (figure 1.27(b)). Sur la figure 1.27(b) la projection perspective du centre virtuel du robot produit un point image virtuel (x_m, y_m) sur le plan image virtuel étendu. Ce point image virtuel devient la cinquième caractéristique ponctuelle dans l'image, qui est utilisée par l'auto-localisation.

Dans [Hay01], les auteurs utilisent la perception visuelle des amers de type plan ayant au moins deux caractéristiques horizontales et deux autres verticales (figure 1.28). Les positions et les tailles de ces amers sont apprises lors d'une phase d'apprentissage. Deux types de localisation sont utilisés, un premier qui se base sur une représentation topologique de l'environnement et un second sur une représentation géométrique. A chaque nœud de la carte topologique sont associés plusieurs amers perceptibles. Des essais en conditions réelles ont conduit à une erreur moyenne en position de 2 cm (6 cm au maximum) et une erreur moyenne en orientation de 2° (3.5° au maximum).

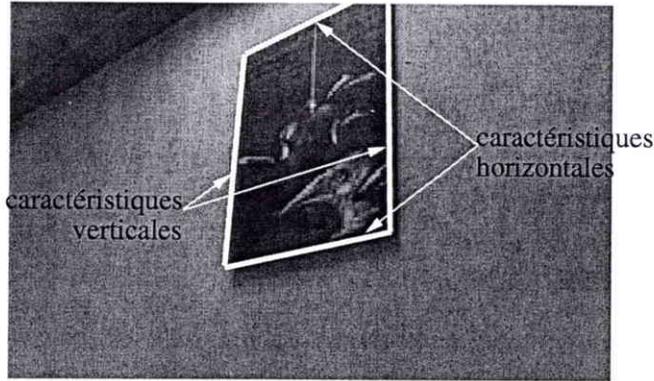


Figure 1.28 – Modélisation d'un amer d'après [Hay01]

1.3.3 Méthodes fondées sur le filtre de Kalman

C'est une méthode couramment utilisée lorsque la représentation de l'environnement est faite à l'aide d'une carte géométrique et qu'une estimation a priori de la position du véhicule est connue, via un odomètre par exemple.

1.3.3.1 Principe

Dans la suite, nous supposons que $X(k) = [x_k, y_k, \theta_k]$ représente l'état du système à l'instant k , c'est à dire sa localisation dans le cas d'un robot mobile.

Le filtre de Kalman se base sur deux modèles, un modèle d'évolution et un modèle de mesure. Le modèle d'évolution décrit comment la position du véhicule $X(k)$ évolue au cours du temps en réponse à une entrée de commande $u(k)$ et d'une perturbation $v(k)$.

$$X(k+1) = F(X(k), u_k) + \mu(k), \quad \text{avec } \mu(k) \approx \mathcal{N}(0, Q(k)) \quad (1.20)$$

où $F(X(k), u_k)$ est une fonction non linéaire de l'évolution de l'état du système. La notation $\mu(k) \approx \mathcal{N}(0, Q(k))$ indique que le bruit $\mu(k)$ est supposé gaussien de moyenne nulle et de variance $Q(k)$.

Le modèle de mesure donne une estimation de l'observation du capteur j en fonction de la position du véhicule et de la géométrie de la cible i devant être observée. Il s'écrit sous la forme suivante :

$$Z_j(k) = h_{ij}(p_i, X(k)) + w_j(k), \quad \text{avec } w_j(k) \approx \mathcal{N}(0, R_j(k)) \quad (1.21)$$

où la fonction $h_{ij}(p_i, X(k))$ exprime une mesure observée $z_j(k)$ en fonction de la position du véhicule $X(k)$ et la position de la cible p_i à l'instant k . Cette observation est supposée perturbée par un bruit gaussien de moyenne nulle et de variance $R_j(k)$. La forme de la fonction d'observation $h_{ij}(\cdot, \cdot)$ est liée au capteur utilisé et au type de cible observée.

L'objectif du filtre de Kalman est de produire une estimation optimale de la position du robot $\hat{X}(k+1|k+1)$ à l'instant $k+1$ en se basant sur l'estimation $\hat{X}(k|k)$ l'entrée de commande $u(k)$ et les nouvelles observations $z_j(k+1)$. Cet estimé minimise la variance de l'erreur d'estimation.

L'algorithme de localisation se fait en quatre étapes :

1. Phase de prédiction : en utilisant le modèle d'évolution, ainsi qu'une connaissance de l'entrée de commande $u(k)$, une nouvelle prédiction de la position du robot à l'instant $k+1$ est calculée :

$$\hat{X}(k+1|k) = F(\hat{X}(k|k), u(k)) \quad (1.22)$$

La variance associée à cette prédiction de la localisation $P(k+1|k)$ peut alors être calculée :

$$\hat{P}(k+1|k) = \nabla F P(k|k) \nabla F^T + Q(k) \quad (1.23)$$

où ∇F est la jacobienne de $F(\cdot, \cdot)$ obtenue par sa linéarisation autour de l'état estimé $\hat{X}(k|k)$. Ensuite cette prédiction de la localisation du robot est utilisée pour effectuer une prédiction des observations de chaque cible géométrique p_i :

$$\hat{z}_j(k+1) = h_{ij}(p_i, \hat{X}(k+1|k)) \quad \text{avec } i = 1, \dots, N_k \quad (1.24)$$

2. Phase de mise en correspondance : durant cette phase les prédictions $\hat{z}_m(k+1)$ sont mises en correspondance avec les observations $z_n(k+1)$. C'est à ce niveau que les méthodes de localisation utilisant un filtre de Kalman diffèrent entre elles et font la particularité de la méthode.
3. Phase d'observation : l'étape suivante consiste à prendre un nombre d'observations $z_j(k+1)$ de ces différentes cibles et à les comparer avec leurs prédictions. La différence entre une prédiction $\hat{z}_j(k+1)$ et une observation $z_j(k+1)$ est le terme d'innovation. Il s'écrit :

$$\begin{aligned} \nu_{ij}(k+1) &= [z_j(k+1) - \hat{z}_j(k+1)] \\ &= [z_j(k+1) - h_{ij}(p_i, \hat{X}(k+1|k))] \end{aligned} \quad (1.25)$$

La covariance de l'innovation peut être déterminée par :

$$\begin{aligned} S_{ij}(k+1) &\equiv E[\nu_{ij}(k+1)\nu_{ij}^T(k+1)] \\ &= \nabla h_{ij} P(k+1|k) \nabla h_{ij}^T + R_j(k+1) \end{aligned} \quad (1.26)$$

où ∇h_{ij} est la jacobienne de h_{ij} et est évaluée en $\hat{x}(k+1|k)$ et p_i .

4. Phase d'estimation : c'est l'étape finale du filtrage de Kalman, c'est ici que les prédictions et observations correctement associées vont être utilisées pour calculer $\hat{X}(k+1|k+1)$. Le gain du filtre de Kalman $W(k+1)$ est calculé par :

$$W(k+1) = P(k+1|k) \nabla h^T S^{-1}(k+1) \quad (1.27)$$

pour ensuite mettre à jour la position estimée du véhicule, $\hat{x}(k+1|k+1)$, ainsi que sa variance associée, $P(k+1|k+1)$.

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)\nu(k+1) \quad (1.28)$$

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W^T(k+1) \quad (1.29)$$

1.3.3.2 Exemples d'application du filtre de Kalman

Dans [Leo91], les auteurs présentent une méthode de localisation en cours de mouvement destinée à un robot d'intérieur. Cette méthode utilise un odomètre pour estimer la localisation du robot et met à jour celle-ci en utilisant les distances fournies par le télémètre à ultrasons. L'environnement est modélisé à l'aide de segments de droites qui définissent les frontières entre l'espace libre et l'espace occupé. Deux configurations de télémètre sont testées dans cet article : un capteur à ultrasons tournant et six capteurs

fixes sur la base mobile. Lors de l'appariement des données avec les prédictions faites en utilisant le modèle, les auteurs utilisent des zones de validité autour des prédictions dans lesquelles doivent se situer les données. Ils utilisent uniquement les appariements qui associent une donnée avec une prédiction. Les auteurs rejettent toutes observations qui peuvent engendrer une ambiguïté (une donnée avec plusieurs prédictions ou une prédiction avec plusieurs données) afin de garantir la qualité de leur appariement.

Différents essais ont été réalisés pour les deux méthodes où le robot se trouvait dans un environnement simple avec de nombreux segments disponibles dans toutes les directions. Pour les deux méthodes ils montrent que lorsque le robot n'est proche que d'une surface plane, la seule partie visible de cette surface est la portion où le faisceau incident du capteur à ultrasons est quasiment perpendiculaire à la surface, ce qui permet une bonne correction dans la direction normale à la surface. L'avantage de la seconde méthode est que le temps d'acquisition des mesures capteur plus court lui permet d'être exécutée en ligne, alors que dans la première on doit stopper le véhicule pour effectuer les acquisitions, problème qui serait moins perceptible avec un capteur comme un télémètre laser rotatif. Par contre, la seconde méthode, au vu de sa configuration, est limitée à suivre des trajectoires presque parallèles aux murs. Au delà d'un écart d'environ 13° l'incertitude de l'estimation du véhicule augmente, mais retrouve une valeur faible dès lors que le robot est de nouveau parallèle à un mur.

Dans [Arr99] les auteurs présentent un système de localisation pour robot mobile basé sur l'utilisation d'un télémètre laser ainsi que d'une caméra CCD. Ils tirent avantage de la possibilité qu'offre le filtre de Kalman pour l'utilisation des sources de données multiples. Ils utilisent les informations fournies par l'odomètre comme vecteur d'entrée de l'équation d'évolution (cf. équation 1.20), soit $u(k) = [\Delta d_L, \Delta d_R]^T$ qui sont respectivement les distances parcourues par chacune des roues motrices. Les appariements entre les verticales de l'image et les verticales du modèle ainsi que les appariements entre les horizontales du laser et les horizontales du modèle sont réalisés en minimisant la distance de Mahalanobis. Ces appariements sont ensuite utilisés pour corriger la position estimée du véhicule. La figure 1.29, extraite de [Arr99] montre un exemple où trois verticales de l'image sont appariées avec celles du modèle ainsi que quatre horizontales du balayage laser avec leurs correspondantes dans le modèle.

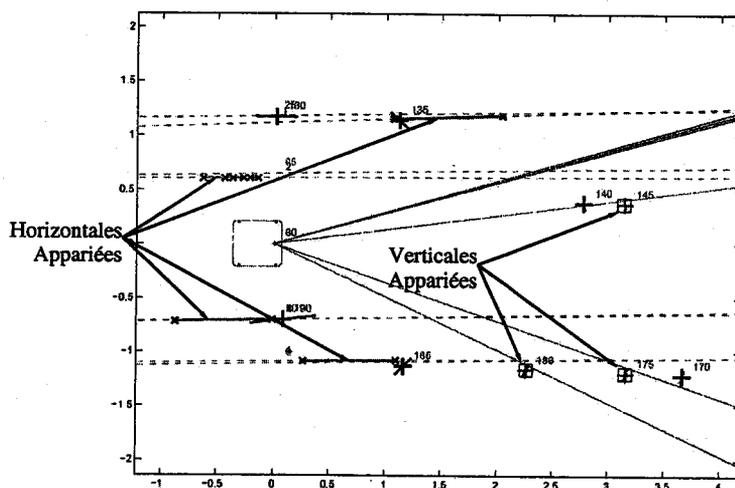


Figure 1.29 – Exemple de correction extrait de [Arr99].

Dans de [Arr00] et [Arr01], les auteurs n'utilisent plus la distance de Mahalanobis que pour les verticales. Pour la recherche des meilleurs appariements des horizontales, ils utilisent l'incertitude d'observation la plus petite comme critère de bon appariement. Un appariement, (\hat{z}_j, z_i) , est validé s'il satisfait le critère suivant :

$$(z_i - \hat{z}_j)S_{ij}^{-1}(z_i - \hat{z}_j)^T \leq \chi_{\alpha,n}^2 \quad (1.30)$$

où \hat{z}_j est la prédiction associée à la mesure z_i et S_{ij} est la matrice de covariance de l'innovation de l'appariement à tester et $\chi_{\alpha,n}^2$ une valeur prise d'une distribution χ^2 avec $n = 2$ qui est le degré de liberté et α le niveau au dessous duquel la validité de l'hypothèse d'appariement est rejetée.

Dans [Nei99], les auteurs présentent un système de localisation basé sur l'utilisation d'un EKF (Ex-

tended Kalman Filter). La prédiction de la position du robot est réalisée par les données fournies par l'odomètre. Un télémètre laser permet d'obtenir des images de profondeur et d'intensité dont sont extraites les informations utiles qui seront comparées à une carte de l'environnement connue a priori afin d'estimer plus précisément la position du robot.

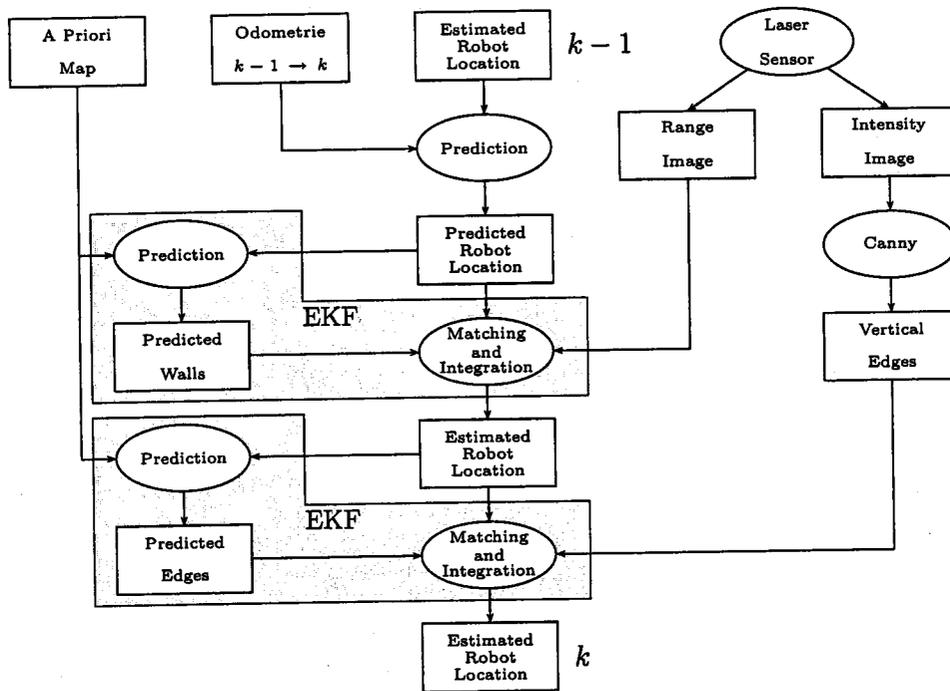


Figure 1.30 – Principe de fonctionnement du système de localisation présenté dans [Nei99].

La figure 1.30 donne une représentation des flux d'informations et de traitements de la méthode. Le capteur laser utilisé fournit d'une part une information de distance (comme le ferait un capteur à ultrasons) et d'autre part une information d'intensité lumineuse (comme pour une caméra). Après un balayage 2D (largeur + hauteur) de l'environnement situé devant le robot, le système obtient deux images, une de distance et une d'intensité lumineuse. Deux estimations de la position se succèdent. La première est réalisée en utilisant l'information issue de l'image de profondeur, alors que la seconde utilise l'information issue de l'image d'intensité. Il s'agit des contours verticaux présents dans l'image détectés par un filtre de Canny [Can83].

Le test utilisé pour valider la mise en correspondance des caractéristiques du modèle local avec celles du modèle global est :

- pour l'image de profondeur un test de compatibilité basé sur une distribution de probabilité χ^2 . De plus, si plusieurs caractéristiques observées sont appariées avec la même caractéristique du modèle, alors l'appariement donnant la plus petite distance de Mahalanobis est gardé comme candidat.
- pour l'image d'intensité un test de visibilité des caractéristiques du modèle est réalisé à partir de la position estimée du robot. Seuls les contours observés ayant un appariement unique sont gardés.

Les résultats obtenus au cours de leur essais ont été obtenus en moins de 1 seconde pour une précision de 10 cm en position et de 0.15° en orientation.

1.3.4 Méthodes fondées sur les moindres carrés

Tout comme le filtre de Kalman, ces méthodes sont très utilisées lorsque la représentation de l'environnement est fait à l'aide d'une carte géométrique et qu'une estimation a priori de la position du véhicule est connue.

1.3.4.1 Principe

L'objectif de la méthode des moindres carrés est de donner une estimation \hat{b} des paramètres d'un modèle b (b_0, \dots, b_k) qui minimise la somme S des carrés des erreurs de prédiction. Ces erreurs ε_i mesurent l'écart entre une sortie prédite par le modèle \hat{y}_i et une sortie réelle y_i , avec $i \in [1, \dots, n]$.

$$S = \sum_{i=1}^n \varepsilon_i^2 = \varepsilon^T \varepsilon \quad \text{avec } \varepsilon = Y - \hat{Y} \quad (1.31)$$

où $Y = [y_0, \dots, y_n]^T$ est le vecteur des mesures réelles et $\hat{Y} = Xb$ est une prédiction de ces mesures déterminée en utilisant X la matrice des entrées. Nous pouvons également écrire :

$$\begin{aligned} S &= (Y - Xb)^T (Y - Xb) \\ &= Y^T Y - 2b^T X^T Y + b^T X^T X b \end{aligned}$$

S est un scalaire qui dépend linéairement de b . Son minimum par rapport à b sera obtenu pour la valeur \hat{b} qui annule la dérivée de S par rapport à b .

$$\frac{\partial S}{\partial b} = \frac{\partial (Y^T Y - 2b^T X^T Y + b^T X^T X b)}{\partial b} = 2X^T Y - 2X^T X b \quad (1.32)$$

comme \hat{b} est tel que $\frac{\partial S}{\partial b} = 0$, alors :

$$-X^T Y + X^T X \hat{b} = 0 \quad (1.33)$$

d'où

$$X^T X \hat{b} = X^T Y \quad (1.34)$$

et

$$\hat{b} = (X^T X)^{-1} X^T Y \quad (1.35)$$

Appliquée à l'estimation de la localisation d'un robot mobile, le modèle b est un vecteur de dimension $[3 \times 1]$, $b = (x, y, \theta)^T$. Le vecteur Y est de taille $[N \times 1]$, où N est le nombre de mesures utilisées pour la localisation et la matrice X est de taille $[N \times 3]$. Le calcul de l'erreur ε_i entre une mesure réelle et une mesure prédite nécessite une mise en correspondance. Comme pour le filtre de Kalman, c'est le mode de représentation des données et la phase de mise en correspondance qui différencient les méthodes.

1.3.4.2 Exemples d'application des moindres carrés

Dans [Cox91], le robot Blanche est conçu pour opérer de façon autonome à l'intérieur d'un environnement structuré, comme un bureau ou un local industriel, sans adjonction de balises actives ou passives. La méthode de positionnement de Blanche utilise une carte a priori de l'environnement de travail, représentant les frontières entre l'espace libre et l'espace occupé, ainsi qu'une carte locale construite à partir des données fournies par un capteur infrarouge tournant. Les données de la carte locale sont représentées sous forme de points définis par leurs coordonnées polaires. Une estimation de la localisation du robot est fournie par l'odomètre.

L'algorithme de correction suppose que les tirs capteurs sont proches des objets réels (ou plutôt des lignes qui les représentent), c'est-à-dire que l'erreur odométrique est faible, en particulier l'erreur en orientation. Cette hypothèse permet de décomposer la transformation composée d'une rotation et d'une translation à appliquer à la localisation estimée en une série de petites transformations calculées itérativement jusqu'à convergence de la méthode, c'est-à-dire jusqu'à obtenir une transformation insignifiante. La phase de mise en correspondance cherche à apparier chaque point de la carte locale, avec le segment de droite de la carte globale le plus proche et détermine la distance qui les sépare. Ensuite la transformation et la rotation sont calculées en cherchant à minimiser la somme des distances au carré séparant les points de leur droite cible. Ces transformations sont alors appliquées à la localisation estimée et la procédure est relancée tant que les transformations sont de valeurs significatives.

Des essais de navigations dans un couloir sont présentés. Blanche naviguait à une vitesse de l'ordre de 5 cm/s . La déviation maximale sur la position calculée par rapport à sa position actuelle est donnée à 1.5 cm .

Dans [Hop98], une méthode de localisation pour une plate-forme mobile téléopérée est proposée. Un odomètre permet d'estimer les mouvements de la plate-forme. La perception de l'environnement situé autour du robot est réalisée à l'aide de huit capteurs à ultrasons répartis essentiellement sur l'avant du robot. Deux modes de localisation sont proposés. Ils utilisent tous les deux un modèle géométrique de l'environnement qui décrit les frontières entre l'espace libre et l'espace occupé à l'aide de segment de droites.

Le premier mode de localisation du robot utilise une seule lecture des 7 capteurs situés à l'avant du robot ainsi que l'estimation de la localisation fournie par l'odomètre. Chaque point d'impact de la carte locale (créé à partir des données des capteurs à ultrasons) est mis en correspondance avec le plus proche segment de la carte globale et la distance point/segments est déterminée comme suit. Si le point est compris dans une région A_0 située entre les deux extrémités du segment alors c'est la distance normale d_0 au segment qui est calculée (cas du point p_b), sinon c'est la distance à la plus proche extrémité qui est prise en compte : d_1 si le point se situe dans la zone A_1 et d_2 s'il se situe dans la zone A_2 (figure 1.31). Ensuite, la transformation à appliquer à la position estimée est calculée à l'aide d'un algorithme des moindres carrés minimisant la somme des distances au carré.

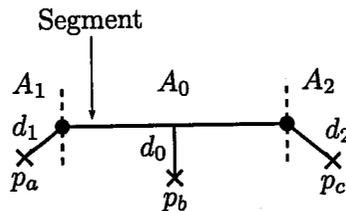


Figure 1.31 – Les trois zones définies pour un segment.

Le second mode de localisation utilise les dix dernières mesures obtenues par chacun des 7 capteurs afin de construire des droites. La construction des droites se déroule de la manière suivante : lors d'une lecture des 7 capteurs à ultrasons une carte locale est construite et chaque point est apparié à un segment de l'environnement de la même manière que dans la première méthode. Lorsque plusieurs points sont appariés à un même segment alors une droite peut être déterminée, c'est la droite mesurée. Lorsque dix acquisitions ont été réalisées et que les droites mesurées sont disponibles, la correction peut être effectuée. Elle se déroule en deux temps. Premièrement, seule l'orientation est corrigée à partir de la moyenne des différences entre les orientations des droites mesurées et les droites du modèle. Puis la position est corrigée en minimisant la somme des distances séparant les centres des segments mesurés des segments du modèle auxquels ils sont appariés. L'algorithme de minimisation des moindres carrés utilisé est le même que celui de la première méthode de localisation.

Récemment, dans [Bor02b], le système proposé par les auteurs utilise deux cartes géométriques, une carte globale \mathcal{M} et une carte locale \mathcal{M}' . Ces deux cartes se composent de caractéristiques géométriques hétérogènes telles que des points qui correspondent à une modélisation de coins ou de bords verticaux présents dans l'environnement. Par coins, les auteurs entendent l'intersection de deux murs et par bord la fin d'un mur par exemple. Les cartes incluent également des caractéristiques géométriques de type lignes représentant les murs et les côtés d'objets polygonaux. La carte locale \mathcal{M}' leur permet également de maintenir à jour la carte globale \mathcal{M} de l'environnement.

Les caractéristiques sont représentées à l'aide d'un vecteur $f = (x, y)^T$ dans le système de coordonnées global et par $f' = (x', y')^T$ dans le référentiel local. La figure 1.32 donne une représentation des paramètres utilisés pour la représentation de ces différentes caractéristiques. Les auteurs utilisent donc :

- Caractéristique Point : $(x, y)^T = (x_p, y_p)^T$,
- Caractéristique Ligne : $(x, y)^T = (x_l, y_l)^T$, avec $(x_l, y_l)^T$ les coordonnées du point L de la ligne, tel que $x_l = \rho \cos(\alpha)$ et $y_l = \rho \sin(\alpha)$

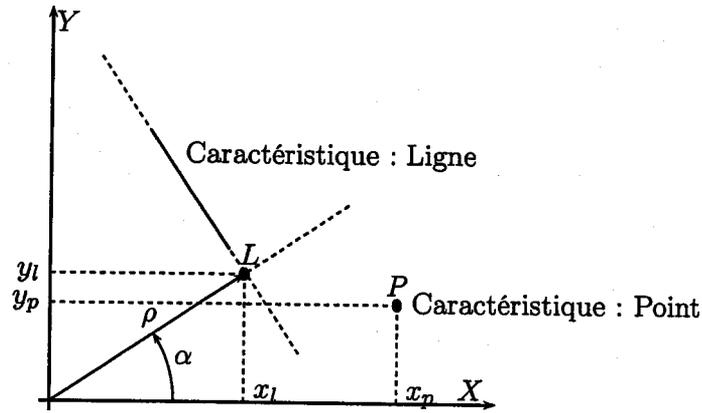


Figure 1.32 – Représentation des paramètres des caractéristiques dans le plan 2D, d'après [Bor02b]

L'apport de ce modèle unique est de permettre la définition d'une fonction de transformation uniforme du repère global au repère local des paramètres d'une caractéristique, à l'aide de :

$$f' = R(\hat{\theta}) \{f - E.\hat{t}\} \quad (1.36)$$

où f' et f sont respectivement les coordonnées de la caractéristique dans le repère local et global. $R(\hat{\theta})$ est la différence entre les axes des x des deux référentiels, $\hat{\theta}$ est l'orientation estimée du robot, \hat{t} est la position estimée du robot, ou encore la translation entre les deux référentiels. Enfin E est la matrice définie comme suit :

$$E = \begin{pmatrix} a^2 & c \\ c & b^2 \end{pmatrix} \quad (1.37)$$

où les entrées a , b et c dépendent du type de caractéristique considérée, telle que pour un point $a = b = 1$ et $c = 0$, et pour une ligne : $a = \cos(\alpha)$, $b = \sin(\alpha)$, et $c = ab$. En utilisant les représentations uniformisées des données, ainsi que la transformation homogène aux différentes caractéristiques, les auteurs présentent le problème de la localisation du robot comme une recherche de la solution d'un système au sens des moindres carrés pondérés. Dans [Bor02b], les auteurs supposent que les correspondances entre l'ensemble des caractéristiques du modèle \mathcal{F} et l'ensemble des caractéristiques de la carte locale \mathcal{F}' sont connues. Leur méthode d'estimation de la localisation du robot se décompose en deux phases. Dans un premier temps ils recherchent toutes les positions et orientations qui minimisent une fonction coût. Puis ils retiennent parmi les solutions pour $\hat{\theta}$ (l'orientation estimée) seulement les minima qui satisfont $\partial^2 J(\hat{z}) / \partial \hat{\theta}_i^{*2}$, où $\hat{\theta}_i^{*2}$ est l'orientation à évaluer. Si après cette seconde phase il reste deux solutions possibles, alors ils gardent l'estimation qui est la plus proche de l'estimation prédite. Une évaluation hors ligne de ce système d'estimation de la position est également présentée. L'objectif de cette expérimentation est de comparer les capacités de leur algorithme face à un estimateur basé sur un filtre de Kalman itératif, un filtre de Kalman non récursif. Ils utilisent comme trajectoire de référence celle relevée à l'aide du système embarqué sur le robot qui combine un odomètre et un gyromètre. Les capteurs utilisés pour la perception de l'environnement local sont un télémètre laser et une caméra CCD [Bor02a]. Dans [Bor02a] les auteurs présentent plus en détail le principe de mise en correspondance des caractéristiques ainsi que la mise à jour de la carte globale.

1.3.5 Méthodes probabilistes

La localisation probabiliste est le processus de détermination de la possibilité que le robot se trouve à une position donnée, étant donnée une certaine perception de l'environnement local et des actions effectuées. Dans [Gut02b], les auteurs font remarquer que le filtre de Kalman est souvent utilisé pour des méthodes de correction de la position, mais ne permet pas de considérer plusieurs hypothèses de localisation, ce qui rend impossible le recouvrement de la position après un échec total de la méthode de localisation. Ils proposent l'utilisation d'une méthode combinant une localisation Markovienne et un filtre de Kalman (ML-EKF [Gut02a]) (figure 1.33). L'idée de base est d'utiliser la localisation Markovienne pour une recherche globale et rapide de la position du robot sans nécessiter d'indication complémentaire

et avec une bonne immunité au bruit capteur. Le filtre de Kalman est quant à lui utilisé localement pour un calcul plus précis de la position du véhicule.

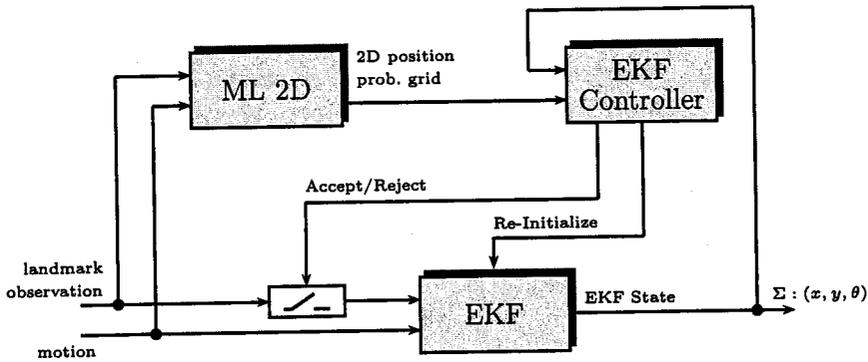


Figure 1.33 – Principe du système de localisation combinant une localisation Markovienne et un filtre de Kalman, d'après [Gut02b]

La grille markovienne est de résolution grossière et représente les positions possibles du robot sans information sur l'orientation. Sur la figure 1.33, on peut voir que les informations du mouvement sont transmises à la fois au filtre de Kalman et à la localisation markovienne. Mais seules les observations ayant validées un seuil de probabilité d'observation déterminé par le "EKF controller" sont transmises au filtre de Kalman. Celui-ci peut alors affiner l'estimation de la position et de l'orientation du robot. Dans [Gut02a], l'auteur compare le comportement de la méthode ML-EKF avec une méthode de localisation markovienne et deux versions de méthode de localisation utilisant un filtre de Kalman. Il conclut alors que la méthode développée hérite de la précision du filtre de Kalman et de la robustesse ainsi que de la vitesse d'exécution des méthodes de localisation markovienne.

Lors de la localisation markovienne il est nécessaire de déterminer la distribution de probabilité sur la localisation du robot. Dans [Gut02b] les auteurs évaluent trois manières de mettre à jour la distribution de probabilité sur la localisation. Ces techniques ont été développées respectivement dans [Len00] (SRL se base sur la vraisemblance des observations actuelles pour déterminer le nombre d'échantillons à ajouter), [Thr00b] (Mix-MCL, dans la version utilisée par les auteurs, ajoute un nombre constant de nouvelles positions possibles, car développée initialement pour un capteur très précis) et [Cri03] (A-MCL, pour MCL adaptatif, se rapproche de la méthode SRL). La méthode A-MCL est une évolution de la méthode de localisation de Monte-Carlo (MCL) développée dans [Fox99]. L'idée est de représenter la distribution de probabilités de la localisation par un ensemble de couple $\langle l_i, w_i \rangle$. Les l_i sont des positions possibles du robot et les w_i sont des poids associés à chaque l_i dont la somme vaut 1. Suite à une action a_n et une information de mesure s_n , chaque distribution de probabilité sur les différents l_i est re-généré ainsi que le poids associé w_i . Durant cette phase de nouvelles possibilités sont ajoutées. Dans leur travail, les auteurs ont mis en œuvre trois manières d'ajouter des positions possibles.

Les auteurs ont donc évalué les différentes versions de leurs méthodes de localisation. Les versions ML-EKF, SRL, A-MCL donnent des résultats sensiblement identiques au vu du temps de calcul, de l'immunité au bruit et de l'erreur en distance. Ils constatent une immunité au bruit légèrement meilleure pour la version A-MCL.

Dans [Lan02, Lan03], le système de localisation d'un fauteuil roulant autonome utilise une représentation hybride de l'environnement qui est une carte topologique métrique (figure 1.34(a)), c'est à dire qu'elle inclut une information de distance entre deux nœuds (longueur des couloirs) et de direction entre deux arcs successifs (changement de direction lors d'un passage d'un couloir à un autre).

Pour la localisation du fauteuil, les auteurs ne mettent pas en correspondance un ensemble de nœuds connectés à l'aide d'arc, mais utilisent une entité qu'ils nomment jonction et mettent donc en correspondance les jonctions reconnues lors d'un déplacement avec les jonctions contenues dans la carte topologique. Une jonction (figure 1.34(b)) est définie comme suit :

$$j := (H, T, \gamma, o, I) \quad (1.38)$$

où H est le point de départ de la jonction j , T son point d'arrivée (ce sont des nœuds du graphe topolo-

gique), o est la longueur du couloir qui relie H et T , I est l'ensemble de toutes les jonctions j'_i conduisant à la jonction j et γ_i est l'angle formé entre le segment de la jonction j'_i et le segment de la jonction j . Autrement dit c'est la valeur du changement de direction que doit exécuter le robot pour pouvoir entrer dans un nouveau couloir.

Pour la localisation le chemin parcouru est comparé avec le modèle de l'environnement. Pour cela ils définissent une route dans le graphe qui est l'ensemble des jonctions parcourues, c'est un historique des jonctions par lesquelles est passé le robot. Ce chemin est ensuite comparé avec le modèle de l'environnement en utilisant une approche probabiliste afin de déterminer le chemin sur lequel le robot a le plus de chance de se trouver.

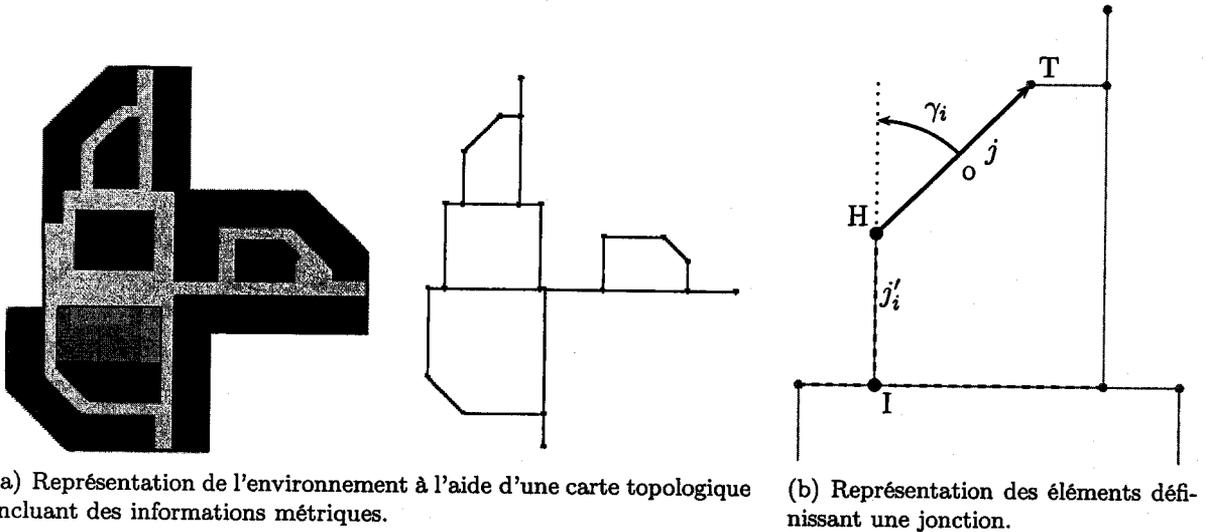


Figure 1.34 – Modélisation de l'environnement dans [Lan03].

1.3.6 Autres méthodes

Bien que de nombreuses recherches visent à développer des systèmes n'utilisant aucune modification de l'environnement certains auteurs conduisent des travaux utilisant des balises artificielles placées dans l'environnement. Ces systèmes donnent de bons résultats mais limitent la zone de travail du véhicule à celle couverte par les balises.

Dans [Hal01], les auteurs utilisent un véhicule équipé de 16 capteurs à ultrasons, 16 capteurs à infrarouge et une caméra en niveau de gris. L'environnement est représenté à l'aide d'une grille qui est construite à partir des informations fournies par les capteurs. La carte est construite itérativement et une méthode floue est utilisée pour la fusion des données. Des points de repère visuels constitués par des balises artificielles sont utilisés pour calculer la position. L'objectif de la méthode de localisation est de mettre à jour la position et l'orientation du robot en utilisant des points de repères artificiels dont les tailles, formes et positions sont connues. La localisation est basée sur l'analyse d'une image. Les images sont fournies par la caméra à niveaux de gris qui est installée au dessus du robot. La carte de l'environnement est une grille d'occupation, dont l'état des cellules est déterminé par la fusion des données des capteurs ultrasons et infrarouges. L'état des cellules peut être inconnu, occupé, ou libre. Cette carte n'est pas utilisée pour la localisation mais pour d'autres tâches comme la planification. En revanche elle nécessite une bonne connaissance de la position du robot pour sa construction. La cible utilisée a une forme de 'H'. La distance entre les deux parallèles verticales du 'H' est connue, la caméra est calibrée. A partir de l'image, deux angles de vue sont calculés, un pour chaque partie verticale. Connaissant la position des parties verticales, les auteurs effectuent une triangulation afin de retrouver la position et l'orientation du robot.

Dans [Her03], les auteurs proposent un système n'utilisant pas de carte de l'environnement, ni d'estimation a priori de la position du véhicule. La méthode de localisation utilise un ensemble constitué d'un laser fixé sur un mur de l'environnement et d'une ceinture de récepteurs photosensibles embarquée sur le robot. L'émetteur effectue un balayage à vitesse constante. La méthode de localisation nécessite la

détermination des paramètres comme le temps absolu mesuré lorsque le faisceau laser commence à balayer les cellules photosensibles de la ceinture, le numéro des cellules couvertes, le temps mis par le laser pour balayer la totalité des cellules visibles. Elle définit la localisation du robot en coordonnées polaires (r, θ, α) par rapport à la position du laser sur le mur. Comme la ceinture est composée de cellules photosensibles de taille fixe, le nombre de cellules percevant le faisceau laser dépend de la distance entre le robot et la cible ; plus le robot sera éloigné de la cible, moins de cellules seront couvertes. Ainsi l'angle d'ouverture β (figure 1.35(b)) est fonction de la distance r . Connaissant le diamètre de la ceinture de cellules, l'algorithme peut ainsi déterminer la distance entre le robot et le laser (origine du repère).

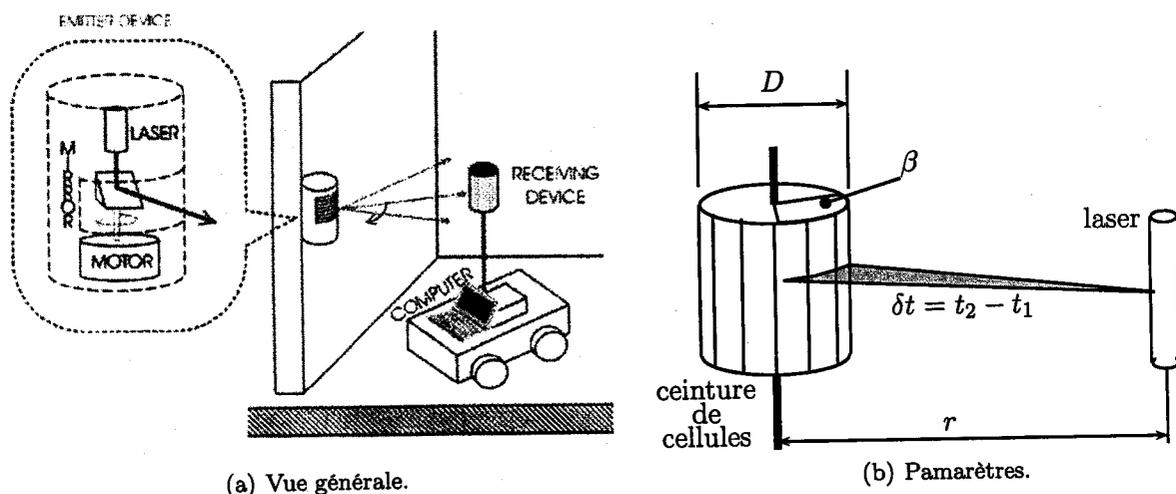


Figure 1.35 – Système décrit dans [Her03].

Ils proposent ensuite deux méthodes de détermination de l'angle θ , suivant l'existence ou non d'une synchronisation entre l'émetteur et le récepteur. La synchronisation se fait par l'ajout d'un couple émetteur récepteur FM qui permet de signaler au robot le début d'un balayage laser. Enfin ils déterminent l'orientation α en identifiant la cellule normale au balayage laser. Les essais présentés donnent de bons résultats mais la localisation du robot est limitée à une zone de l'environnement. De plus il n'est pas possible de tenir compte de masquages partiels de la cible.

1.3.7 Conclusion

Nous avons pu voir dans ce qui précède diverses méthodes de localisation. Elles se singularisent par les capteurs utilisés, le choix de représentation des mesures et de l'environnement dans lequel évolue le véhicule ainsi que la mise en correspondance des mesures avec le modèle. La plupart des méthodes ayant été développées dans un contexte particulier de nombreuses solutions existent pour différentes situations.

Le domaine de l'assistance aux personnes handicapées moteur connaît depuis ces 20 dernières années un intérêt grandissant. Ainsi de nombreuses équipes de robotique ont progressivement dirigé leur thématique de recherche vers l'aide à la conduite de fauteuil roulant électrique, où vers la conception de plate-formes d'assistance à des personnes sévèrement handicapées. Notre travail se situe dans ce contexte et la suite de ce chapitre est destiné à présenter quelques plate-formes existantes ainsi que leur mode de perception.

1.4 Robotique mobile pour personnes handicapées : méthodes de localisation

1.4.1 Introduction

L'une des premières chaises roulantes fut fabriquée vers 1595 pour le roi Philippe II d'Espagne, âgé alors de 68 ans. C'était une chaise montée sur quatre roues, avec des accoudoirs, un repose pieds et un

dossier ajustable. Mais l'origine de la chaise roulante moderne remonte à 1933. L'ingénieur Herbert A. Everest, qui était devenu paraplégique en 1919 à la suite d'un accident minier, voulait avoir un véhicule qui lui redonne son ancienne mobilité. Associé à un autre ingénieur, Harry C. Jennings, il mit au point une chaise en métal sur quatre roues. Les progrès de la science et des technologies ont permis de faire évoluer cette invention afin de favoriser l'autonomie dans les déplacements, l'ouverture sociale, l'accès à l'environnement extérieur et le confort des utilisateurs. Les premiers fauteuils motorisés apparurent dans les années 70 [Hon01]. Ils permirent aux personnes ayant une force réduite ou un handicap plus important de commander manuellement leur véhicule avec une sollicitation physique moindre.

Depuis une quinzaine d'années de nouveaux types de fauteuils faisant appel au transfert des technologies développées en robotique mobile font leur apparition dans les laboratoires de recherche. L'objectif est de permettre aux personnes à lourd handicap d'accéder à la mobilité autonome en augmentant les fonctionnalités de la machine et en améliorant l'interface homme machine. L'application de la robotique mobile ne se limite pas au développement de fauteuils robotisés mais propose également des plate-formes satellites qui peuvent aider la personne dans l'exécution de certaines tâches comme la préhension d'objets (projet ARPH : [Hop97, Hop01]).

Beaucoup de projets de recherche ont montré que le développement d'un fauteuil roulant intelligent est un travail de longue haleine. De plus, l'évaluation dans des situations réelles exige des années supplémentaires, qui ne débouchent pas automatiquement sur l'approbation du système comme assurant la sécurité des personnes handicapées. De ce fait, certains projets de recherche en matière de robotique d'assistance sont considérés comme incertains en termes de sûreté en raison de résultats expérimentaux insuffisants [Nis02].

De nombreuses techniques d'assistance à la navigation pour fauteuils roulants électriques ont été proposées, mais à ce jour il n'en existe pas de versions commercialisées. Il est nécessaire que les chercheurs et constructeurs travaillent ensemble afin de concevoir des fauteuils intelligents commercialisables [Nis02]. Actuellement, deux fauteuils roulants électriques équipés de capteurs utiles à la réalisation de système d'aide à la navigation sont directement exploitables pour la recherche : l'un développé par la société ActivMedia Robotics⁴ l'autre par Applied AI Systems Inc.⁵ sous le nom de TAO-7.

Les premiers prototypes de fauteuils électriques intelligents apparurent au début des années 1990. Ces projets sont pour la plupart toujours en évolution. Parmi ceux-ci, nous pouvons citer le projet Smart Wheelchair réalisé au CALL center (Communication Aids for Language and Learning) [Nis88] et le projet VAHM (Véhicule Autonome pour Handicapés Moteur) conduit par le LASC (Laboratoire d'Automatique des Systèmes Coopératifs) [Bou86]. Le projet Wheelchair, démarré en 1995 au département d'informatique du collège de Wellesley [Yan95], est basé sur le prototype du TinMan II du KISS Institute. Il a depuis été transféré au département d'intelligence artificielle du MIT (Massachusetts Institute of Technology). Le projet NavChair, conduit de 1990 à 1999 à l'université du Michigan [Bor90b], est depuis 1999 repris sous le nom de Hephaestus à l'université de Pittsburgh [Sim02]. Le projet OMNI (Office Wheelchair with High Manoeuvrability and Navigational Intelligence for People with Severe Handicap) est mené à l'université de Hagen depuis 1994 [Hoy94]. On peut encore évoquer le projet INRO de l'université de Würzburg (Intelligenter Rollstuhl \approx Fauteuil Intelligent), et plus récemment les projets Rolland de l'université de Brême, MAid (Mobility Aid for Elderly and Disabled People) ainsi que RobChair développé à l'université de Coimbra.

Ces systèmes possèdent différents niveaux d'autonomie. Pour certaines tâches le pilotage du véhicule est partagé entre l'utilisateur et la machine, par exemple lors de l'évitement d'obstacles le système prend temporairement la commande du véhicule. D'autres fonctionnalités requièrent une plus grande autonomie du véhicule comme le suivi de trajectoire planifiée où la charge du déplacement est totalement laissée au système. Ceci nécessite une estimation de la position du véhicule. En effet la position de départ du fauteuil doit être connue pour calculer sa trajectoire dans l'environnement. Et le contrôle de l'exécution du déplacement requiert une mesure de la position courante. Certains de ces projets n'ont pas suivi l'objectif de concevoir un système pouvant être complètement autonome. Nous allons dans la suite de cette partie présenter quelques projets et préciser, lorsqu'ils en font usage, leur mode de localisation.

⁴<http://www.activrobots.com/ROBOTS/RoboChariot.html>

⁵<http://www.aai.ca>

1.4.2 Projets

1.4.2.1 Le projet "A.R.P.H." – Assistance Robotique aux Personnes Handicapées

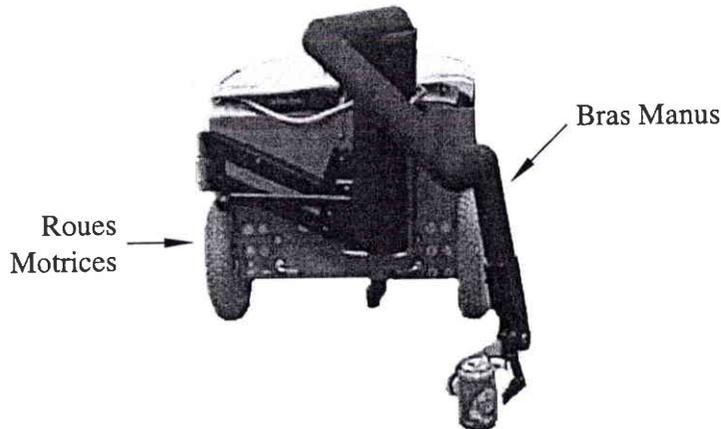


Figure 1.36 – Prototype du robot ARPH

Le projet ARPH développé au Laboratoire des Systèmes Complexes (LSC/CEMIF) est une plate-forme télémanipulée équipée d'un bras Manus [Laf01]. Elle est d'une taille plus petite qu'un fauteuil roulant et permet à son utilisateur de la télécommander à partir du fauteuil ou d'un lit pour aller prendre des objets inaccessibles.

(a) **Capteurs de la plate-forme** Afin d'estimer ses mouvements la plate-forme est équipée d'un odomètre. La perception de l'environnement situé autour du robot est réalisée à l'aide de huit capteurs à ultrasons répartis essentiellement sur l'avant du robot et d'une caméra.

(b) **Localisation de la base mobile** Deux approches de la localisation de la plate-forme ont été proposées. Dans [Hop98], les auteurs proposent uniquement l'utilisation de capteurs à ultrasons afin de localiser le robot. L'environnement est décrit sous forme géométrique. Des segments définissent les frontières entre l'espace libre et l'espace occupé. Les informations fournies par les capteurs à ultrasons permettent de construire une carte locale constituée de points (lors d'une seule acquisition) ou de droites (lors de plusieurs acquisitions successives). Cette carte est mise en correspondance avec le modèle de l'environnement afin de corriger l'estimation odométrique de la localisation. Une seconde approche de la localisation [Aid02] utilise une caméra et un modèle de l'environnement qui décrit les contours détectables à l'aide de segments de droite. La méthode met en correspondance les segments qu'elle extrait de l'image courante avec ceux du modèle de l'environnement en utilisant un arbre. Le calcul de la position du robot se base sur des méthodes issues du domaine de la reconnaissance d'objets.

(c) **Coopération Homme Machine** Plusieurs modes de commande sont offerts à l'opérateur pour piloter la base mobile :

- Un mode manuel qui donne tous les degrés de liberté à l'opérateur.
- Un mode automatique qui permet à l'opérateur de se décharger totalement de l'exécution de la mission.
- Un ensemble de modes mixtes dans lesquels la commande des degrés de liberté est partagée entre le système et l'opérateur humain.

Dans le mode automatique une localisation du système est requise mais elle reste sous contrôle humain afin de résoudre les situations ambiguës qui ne peuvent être résolues automatiquement. Le système de localisation donne à l'utilisateur une représentation graphique de la position du système dans l'environnement. Des essais ont été menés avec des personnes valides et des personnes handicapées (atteintes de

myopathie). Ils mettent en évidence une difficulté importante pour l'utilisateur à interpréter les informations fournies par le système [Gai03]. L'objectif suivant est d'utiliser une image comme information de retour pour l'opérateur.

1.4.2.2 Le projet "NavChair"

Le NavChair a été développé en 1991 pour satisfaire les besoins de personnes handicapées qui ne peuvent pas utiliser les fauteuils roulants disponibles (figure 1.37). Ce projet a été conçu principalement pour faciliter les évitements d'obstacles.

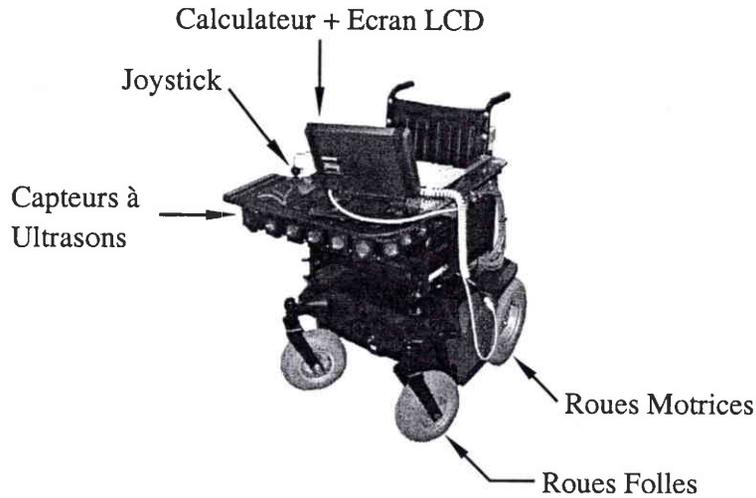


Figure 1.37 – Prototype du projet NavChair.

(a) **Prototype du NavChair** Les composants du système NavChair, sont embarqués sur un fauteuil commercial "Lancer" de chez Everest & Jennings, et sont alimentés par les batteries du fauteuil. Le système NavChair est équipé d'un ordinateur basé sur un processeur 486 cadencé à 33MHz, d'un ensemble de douze capteurs à ultrasons Polaroid montés à l'avant du véhicule et d'un module qui permet de réaliser une interface avec l'électronique du système.

Lors de la réalisation de tâches le système NavChair interrompt le lien direct entre le joystick et le module de puissance qui commande les moteurs (système de base du fauteuil commercial).

(b) **Module d'assistance à la navigation** Le module d'assistance à la navigation du "NavChair" [Bor90b, Sim95] est conçu pour améliorer la mobilité des personnes atteintes de handicaps qui limitent leur capacité à diriger un fauteuil roulant électrique. Le système de commande du NavChair est conçu pour éviter des obstacles, suivre des murs, et permettre la navigation en environnements encombrés ; il met en avant la coopération homme-machine pour la commande du fauteuil.

Equipé du système VFH [Bor91b] d'évitement d'obstacles, le fauteuil est capable de modifier la commande entrée par l'utilisateur afin de rendre les déplacements plus fiables. La position du joystick représente la direction désirée par l'utilisateur, et les lectures fournies par les capteurs à ultrasons reflètent la configuration de l'environnement proche du véhicule. La combinaison de ces informations est utilisée pour modifier le signal qui est envoyé au module de puissance.

Actuellement, le NavChair peut voyager dans des environnements d'intérieur non structurés à des vitesses moyennes élevées (jusqu'à 3 km/h) sans aucune collision. Les résultats présentés dans [Bel94] montrent un évitement des collisions dans les environnements d'intérieur représentatifs qui incluent des obstacles tels que des murs lisses, des poteaux de petites tailles et des meubles de bureau. Des utilisateurs valides ayant évalué le système le qualifient de sûr, confortable, et intuitif. Cependant, le NavChair donne une priorité à l'exécution d'un déplacement sans collisions. Par exemple, l'action d'évitement d'obstacle (VFH) ne permet pas au NavChair l'action de pousser une porte entre-ouverte pour son franchissement ou peut dans certaines configurations empêcher l'accostage à un bureau.

Depuis 1999, ce projet n'est plus poursuivi à l'université du Michigan mais a été repris à Pittsburgh sous le nom de Hephaestus [Sim02]. Le système du NavChair a depuis été mise en œuvre sur différents fauteuils

1.4.2.3 Le projet "RobChair"

RobChair est la contraction de "Rob" pour robot et de "Chair" pour Wheelchair (fauteuil roulant)[Pir02]. Ce projet, en cours de développement à l'Institute of Systems and Robotics, propose une assistance à la conduite d'un fauteuil roulant électrique via un module de navigation et une interface homme-machine utilisant la parole.

(a) **Principe du contrôle réactif partagé à l'aide d'une IHM vocale** La figure 1.38 donne une vue d'ensemble du système de commande partagée implémenté sur le fauteuil RobChair. L'interface de contrôle s'occupe de l'acquisition des données capteurs, de la lecture des informations du joystick ainsi que de la conversion des signaux de commande des moteurs en signaux de puissance permettant l'activation de ces derniers. Elle diffuse ces informations au calculateur qui lui fournit les vitesses linéaire, v_l , et angulaire v_a que doit avoir le véhicule afin de suivre la bonne trajectoire. L'interface homme machine vocale utilise un système de reconnaissance de la parole. Ce système donne accès à un ensemble de commandes de bas niveau (suivre le mur, en avant, à droite, à gauche, approcher de,...) permettant de diriger le véhicule.

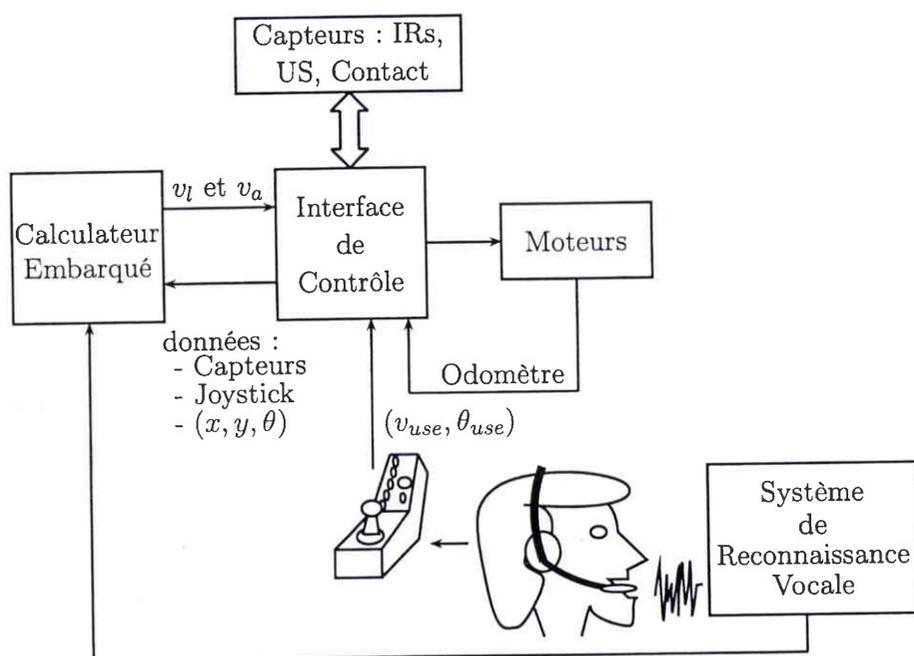


Figure 1.38 – Vue d'ensemble du système de commande de RobChair.

(b) **Le fauteuil et les capteurs** La figure 1.39 montre le fauteuil électrique équipé des différents capteurs, de l'ordinateur embarqué et de l'interface de commande. C'est un système de propulsion différentiel, avec deux roues folles à l'avant. Le fauteuil électrique peut être commandé en utilisant le joystick qui équipe le modèle de série ou en utilisant des ordres vocaux. Le système de perception est composé de 14 capteurs infrarouges réfléchissants tout ou rien (Sunx), 14 capteurs infrarouges analogiques réalisant une mesure de distance par triangulation (GP2D12), une ceinture de 7 capteurs à ultrasons, un capteur de chocs (Bumper) à l'avant du véhicule et d'un codeur optique sur chaque roue (odométrie).

(c) **Interface Homme Machine Vocale** L'interface homme machine vocale (IHMV) a pour objectif de remplacer la manette de commande lorsque l'utilisateur du fauteuil a un handicap tel qu'il lui est impossible d'utiliser une commande manuelle. L'utilisateur du Robchair peut alors piloter le fauteuil

électrique en utilisant uniquement la commande vocale comme par exemple : avance, recule, stop, gauche, droite, demi-tour, plus vite, ralentit.

La voix de l'utilisateur est acquise par un micro et traitée par l'unité de reconnaissance vocale. La commande vocale est alors envoyée au module de navigation qui calcule la trajectoire du fauteuil. Les commandes vocales donnent des indications grossières sur la direction que l'utilisateur veut suivre. En se basant sur cette direction et sur la perception de l'environnement proche fourni par les capteurs, l'algorithme de navigation calcule la meilleure trajectoire [Nun00, Con02].

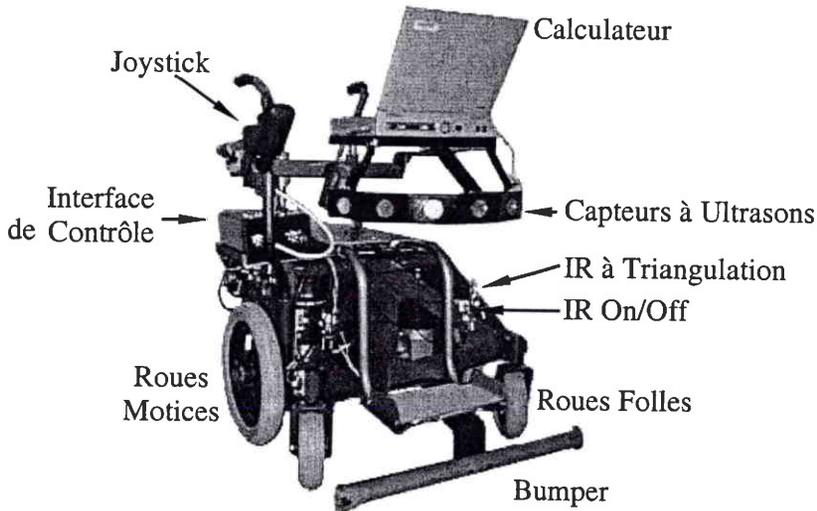


Figure 1.39 – Le prototype de RobChair avec ses équipements

(d) **Module de navigation** Le module de navigation possède une architecture hybride. Il combine les informations de contrôle fournies par les différents modules, comme l'évitement d'obstacle ou le suivi de murs, afin de modifier les informations de direction fournies par l'utilisateur ou le planificateur de trajectoires locales. Le module d'évitement d'obstacle et de suivi de murs sont basés sur l'utilisation de règles floues.

L'ensemble de ces fonctionnalités est mis en œuvre sur le prototype du RobChair destiné à naviguer dans des environnements semi-connus avec des obstacles en mouvement à l'aide d'une commande vocale.

1.4.2.4 Le projet "INRO"

(a) **Objectifs du projet INRO** Afin d'augmenter les possibilités de mobilité des personnes sévèrement handicapées le projet de recherche INRO (**I**ntelligent **R**ollstuhl : intelligent wheelchair) [Sch98] inclut la réalisation des fonctionnalités suivantes :

- un évitement d'obstacles fiable, permettant également l'évitement des escaliers descendants,
- la conduite d'un groupe de plusieurs fauteuils roulants, permettant à une infirmière d'exécuter des excursions avec un groupe de personnes handicapées en commandant juste le premier véhicule,
- la répétition autonome d'itinéraires appris à l'intérieur et dehors, permettant la mémorisation de tâches de transport répétitives à la charge de l'ordinateur embarqué, que l'infirmière peut lancer sur demande,
- une aide pour réaliser "un retour à la maison" sûr si l'utilisateur s'est égaré pendant un voyage,
- informer le personnel soignant en cas d'urgence sur la position du véhicule.

(b) **Le système** La figure 1.40 donne la configuration matérielle du prototype. Dans le cadre du projet, deux fauteuils roulants électriques commerciaux ont été équipés avec des capteurs et un ordinateur portable. Le PC est relié par l'intermédiaire des ports de communication série à l'interface de commande des moteurs, au système gérant les capteurs à ultrasons, au GPS différentiel et au modem radio. Le port parallèle est utilisé pour l'acquisition vidéo. Le joystick est utilisé comme interface d'entrée et l'afficheur

LCD en tant qu'interface de sortie. Les fauteuils roulant fixent leur direction par l'intermédiaire des roues arrière. Ils sont équipés d'un odomètre et d'un système de communication interne par RS 232.

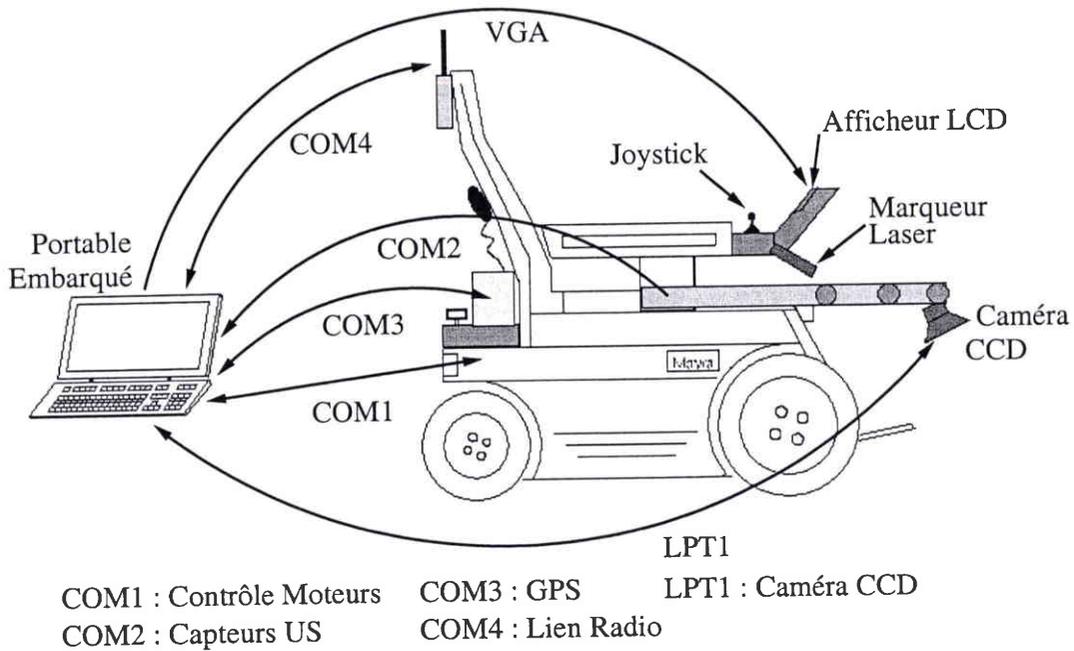


Figure 1.40 – Système du fauteuil INRO.

(c) Utilisation des capteurs

(c).1 Système à ultrasons Une ceinture de cinq capteurs à ultrasons est attachée à l'avant du fauteuil. Dans ce projet, ils sont combinés avec le marqueur laser et la vision pour une meilleure détection des obstacles. Le système de capteurs à ultrasons est utilisé pour deux tâches principales :

- Conduite en groupe : Le fauteuil électrique peut suivre automatiquement un fauteuil situé devant lui. Les capteurs à ultrasons fournissent une information sur la distance et la direction du véhicule de devant à un algorithme de contrôle qui tente de maintenir cette distance constante.
- Navigation intérieure : Dans ce cas les mesures de distances fournies par les capteurs à ultrasons combinées avec l'information odométrique sont comparées avec une carte enregistrée du bâtiment afin de retrouver la position actuelle du véhicule [Sch97].

(c).2 Marqueur laser Ce système vient en complément des capteurs à ultrasons, afin d'accroître la fiabilité de la navigation. Dans le contexte du projet INRO le système de marquage actif est mis en œuvre en combinant un marqueur laser, projetant trois lignes perpendiculaires à la direction de déplacement du véhicule, et d'une caméra CCD détectant ces lignes. A partir de la différence entre la position attendue des lignes (sans la présence d'un obstacle) et la position mesurée des lignes déformées (par la présence d'un objet), le système en déduit alors la position et la forme de l'objet. Dans de bonnes conditions d'éclairage, cette technique permet au véhicule de détecter des obstacles de petites taille ou la présence d'un escalier descendant que les capteurs à ultrasons n'auraient pas pu percevoir.

(c).3 GPS Le GPS est utilisé à l'extérieur pour déterminer la position du véhicule en combinant les informations (latitude, longitude) fournies par le récepteur GPS différentiel avec une carte locale enregistrée dans la mémoire de l'ordinateur.

(c).4 Fusion des données Le système sélectionne automatiquement les capteurs dont il a besoin en fonction des tâches qu'il doit réaliser.

- Pour la navigation à l'extérieur le DGPS est utilisé.

- Pour la navigation en intérieur ce sont les données des capteurs à ultrasons associées à l'odométrie qui permettent de maintenir une estimation de la position du véhicule.
- Pour la détection d'obstacle, les capteurs à ultrasons et la vision active (marqueur laser + caméra CCD) sont utilisés.

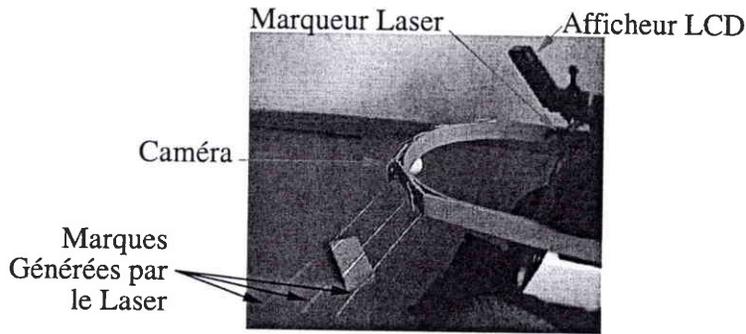


Figure 1.41 – Marqueur Laser installé sur le prototype INRO.

1.4.2.5 Le projet "Rolland"



Figure 1.42 – Prototype du fauteuil Rolland

Le groupe robotique cognitive de l'institut de fiabilité des systèmes de l'Université de Brême développe un moyen de transport intelligent pour personnes âgées et/ou handicapées, le fauteuil roulant autonome Rolland [Röf98]. Basé sur le fauteuil roulant commercial Meyra Genius, le système est équipé de capteurs à ultrasons, d'un système de vision et d'un PC. Il utilise des techniques qui lui permettent de réaliser des comportements complexes en combinant plusieurs comportements élémentaires tels que le suivi de murs ou le passage de portes. Les approches de navigation mises en application sont basées sur l'identification optique de balises (naturelles ou artificielles) : les objets significatifs dans l'environnement sont reconnus à l'aide d'un système de vision.

Le fauteuil "Rolland" met en œuvre une aide à la conduite qui adapte la vitesse commandée par l'utilisateur par l'intermédiaire du manche à la situation courante vis à vis d'obstacles plus ou moins proches. Ainsi le fauteuil roulant ralentit graduellement en approchant un obstacle. En outre, la conduite assistée aide l'utilisateur pour le passage de portes et les manoeuvres en espace contraint.

1.4.2.6 Le projet "OMNI"

(a) **La base mobile et le système de perception** Le projet OMNI (figure 1.43) est un robot mobile équipé d'un système de propulsion omnidirectionnel donnant trois degrés de liberté de déplacement sur le

sol, tel que n'importe quelle combinaison de mouvements vers l'avant, latéral, et de rotation soit possible. Cette configuration facilite les déplacements dans des environnements étroits [Hoy95, Hoy96]. De plus le véhicule intègre un système d'élévation de l'assise permettant de rehausser la personne de quatre vingt dix centimètres. Cela lui permet d'atteindre des objets situés en hauteur dans des environnements non adaptés pour les personnes en fauteuil roulant.

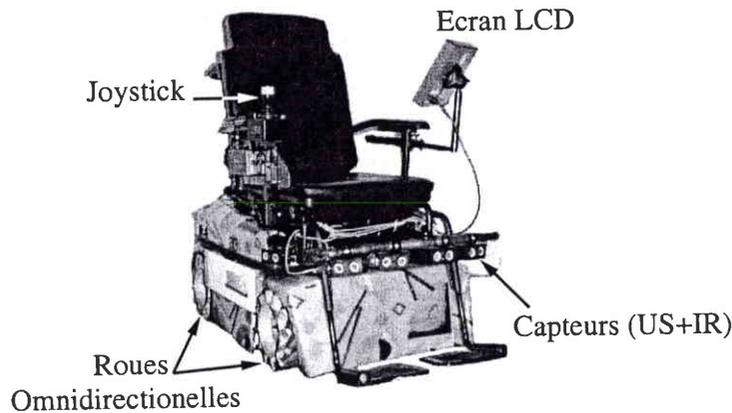


Figure 1.43 – Prototype du fauteuil OMNI.

Le véhicule est équipé d'un système de capteurs mixtes. Chaque capteur mixte est constitué d'un télémètre à ultrasons (US) et d'un télémètre infrarouge (IR). Cette combinaison particulière des capteurs permet à OMNI d'avoir une perception des obstacles qui n'est pas limitée par les caractéristiques des capteurs [Bor95]. C'est-à-dire qu'il fusionne les perceptions US et IR afin d'obtenir des informations complémentaires. Les ultrasons permettent une perception à longue distance avec une certaine imprécision sur la direction de l'objet alors que les infrarouges permettent une perception sur une plus petite distance mais avec une précision en direction plus importante.

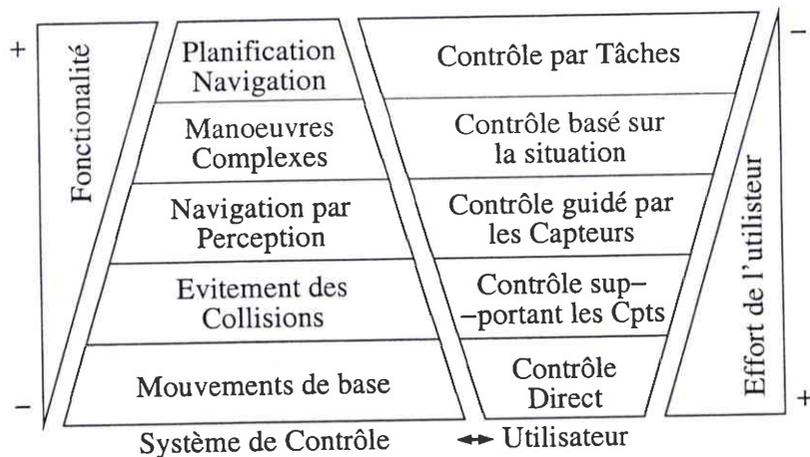


Figure 1.44 – Répartition de la charge de travail entre l'utilisateur et le système.

(b) **L'interface Homme-Machine et l'assistance proposée** Le dispositif de guidage du véhicule est un joystick, mais il est possible d'utiliser d'autres actionneurs, comme un bouton poussoir [Hoy99]. L'utilisateur peut également choisir des fonctions d'assistance via une représentation graphique sur l'écran LCD. La répartition de la charge de travail à fournir par l'utilisateur et le système d'assistance est représenté sur la figure 1.44. Sur cette figure, nous pouvons noter que plus la fonctionnalité du fauteuil est élevée, plus l'effort que doit fournir l'utilisateur pour piloter son fauteuil est faible. Au niveau de fonctionnalité le plus bas, l'effort que doit fournir l'utilisateur est maximal ; dans ce cas il dirige manuellement son fauteuil à l'aide du joystick. Au second niveau, les informations capteurs sont utilisées afin de fournir

des réactions évitement d'obstacles en laissant le reste du temps la commande à l'utilisateur. Le troisième niveau utilise les informations capteurs afin de réaliser des mouvements plus évolués comme le suivi de mur, le passage d'une porte, la réalisation d'un mouvement d'approche, permettant alors à l'utilisateur de se reposer lors de ces tâches fastidieuses, voire difficiles pour certaines personnes atteintes de troubles du mouvement, comme dans la maladie de Parkinson. Le quatrième niveau ne nécessite qu'une requête de la part de l'utilisateur afin de réaliser un mouvement de reproduction d'un chemin appris (par exemple : aller du salon à la cuisine) ou l'exécution inverse d'une manœuvre récente. Enfin, au dernier niveau, le véhicule prend tout en charge de la planification du chemin à l'exécution. Ce mode peut être intéressant si l'on peut détecter un malaise de l'utilisateur. Dans ce cas le fauteuil pourrait prendre en charge le retour vers les services médicaux appropriés. Sur l'ensemble des cinq niveaux présentés, seul les trois premiers niveaux sont opérationnels.

1.4.3 Conclusion sur la robotique mobile appliquée à l'assistance aux personnes handicapées

Les applications de la robotique mobile pour l'assistance des personnes handicapées et/ou âgées s'orientent de plus en plus vers une assistance à la conduite, mais n'ignorent pas totalement le mode autonome qui peut être utile lorsque la personne utilisant le fauteuil ne peut pas le diriger sans danger. Chaque projet propose différents niveaux d'autonomie du système soulageant ainsi la charge de travail de l'utilisateur du fauteuil. Ces différents niveaux d'autonomie sont semblables au schéma de la figure 1.44 issu du projet OMNI. Plus le système est autonome plus il assistera l'utilisateur dans la conduite du véhicule jusqu'à un niveau extrême où il devient totalement autonome. Dans ce mode de fonctionnement le système prend en charge la planification du chemin à suivre ainsi que son exécution, il lui est donc nécessaire de connaître sa position au cours du déplacement afin de déterminer si la trajectoire est correctement suivie.

1.5 Conclusion

La dernière partie de cette étude bibliographique traitait des systèmes d'assistance à la navigation de fauteuils roulants électriques. Nous pouvons conclure que les systèmes destinés à fournir une assistance à la navigation en proposant des primitives de base telles que le suivi de mur ou de direction ne nécessitent pas la connaissance globale de leur position. Par contre les systèmes proposant des fonctionnalités comme la navigation par tâches (ou objectifs) requièrent une connaissance précise de la localisation. Dans ce cas nous nous rapprochons de ce qui est fait en robotique mobile. Les systèmes de localisation sont très diversifiés. Nous pouvons pratiquement dire qu'il existe une solution particulière pour chaque système. Cette solution dépend d'une part du contexte de l'application et d'autre part des types de capteurs utilisés. De plus en plus de méthodes font appel à différents capteurs permettant ainsi des perceptions complémentaires de l'environnement.

Le ou les modèles de l'environnement sont conditionnés par les types de capteurs utilisés et la modélisation de la mesure faite par le capteur. Les capteurs de vision sont souvent associés à des modèles géométriques, alors que les capteurs comme les télémètres laser ou à ultrasons peuvent être indifféremment utilisés avec un modèle à grilles ou géométrique.

Lorsque les mesures sont formatées afin d'être rendues compatibles avec la représentation que l'on se fait de l'environnement, il reste encore le choix de la méthode d'estimation de la position :

- Des méthodes probabilistes, souvent utilisées avec une représentation sous forme de grille.
- Des méthodes de classification qui conviennent pour différents types de modélisation (géométrique ou vecteur d'information) mais dont les représentations doivent être exactement les mêmes pour les mesures et le modèle.
- Des méthodes de prédiction qui conviennent également à plusieurs types de représentation de l'environnement et des mesures.

Les représentations géométriques, bien que plus sensibles aux erreurs de modélisations que les représentations sous forme de grille, semblent bien adaptées à l'estimation en cours de mouvement car elles manipulent des petites entités de données.

La méthode de localisation en cours de mouvement présentée dans ce mémoire doit permettre la correction de l'estimation odométrique de la localisation du fauteuil autonome. Si elle est effectuée fréquemment, l'erreur qu'elle aura à corriger sera de faible amplitude. Dans ce cas, elle devra être peu coûteuse en temps de calcul. Le véhicule étant déjà équipé de capteurs à ultrasons, leur utilisation est donc envisagée pour la perception de l'environnement local. L'utilisation d'un système de vision monoculaire est également envisagé car il permettra une toute autre perception de l'environnement venant compléter celle fournie par les capteurs à ultrasons.

Cette page est laissée blanche intentionnellement

Chapitre 2

Méthode de localisation dynamique du VAHM

2.1 Introduction

Dans ce chapitre nous présentons tout d'abord le projet VAHM. Nous décrivons son architecture matérielle et logicielle ainsi que les différents modules qui le composent. Puis nous présentons l'évaluation des performances de l'odomètre qui est le moyen courant d'estimation de la localisation du fauteuil, justifiant ainsi de la nécessité d'une correction régulière de l'estimation de la localisation odométrique. Nous définissons ensuite la précision que nous désirons maintenir pour l'estimation de la localisation du véhicule. Enfin la suite du chapitre sera consacrée à la présentation de la méthode destinée à maintenir cette estimation correcte en corrigeant celle proposée par l'odomètre grâce aux données fournies par les télémètres à ultrasons.

2.2 Le projet VAHM

Initié par le LASC au début des années 90, le projet VAHM a pour objectif de développer un nouveau type de fauteuil roulant électrique [Bou01]. Ce fauteuil fait partie de la génération des véhicules intelligents. Il propose à son utilisateur des fonctionnalités plus ou moins évoluées facilitant ainsi la conduite du fauteuil.

Le système propose trois modes de fonctionnement :

- un premier mode, dans lequel l'utilisateur dirige le fauteuil via la manette de commande sans aucune assistance. Dans ce mode, le fauteuil fonctionne de manière similaire à la version commerciale.
- un second mode, dans lequel l'utilisateur peut toujours diriger le fauteuil à l'aide de la manette de commande mais en bénéficiant maintenant d'une assistance à la navigation comme par exemple l'évitement d'obstacle. Une autre possibilité de commande est réalisée à l'aide d'une interface graphique où l'utilisateur choisit une des primitives de déplacement proposées par le système, qui se chargera de la réaliser, comme par exemple le suivi de mur, ou de direction.
- Enfin le dernier mode, totalement autonome, où l'utilisateur indique le lieu qu'il veut atteindre à l'aide de l'interface graphique. Le système s'occupe ensuite de la recherche du meilleur chemin, ainsi que de sa réalisation.

2.2.1 Architecture matérielle

Le prototype actuel du VAHM est construit sur la base d'un fauteuil commercial, le PP201 de la compagnie suisse PowerPush (voir figure 2.1). L'électronique de gestion du fauteuil est architecturée autour d'un bus propriétaire de type CAN : le Bus DX développé par la société Control Dynamics Ltd. Le système DX minimal se compose de deux modules (voir figure 2.2) :

le module maître qui inclut physiquement la manette de commande et différents boutons de commandes et d'activations d'options ainsi que des indicateurs d'erreurs et de niveau d'énergie.

le **module de puissance** qui convertit les signaux de commande provenant du module maître en signaux de puissance permettant l'activation des moteurs.

Le bus système étant de type CAN seulement quatre fils sont nécessaires : deux pour l'alimentation (dont un fait office de masse/référence), et deux pour les signaux de commande.

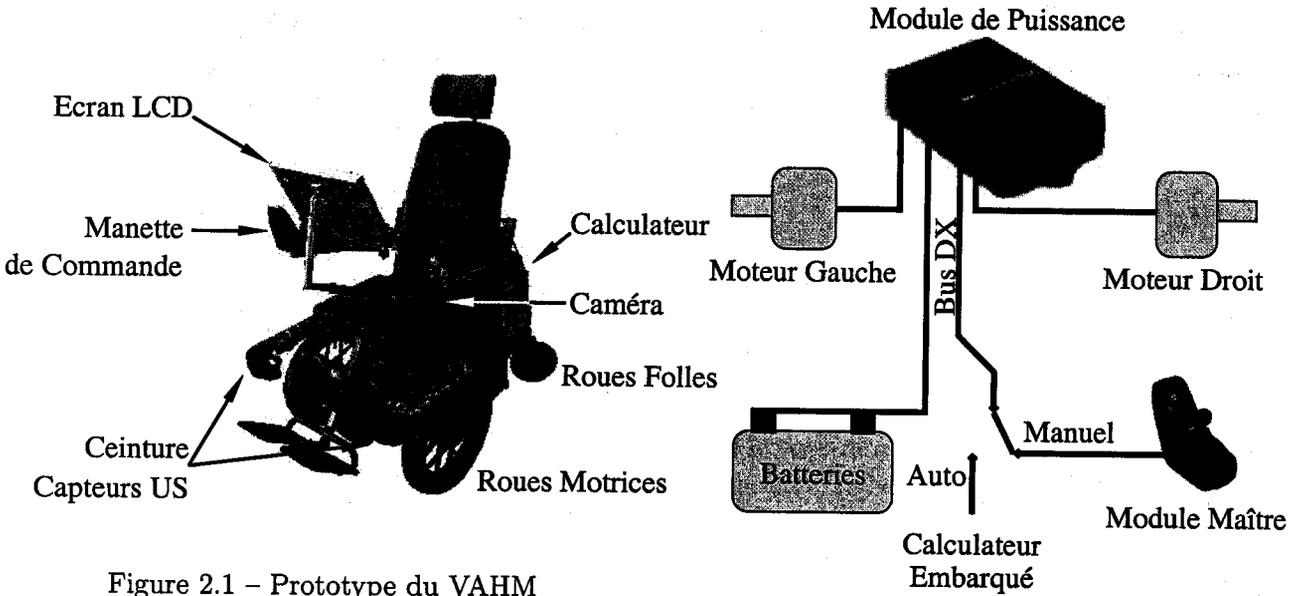


Figure 2.1 – Prototype du VAHM

Figure 2.2 – Architecture du Bus DX

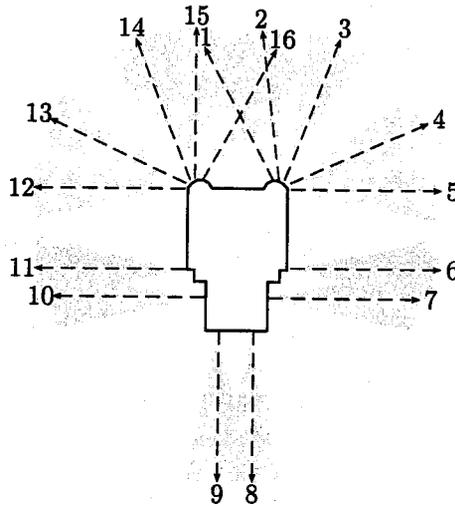


Figure 2.3 – Positionnement des capteurs à ultrasons tout autour du prototype du VAHM.

Le fauteuil a été équipé de seize capteurs à ultrasons répartis tout autour du véhicule comme le montre la figure 2.3. Le choix de la séparation de la ceinture de capteurs à ultrasons en deux demi ceintures a été guidé par des raisons pratiques, afin que la personne puisse facilement s'installer sur le fauteuil. Les capteurs 1 et 16 ont été volontairement croisés, pour éviter une zone morte devant l'assise du fauteuil. Ces capteurs sont des capteurs de type POLAROÏD dont l'activation est gérée par une carte à base d'un microcontrôleur Motorola 68HC811 qui s'occupe également de transmettre les seize mesures à l'unité de traitement via une liaison série RS-232. Le système odométrique se compose de deux codeurs optiques incrémentaux (un par roue motrice) couplés à une carte de comptage permettant d'atteindre une résolution de 4096 pas par tour de roue. Le fauteuil est également équipé d'un système de vision composé d'une caméra CCD Watec 505-EX et d'une carte d'acquisition grand public MiroPC/TV de chez PINACLE™ . L'unité de traitement des informations se compose d'un calculateur embarqué à l'arrière du fauteuil et destiné à la réalisation de diverses tâches comme :

- le traitement des mesures capteurs et la modélisation de l'environnement,
- la planification et le suivi de direction,
- la localisation,
- l'aide à la navigation,
- la gestion de l'interface homme machine (IHM).

En plus des cartes de comptage et d'acquisition vidéo, le calculateur accueille une carte de conversion numérique/analogique qui lui permet de générer sur le Bus DX des signaux compatibles avec ceux normalement envoyés par le module de contrôle du bus vers le module de puissance lorsque le sélecteur est en position Auto. Enfin, le fauteuil est équipé d'un écran LCD offrant une interface graphique entre l'utilisateur et le système. Une fois doté de l'ensemble de ces équipements, le fauteuil a un encombrement de $70 \times 110 \text{ cm}$.

2.2.2 Architecture logicielle

Le système est organisé autour de l'utilisateur afin d'utiliser au mieux ses capacités et d'alléger la charge de travail induite par la conduite du véhicule. Le degré d'autonomie du système dépend des capacités physiques et cognitives de la personne. Plus le handicap de la personne lui rend la tâche de conduite difficile plus l'autonomie du fauteuil est importante.

En mode assisté, l'utilisateur peut diriger le fauteuil à l'aide de la manette de commande. En présence d'un obstacle non perçu par celui-ci, le module d'évitement d'obstacle s'active et contourne l'obstacle. Dans le mode de fonctionnement assisté, l'utilisateur du système a également la possibilité de choisir des primitives de déplacement telles que :

- le suivi d'espace libre** : ce mode est similaire à un évitement d'obstacle ; le robot choisit la direction qui contient l'espace le plus ouvert en fonction des données fournies par les capteurs à ultrasons. C'est un mode de navigation sans objectif. L'utilisateur peut l'interrompre dès qu'il le souhaite.
- le suivi de direction** : dans ce mode l'utilisateur indique au système la direction à suivre à l'aide de l'interface graphique. Le système suit cette direction tant que l'utilisateur ne lui donne pas l'ordre de s'arrêter ou de suivre une autre direction. Si le système perçoit un obstacle alors il active le module d'évitement d'obstacle. Une fois l'obstacle évité, le système rejoint la direction à suivre.
- le suivi de mur** : ce mode consiste à suivre une ligne parallèle au mur, qui est déterminée en fonction des données fournies par les capteurs à ultrasons latéraux.

En mode autonome, le système attend que l'utilisateur lui indique le lieu à atteindre, puis il détermine la meilleure trajectoire et ensuite l'exécute. Le suivi de trajectoire est réalisé en décomposant la trajectoire en segments de droite. Ainsi la trajectoire se compose d'une suite de directions à suivre. Comme précédemment tout évitement d'obstacle est prédominant sur le suivi de trajectoire qui est repris lorsque l'obstacle est dépassé.

Pour que le mode de fonctionnement autonome s'exécute dans de bonnes conditions une connaissance de la localisation est nécessaire. En effet, pour que le calcul de la trajectoire soit optimal, il ne suffit pas de savoir où l'on va mais également d'où nous partons. Sur le VAHM, deux modes de localisation existent [Hor00].

Un premier mode de localisation statique permet de déterminer la position et l'orientation du véhicule sans connaissance a priori de ceux-ci [Cou00, Hor03a]. Cette méthode utilise des grilles d'occupations. Une première grille, appelée grille locale (figure 2.5), décrit l'environnement proche du fauteuil perçue par les capteurs à ultrasons. La seconde grille, appelée grille globale (figure 2.4), représente l'environnement dans sa totalité, elle est construite une seule fois et mémorisée dans la mémoire du calculateur. Une cellule de l'une ou l'autre grille représente l'occupation d'un élément de surface de l'environnement réel. La recherche de la position et de l'orientation du véhicule est réalisée en déplaçant la carte locale sur la carte globale et en retenant la configuration qui superpose un nombre maximal de cellules ayant la même valeur dans les deux cartes.

Le second mode de localisation, "dynamique" [Hor03b], est exécuté alors que le véhicule est en mouvement et doit donc donner un résultat rapidement. Il se base sur les informations fournies par l'odomètre et sur la connaissance d'une position antérieure du véhicule (voir section 1.3.1). Mais comme nous l'avons fait

remarquer au paragraphe 1.3.1.2 cette estimation est sujette à des erreurs qui peuvent très vite atteindre des amplitudes importantes. L'origine des erreurs de mesure sont très diverses, comme par exemple :

- les erreurs de modélisation des caractéristiques du fauteuil ou des variations de celles-ci (diamètre de roues qui varie pour des roues à chambre à air, position et corpulence de la personne sur le fauteuil),
- le bruit de quantification inhérent aux codeurs utilisés,
- les irrégularités sur la surface du sol provoquant la perception d'un mouvement qui n'est pas réel.

De plus, en raison du caractère intégrateur de l'odomètre, l'erreur d'estimation n'est pas bornée. Cette incertitude sur la localisation odométrique peut perturber l'exécution de modules comme le suivi de trajectoire. Le paragraphe suivant présente une évaluation de l'erreur d'odométrie.

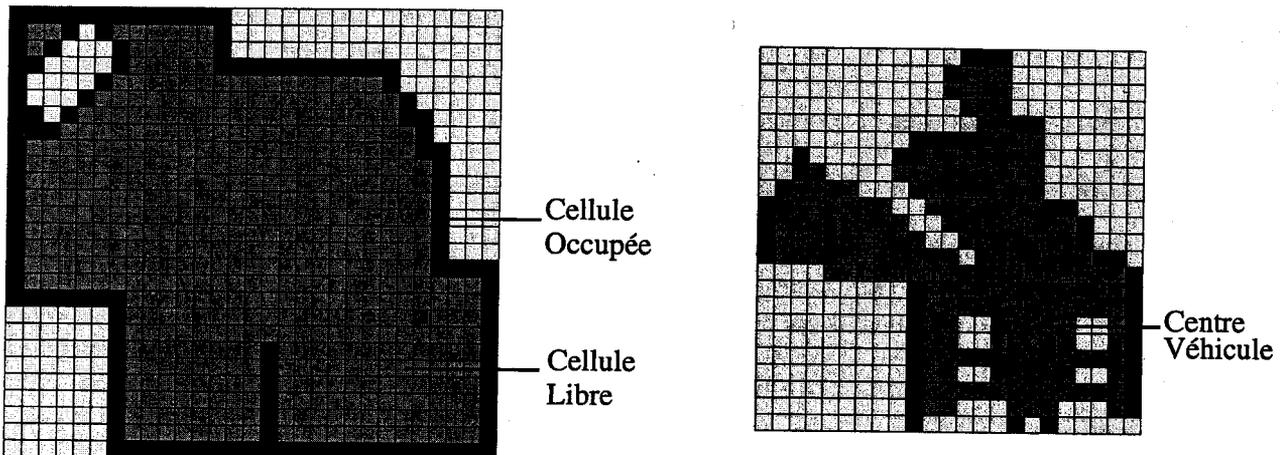


Figure 2.4 – Grille globale construite à partir d'un plan d'occupation de l'environnement.

Figure 2.5 – Grille locale construite autour du robot à partir des données des capteurs à ultrasons.

2.3 Evaluation de l'erreur odométrique

L'odomètre est un capteur couramment utilisé en robotique mobile (voir paragraphe 1.3.1). Son rôle est d'estimer le déplacement relatif du robot. Nous présentons dans ce paragraphe l'évolution de l'erreur sur la position odométrique mesurée. L'étude a été réalisée expérimentalement en faisant effectuer au prototype du VAHM un déplacement rectiligne à vitesse réduite vers l'avant d'environ cent cinquante centimètres. Nous avons relevé à chaque essai la position et l'orientation au sol ainsi que celles fournies par l'odomètre pour cinquante, cent et cent cinquante centimètres parcourus. Ceci nous permet de comparer l'estimation de la position avec sa valeur exacte et d'en déduire l'erreur commise par l'odomètre sur un parcours en ligne droite.

Sur la figure 2.6, les croix indiquent les positions fournies par l'odomètre, la ligne en trait plein est la trajectoire réelle. Les ellipses représentent l'incertitude sur la position estimée. Les points situés dans ces ellipses ont un taux de confiance de 90% en supposant que le vecteur constitué des erreurs sur les coordonnées est gaussien. Les cônes en pointillés représentent l'incertitude en orientation. Le tableau 2.1 donne la moyenne et l'écart-type pour la position et l'orientation estimées après un parcours de cinquante, cent et cent cinquante centimètres. Sur la figure 2.6 nous constatons qu'à partir d'une distance parcourue de l'ordre de 50 cm l'ellipse d'incertitude sur la position estimée possède un rayon sur l'axe des x de l'ordre de 10 cm et un rayon d'environ 15 cm sur l'axe des y . L'erreur sur l'orientation est elle comprise entre $\pm 10^\circ$.

Le véhicule du projet VAHM ayant pour but de naviguer dans des environnements d'intérieur comme un appartement ou un hôpital, où il arrive que le fauteuil ait à passer dans des endroits étroits, une estimation précise de la position et de l'orientation est nécessaire afin de bien suivre les trajectoires programmées. Compte tenu des résultats obtenus pour l'odométrie un recalage régulier de l'estimation de la localisation odométrique est nécessaire afin de limiter les erreurs d'estimation en position et en orientation. Dans le contexte des déplacements d'intérieur une erreur en position de 5 cm sur chaque axe et une erreur en

orientation de 3° sont des limites acceptables. Ainsi l'objectif de notre méthode de localisation du VAHM par l'utilisation de divers capteurs est de replacer l'estimation de la position dans ces limites.

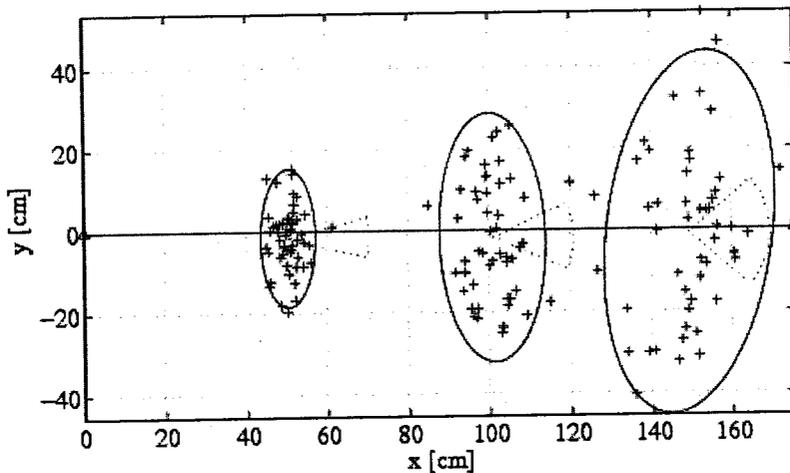


Figure 2.6 – Test de l'odomètre du prototype du VAHM

distance	ε_x [cm]		ε_y [cm]		ε_θ [°]	
	μ	σ	μ	σ	μ	σ
50 cm	0.2	3.2	-1.7	7.9	0.2	4.8
100 cm	0.7	6.1	-1.9	14.2	0.6	8.0
150 cm	-0.8	9.9	-0.9	20.8	-0.2	13.5

Tableau 2.1 – Valeur moyenne et écart-type sur la localisation estimée pour trois distances (50 essais par distances).

2.4 Contexte et principe

2.4.1 Contexte

La méthode développée dans ce chapitre, doit permettre au véhicule du projet VAHM de connaître sa localisation avec une précision telle que ses déplacements autonomes et son assistance au pilotage soient fiables. Il est alors nécessaire de corriger régulièrement la localisation estimée par l'odomètre au moyen de mesures extéroceptives permettant de se recalibrer par rapport à l'environnement. Pour réaliser cette correction, nous utilisons une carte de l'environnement préalablement mémorisée décrivant, à l'aide de segments de droite, les frontières entre l'espace libre et l'espace occupé ainsi qu'un ensemble de seize mesures de distances fournies par des capteurs à ultrasons embarqués sur le robot [Kre01]. La position et la direction de tir de chaque capteur sont connues dans le référentiel lié au robot. La fréquence des recalages doit être telle que l'erreur odométrique reste limitée. Les résultats obtenus au paragraphe 2.3 nous mènent à exécuter une correction tous les 50 cm. Le véhicule navigant à une vitesse moyenne de l'ordre de 2 km/h, le temps de parcours des 50 cm est d'environ 900 ms. Il faudra donc que la correction de l'estimation odométrique de la localisation soit réalisée dans ce laps de temps.

2.4.2 Principe de la méthode

Lors d'une correction, le système effectue une lecture des seize capteurs à ultrasons. Puis il construit une carte locale de son environnement composée de seize points définis dans le repère lié au véhicule. La méthode compare ensuite le modèle local avec le modèle global de l'environnement au voisinage de la localisation estimée par l'odomètre et tente de faire correspondre au mieux les deux modèles. Il s'agit

de déterminer la position et l'orientation du véhicule dans le référentiel global qui amènent les points du modèle local à être les plus proches possible des segments auxquels ils correspondent. Cela revient à rechercher une translation et une rotation du référentiel du robot par rapport au référentiel global. Le calcul de cette transformation est basé sur le critère des moindres carrés.

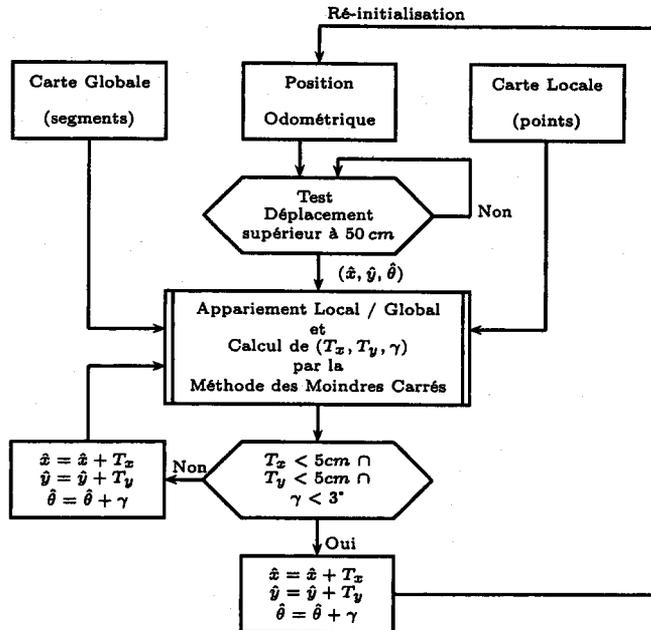


Figure 2.7 – Principe de la méthode de correction de la localisation par capteurs à ultrasons.

La figure 2.7 donne le principe de fonctionnement de la méthode de correction. Le mode de calcul de la transformation est itératif; nous déterminons de petites transformations à appliquer successivement à la position estimée du robot, jusqu'à ce que celles-ci ne soient plus significatives. Lors de la première itération le modèle local est projeté dans le référentiel du modèle global autour de la position estimée par l'odomètre du robot. Ensuite les points du modèle local sont mis en correspondance avec les segments du modèle global et l'algorithme détermine la transformation – composée des translations T_x et T_y ainsi que de la rotation γ – à appliquer à la position estimée afin de minimiser la somme des distances entre les caractéristiques appariées des différents modèles. Si les trois composantes T_x , T_y et γ de la transformation sont inférieures à leurs seuils respectifs, fixés en fonction des performances exigées, alors l'algorithme prend fin. Par contre, si la transformation est d'amplitude significative, l'algorithme recommence l'appariement des données locales avec les données globales à partir de la nouvelle position estimée, et calcule de nouveau une transformation.

Lors de la mise en œuvre de la méthode de correction nous choisissons donc pour les seuils de T_x et T_y la valeur de 5 cm et pour le seuil de γ la valeur de 3°.

2.5 Modèle de l'environnement et des capteurs à ultrasons

Dans cette partie, nous allons présenter les modèles utilisés pour la correction en cours de mouvement de la position odométrique, à savoir :

- le **modèle global** qui est connu a priori par le robot ; il peut être généré à partir d'un plan d'architecte et mémorisé sur la mémoire de masse du calculateur embarqué,
- le **modèle local** qui est construit automatiquement par le robot lorsque la procédure de correction de la localisation estimée du robot est exécutée.

2.5.1 Modèle global de l'environnement

L'environnement dans lequel évolue le véhicule est un environnement d'intérieur structuré, comme peut l'être un appartement. Ainsi la plupart des objets présents dans ce type d'environnement ont des formes

géométriques simples. Nous avons donc choisi de représenter par des segments de droites les frontières entre l'espace libre et l'espace occupé par les obstacles dans le plan horizontal des capteurs à ultrasons. Ces segments sont orientés dans le sens p_1 vers p_2 . Le point p_1 (respectivement p_2) est l'extrémité droite (respectivement gauche) du segment lorsque l'observateur fait face à l'obstacle, comme cela est visible sur la figure 2.8(a), où la partie hachurée représente l'obstacle. Lors de l'appariement des données des deux modèles (local et global), la position odométrique n'est pas exactement connue. Alors, même si les mesures fournies par les capteurs à ultrasons sont exemptes de perturbations, il est impossible que chacune d'elles se situe exactement sur un des segments du modèle global. De ce fait, une ellipse est adjointe à chaque segment (figure 2.8(b)). Elle définit une zone de validité d'appariement des données au voisinage de ce segment.

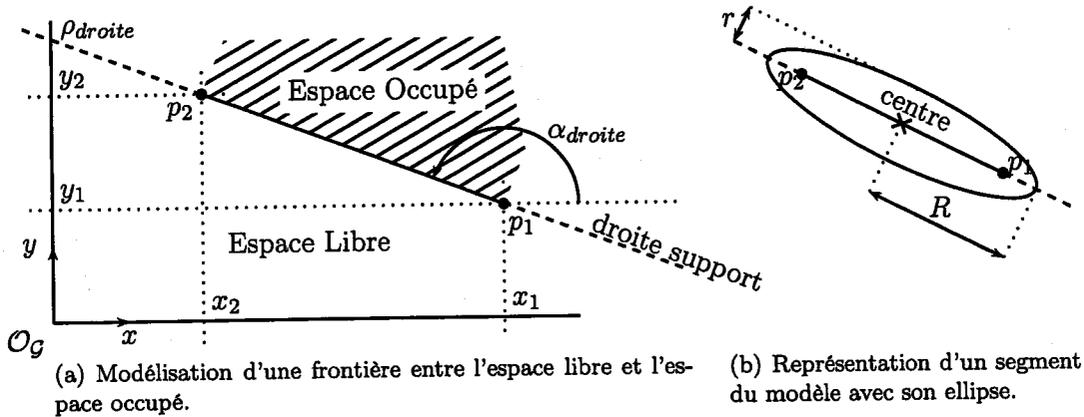


Figure 2.8 – Modélisation des frontières de l'environnement.

Chaque segment est initialement défini par ses deux extrémités dans le plan où évolue le robot, soit p_1 et p_2 ces points (figure 2.8(a)). Ils nous permettent de déterminer les paramètres suivants de la droite support du segment :

- l'angle α_{droite} que fait la droite $\overline{p_1p_2}$ avec l'axe des abscisses du référentiel global,
- l'ordonnée à l'origine ρ_{droite} .

L'équation de la droite support du segment s'écrit :

$$\sin(\alpha_{droite}) \cdot x - \cos(\alpha_{droite}) \cdot y + \rho_{droite} = 0 \quad (2.1)$$

L'ellipse associée au segment est définie par (figure 2.8(b)) :

- son centre : il est situé sur la droite support du segment du modèle et à équidistance des extrémités du segment,
- son petit rayon r : il est normal au segment du modèle, et sa valeur dépend de l'erreur possible d'estimation commise par l'odomètre,
- son grand rayon R : il est confondu avec le segment du modèle, et sa valeur est égale à la demi-longueur du segment augmentée par la valeur de r : $R = \frac{l}{2} + r$, où l est la longueur du segment.

Les résultats obtenus au paragraphe 2.3 montrent que l'erreur d'estimation peut atteindre $\pm 15 \text{ cm}$ (carré de coté de 30 cm), nous choisirons donc pour r une valeur de 15 cm .

2.5.2 Construction du modèle local

2.5.2.1 Modèle d'un capteur à ultrasons

Comme nous l'avons vu précédemment (paragraphe 1.1.2.1) la connaissance d'une seule information télémétrique fournie par un capteur à ultrasons ne permet pas de connaître avec précision la position du point d'impact avec l'obstacle dans le cône d'émission du capteur. Sur la figure 2.9 nous avons représenté la distance retournée par un capteur à ultrasons dont l'axe acoustique est normal à la surface détectée. Dans cette configuration, d (la plus courte distance qui sépare le capteur de l'obstacle) se mesure sur l'axe acoustique du capteur. L'arc de cercle marqué "Zone de détection" définit la région sur laquelle il est possible que se situe l'objet ayant renvoyé l'onde ultrasonore (étant donné que, lors des corrections

de la localisation odométrique, nous ne savons pas a priori si le capteur est normal à un segment de l'environnement). Nous avons choisi comme modèle pour les capteurs à ultrasons le modèle ponctuel pour les raisons suivantes :

- la limitation de la charge de traitement des données fournies par les capteurs à ultrasons,
- son intégration aisée dans un algorithme d'ajustement de la position et de l'orientation d'un robot.

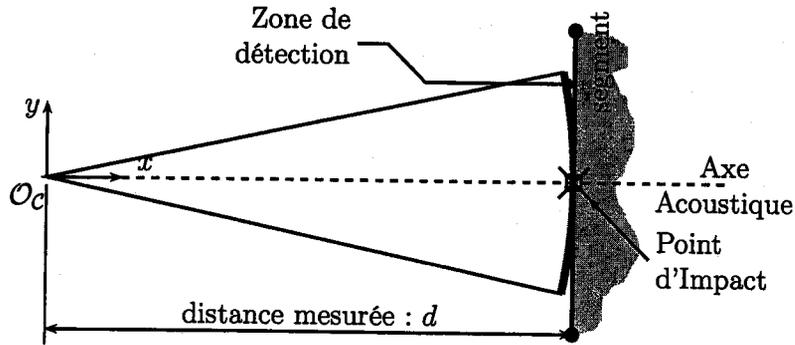


Figure 2.9 – Modélisation d'une mesure de distance fournie par un capteur à ultrasons.

Nous supposons donc que la distance renvoyée par le capteur à ultrasons est située sur l'axe acoustique de ce dernier, car ce choix limite l'erreur de représentation (l'erreur maximum est le demi-arc de détection).

Les informations dont nous disposons sont :

- la distance qui sépare le capteur à ultrasons de l'objet le plus proche situé dans son cône d'émission en supposant qu'aucun phénomène comme la diaphonie ou des réflexions multiples ne se produisent,
- une estimation de la position et de la direction de tir de ce capteur dans le référentiel global, issue de la connaissance de la position du capteur sur le robot et de l'estimation odométrique de la localisation du robot.

En disposant de ces informations pour chaque capteur, nous pouvons alors établir une carte de l'environnement proche du véhicule. Avec ce modèle, l'erreur maximale de modélisation a lieu lorsque la détection se fait avec un angle d'incidence limite égale au demi-angle d'ouverture du faisceau émis. L'erreur d'interprétation de la mesure est équivalente à l'arc de cercle liant le point d'impact modélisé à l'extrémité qui a été en contact avec la surface. Etant donné que les capteurs peuvent détecter des obstacles situés à environ 280 cm, l'erreur maximale est d'environ 6 cm lorsque le capteur est en incidence limite (soit 12°).

$$err_{max} = distance_{max} \left(1 - \cos \left(\frac{ouverture}{2} \right) \right) = 280 (1 - \cos (12)) \approx 6 \text{ cm} \quad (2.2)$$

Cette erreur n'est pas négligeable mais nous avons choisi ce modèle pour sa simplicité et le peu de ressources calculatoires nécessaires à son utilisation.

2.5.2.2 Création de la carte locale

Chacun des seize capteurs fournit une mesure de distance dans une direction spécifique. La position et la direction de chaque capteur sont connues dans le référentiel du robot. L'axe acoustique d'un capteur à ultrasons est confondu avec l'axe des abscisses du repère lié au capteur (figure 2.9). Nous pouvons alors calculer la position des différents points d'impact dans le référentiel du robot, via la relation :

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_C \\ y_C \end{bmatrix} + \begin{bmatrix} \cos(\alpha_C) & -\sin(\alpha_C) \\ \sin(\alpha_C) & \cos(\alpha_C) \end{bmatrix} \cdot \begin{bmatrix} d_i \\ 0 \end{bmatrix} \quad (2.3)$$

où $[x_i, y_i]^T$ est la position du point d'impact dans le référentiel du robot, $[x_C, y_C]^T$ la position du capteur dans le référentiel du robot, α_C sa direction de tir et d_i la distance mesurée (figure 2.10).

Ensuite ces positions sont projetées dans le référentiel global, via la relation :

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} x_R \\ y_R \end{bmatrix} + \begin{bmatrix} \cos(\theta_R) & -\sin(\theta_R) \\ \sin(\theta_R) & \cos(\theta_R) \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2.4)$$

où (x_R, y_R, θ_R) est la position du robot, et (x_g, y_g) est la position du point d'impact dans le référentiel global (figure 2.10).

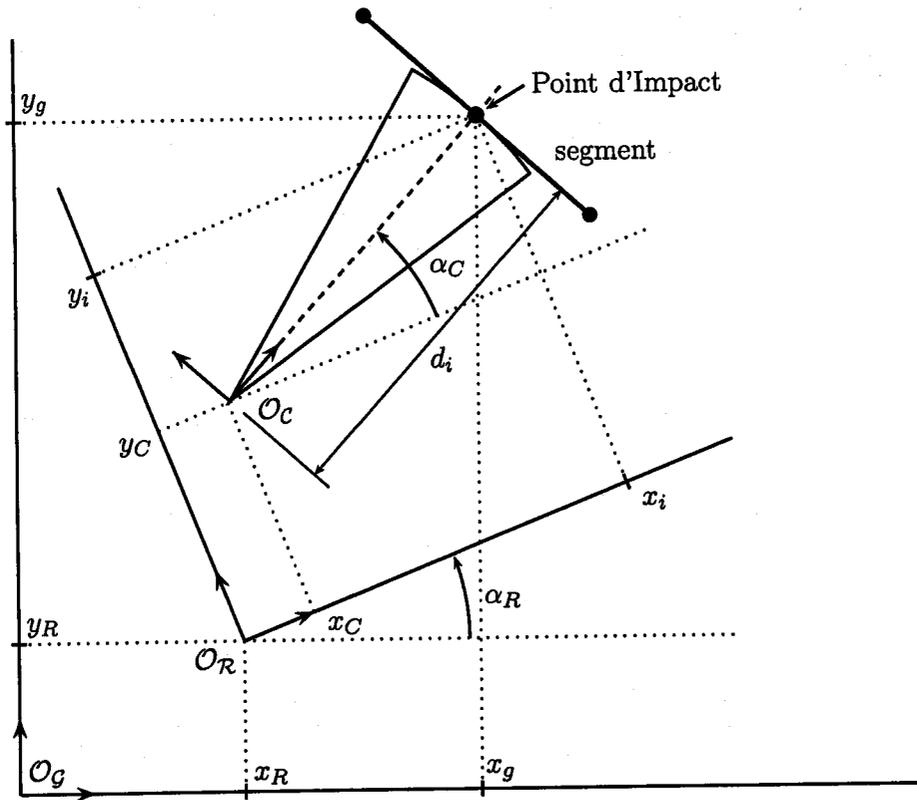


Figure 2.10 – Représentation des variables utilisées pour la projection des mesures des capteurs à ultrasons dans le référentiel global – projection de la carte locale dans la carte globale.

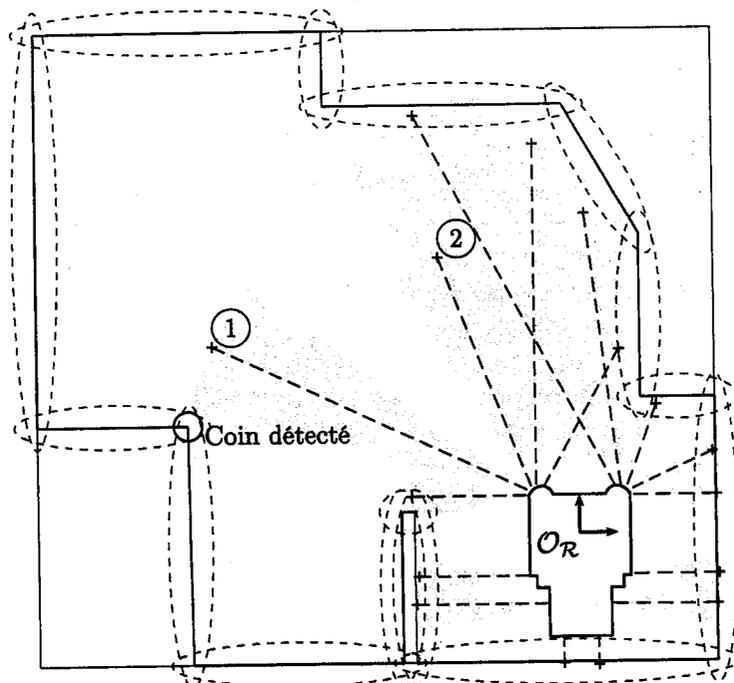


Figure 2.11 – Exemple d'une carte locale après acquisition de 16 capteurs à ultrasons.

La figure 2.11 montre un exemple de carte locale construite à partir des données renvoyées par un ensemble de seize capteurs à ultrasons. Les régions grises montrent les zones de détection de chaque capteur. Les segments en tirets représentent les axes acoustiques des différents capteurs. Les extrémités de ces segments donnent les positions des points du modèle local dans le référentiel global une fois projetés

autour de la position du robot. Sur la figure 2.11 nous avons choisi de présenter une projection autour de la position réelle du robot. Nous constatons que plus les capteurs ont leurs axes acoustiques proches de la normale des surfaces détectées, plus leurs points dans le modèle local sont proches des segments modélisant les frontières entre l'espace libre et l'espace occupé.

Sur la figure 2.11 deux points du modèle local ne sont proches d'aucun segment du modèle global. Cela provient des erreurs suivantes :

- Pour le point le plus à gauche (point ①), l'erreur de modélisation provient de la détection du coin proche de l'extrémité gauche du cône du capteur à ultrasons, marqué par un cercle noir sur la figure 2.11.
- Pour le second point (point ②) il s'agit du phénomène de diaphonie, c'est à dire que le capteur a perçu l'onde réfléchié émise par un autre capteur. Ce type d'erreur n'est pas prévisible et difficilement détectable. Nous tenterons d'en limiter l'influence par l'intégration d'un critère de validation des appariements reposant sur des caractéristiques des capteurs à ultrasons telles que leur angle d'ouverture, l'angle d'incidence maximal de perception et la position du capteur dans l'environnement.

2.6 Méthode de correction de la position odométrique

La méthode de correction de la localisation estimée utilisant les mesures renvoyées par les capteurs à ultrasons se compose de deux phases :

1. la phase d'appariement des points d'impact de la carte locale avec les segments de la carte globale,
2. la phase de calcul de la transformation à appliquer à la localisation du robot afin que les deux modèles coïncident au mieux au sens des moindres carrés.

2.6.1 Mise en correspondance des modèles

Lorsque les points de la carte locale ont été projetés sur la carte globale autour de la position estimée du robot, la mise en correspondance des modèles consiste à chercher pour chaque capteur quel segment de la carte globale il a détecté et ainsi créer un appariement des points de la carte locale avec les segments de la carte globale. Une fois l'appariement déterminé, il est possible de calculer les distances entre les points et les segments auxquels ils ont été associés.

Si la localisation estimée à l'aide de l'odomètre était exacte et que les obstacles détectés par les capteurs étaient situés sur les axes acoustiques respectifs de chaque capteur, alors les points de la carte locale seraient situés exactement sur les segments de la carte globale. Or, comme la localisation estimée est connue avec une certaine imprécision, et que les capteurs à ultrasons ne détectent pas uniquement les objets situés sur leur axe acoustique, l'appariement des données locales avec les données globales est réalisé au moyen des ellipses associées aux segments représentant les frontières de l'environnement. L'ellipse associée au segment est utilisée afin de tenir compte des erreurs sur la position et l'orientation estimées et du fait qu'un capteur détecte des obstacles dans un cône et pas uniquement sur son axe acoustique.

2.6.1.1 Affectation des points d'impact du modèle local

Le principe d'appariement des points de la carte locale avec les segments de la carte globale consiste à vérifier pour chaque point si celui-ci, une fois projeté autour de la position estimée du robot dans le référentiel global, est situé à l'intérieur d'une ellipse. Dans l'affirmative, le point est apparié avec le segment auquel l'ellipse est associée. La présence du point dans une ellipse est vérifiée pour chaque segment de la carte globale. Ce principe est résumé par l'algorithme 2.12.

Pour chaque point de la carte locale, l'algorithme teste si le point est situé dans l'ellipse de chacun des segments de la carte globale, c'est-à-dire qu'il ne s'arrête pas à la première ellipse trouvée. Cet algorithme peut donc générer des appariements multiples, mettant en correspondance un point de la carte locale avec plusieurs ellipses, comme cela peut être le cas lorsque le point d'impact est situé à proximité de l'intersection de deux segments. La figure 2.13 montre un exemple où deux points p_1 et p_2 sont situés dans

plusieurs ellipses à la fois. L'algorithme ne permet pas de déterminer quel est le meilleur des appariements. Nous verrons dans les paragraphes suivants les critères à utiliser pour déterminer le segment qui convient.

```

ENTREES : Les points de la carte locale et les segments de la carte globale
SORTIES : Liste des appariements et le nombre d'appariements
POUR tous les points de la carte locale
    POUR tous les segments de la carte globale
        Tester si le point courant est dans l'ellipse courante
        SI OUI
            Apparier le point courant et le segment courant
            Incrémenter le nombre d'appariement
        FIN SI
    FIN POUR
FIN POUR
    
```

Figure 2.12 – Algorithme d'appariement des points de la carte locale avec les segments de la carte globale.

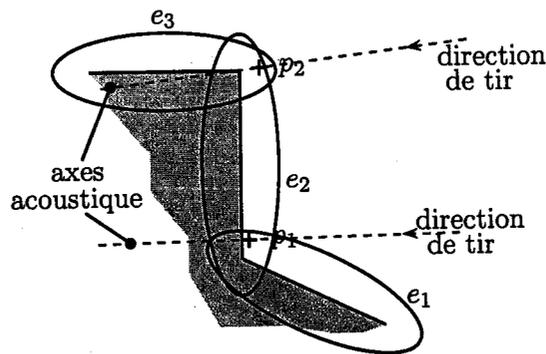


Figure 2.13 – Exemples d'appariements multiples possibles.

2.6.1.2 Calcul de la distance entre le point et le segment

L'algorithme de correction de la localisation nécessite le calcul des distances l_{PS} qui séparent les points P des segments S auxquels ils sont appariés. En effet, la transformation (T_x, T_y, γ) à apporter à la localisation estimée du véhicule est calculée en minimisant la somme de ces distances au carré (voir paragraphe 2.6.2). La distance l_{PS} est donnée par la relation suivante :

$$l_{PS} = \sin(\alpha_S) \cdot x_P - \cos(\alpha_S) \cdot y_P + \rho_S \quad (2.5)$$

où $\cos(\alpha_S)$, $\sin(\alpha_S)$ et ρ_S sont les paramètres définissant la droite support du segment S dans le référentiel global (équation 2.1) et (x_P, y_P) sont les coordonnées du point P dans le référentiel global.

L'algorithme d'appariement (figure 2.12), teste l'ensemble des frontières du modèle global pour chaque point du modèle local. Lorsqu'un point est situé dans une ellipse associée à un segment, la distance qui sépare le point du segment est calculée. Chaque appariement est donc mémorisé par :

- le point du modèle local,
- le segment qui lui est associé,
- la distance qui les sépare.

Ce paramètre de distance est aussi employé comme premier critère de discrimination lorsque plusieurs appariements sont envisagés pour un même point. En effet, lorsque l'algorithme trouve deux ellipses recouvrant un même point, il sélectionne celle pour laquelle la distance point/segment est la plus petite. L'algorithme modifié (Algorithme A.1) est donné en annexe A. Dans la suite nous considérons cette version de l'appariement des données comme la version initiale.

La définition de ce critère permet de lever l'ambiguïté lorsqu'un point est situé dans plusieurs ellipses simultanément, mais elle n'assure pas une bonne affectation. Par exemple, sur la figure 2.13, le point p_1

est apparié avec le segment e_2 ce qui est convenable, mais l'appariement du point p_2 avec le segment e_3 n'est pas exact. Etant donnée la direction de tir du capteur donnant le point d'impact 2, il est plus vraisemblable que la réflexion ait été provoquée par la surface e_2 , plutôt que par la surface e_3 . Nous devons donc également considérer l'angle d'incidence de l'axe acoustique sur la surface perçue afin de juger de la pertinence d'un appariement.

2.6.1.3 Estimation de l'angle d'incidence

(a) **Présentation du problème** Lorsque le capteur a une direction de tir normale au segment qu'il détecte, l'onde réfléchi couvre la position du capteur, et la mesure renvoyée par le capteur correspond exactement à la distance qui le sépare de l'objet (figure 2.9). Si la direction de tir du capteur s'écarte de la normale, alors, le cône réfléchi se décale (figure 2.14). Plus cet angle est important plus le capteur sera en marge du cône de l'onde réfléchi. Lorsque l'angle d'incidence devient supérieur au demi-angle d'ouverture du cône d'émission du capteur (α), alors le capteur sort de la région de l'onde réfléchi et ne détecte donc pas l'obstacle (figure 2.15).

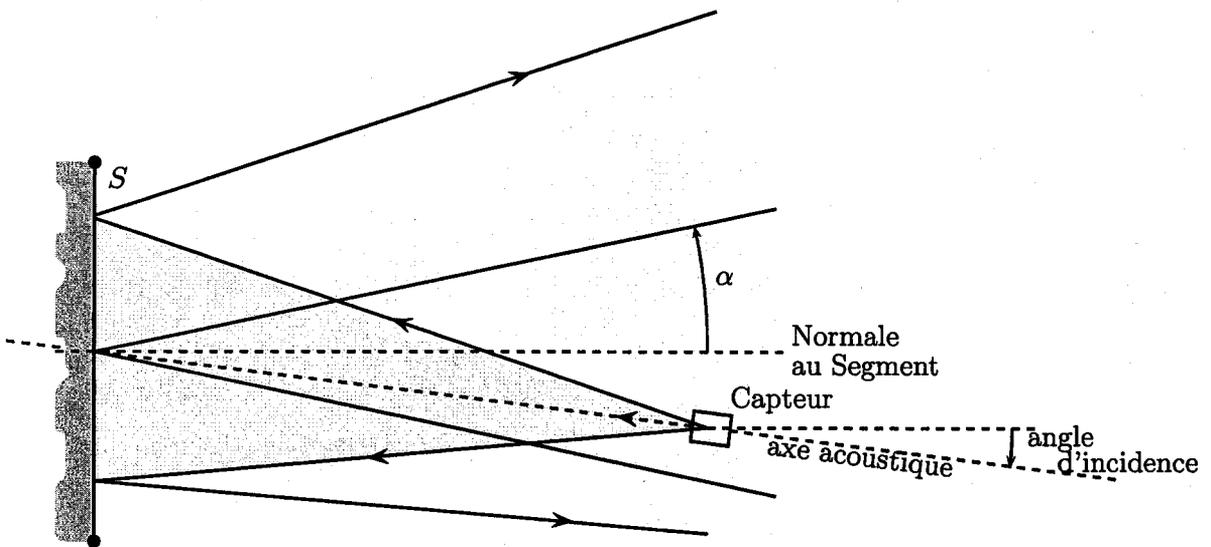


Figure 2.14 – Incidence inférieure à l'incidence limite de l'onde acoustique sur une surface.

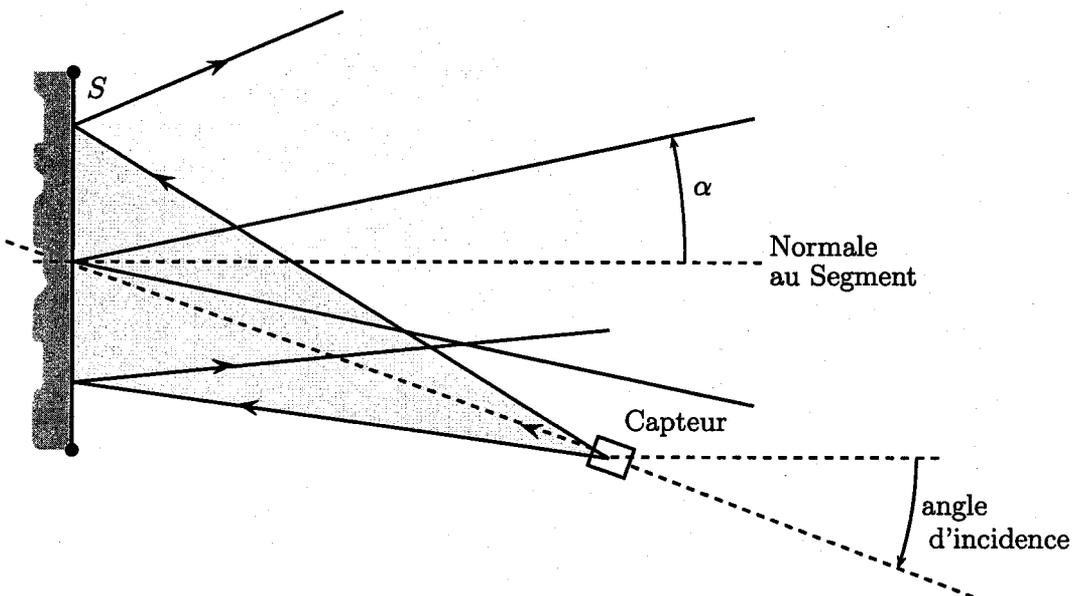


Figure 2.15 – Incidence supérieure à l'incidence limite de l'onde acoustique sur une surface.

En effectuant le test de l'angle d'incidence, nous pouvons valider une mise en correspondance entre point et segment et éviter le problème illustré par la figure 2.13. Dans ce cas le point d'impact p_2 ne peut pas correspondre au segment e_3 , étant donné que l'angle d'incidence de l'onde du capteur à ultrasons sur cette surface est de l'ordre de $-\pi/2$. Ce point correspond à la détection de la surface modélisée par le segment e_2 .

(b) **Mise en œuvre** Pour chaque appariement nous déterminons l'angle que fait l'axe acoustique du capteur avec la normale au segment. Si celui-ci est supérieur à la demi-ouverture du cône de détection du capteur à ultrasons, alors l'appariement est rejeté, sinon il est validé.

Ce second test de validité, placé en amont de la discrimination par distance point-segment, conduit à l'algorithme A.2 donné en annexe A.

2.6.1.4 Illustration d'une mise en correspondance

La figure 2.16 montre un exemple de projection d'une carte locale sur la carte globale autour de la localisation estimée du robot. Dans cet exemple les points 4 à 11 sont associés à des segments car ils se situent dans des ellipses et que leurs angles d'incidences sont inférieurs à l'angle d'incidence limite. Ce qui n'est pas le cas pour le reste des points qui ne seront donc pas utilisés pour le calcul des transformations.

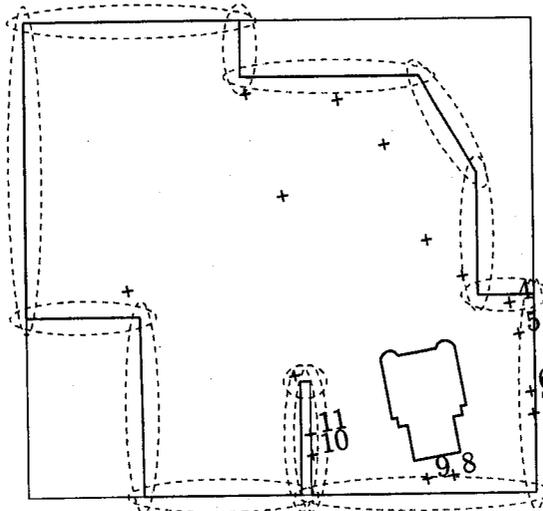


Figure 2.16 – Exemple de projection d'une carte locale dans une carte globale.

2.6.1.5 Problème lié à la diaphonie

Lorsqu'une onde ultrasonore rencontre un obstacle, l'onde renvoyée ne représente qu'une partie du signal initial. Le reste de l'onde acoustique est soit émise dans une autre direction soit absorbée par la surface suivant le type de matériau et l'angle d'incidence [LoP02]. Lorsqu'aucun signal n'est renvoyé dans la direction du capteur ayant émis le signal, nous parlons de réflexion spéculaire. Ceci a lieu, comme nous l'avons vu au paragraphe précédent, lorsque l'angle d'incidence dépasse une valeur limite voisine de la demi-ouverture du cône d'émission du capteur. Le signal renvoyé par la surface rencontrée est alors totalement réfléchi dans une direction autre que celle du capteur ayant émis l'onde. Dans des environnements fortement encombrés, l'onde peut parcourir un chemin tel qu'elle soit perçue par un autre capteur, faussant ainsi la mesure effectuée par le second capteur. Ce phénomène est connu sous le nom de diaphonie (ou crosstalk).

Dans ce paragraphe nous allons traiter du problème des réflexions multiples qui donnent lieu à des mesures de distances erronées. Dans certains cas elles peuvent fournir un point du modèle local compatible avec un des segments du modèle global, car la configuration point/segment validerait les critères utilisés. Cette association serait préjudiciable aux résultats de la procédure de recalage. Des considérations sur le parcours de l'onde ultrasonore peuvent mettre en évidence ce phénomène.

(a) **Présentation du problème** Si les capteurs 1 et 2 de la figure 2.17 sont synchrones, ils sont alors en mode de réception et d'émission en même temps. Comme on peut le voir sur cette figure, l'onde émise

par le capteur 1 est réfléchi à plusieurs reprises (②, ③, ④) et recouvre le capteur 2. Si l'onde réfléchie du capteur 1 arrive au niveau du capteur 2 lorsque ce dernier est en mode réception, alors il peut percevoir une onde réfléchie qui ne correspond pas à la détection d'un obstacle réel. De ce fait la mesure renvoyée par le capteur (mesure perturbée de la figure 2.17) est différente de la distance réelle.

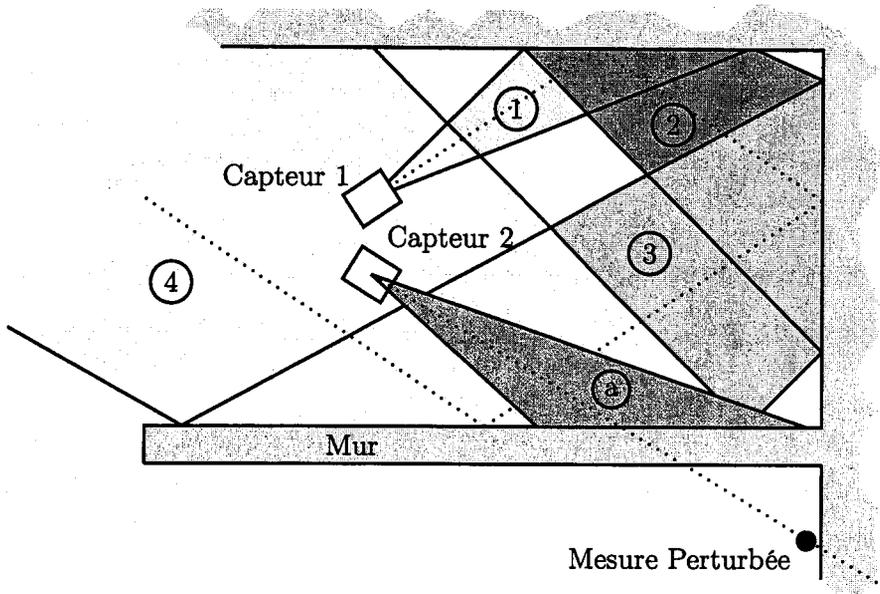


Figure 2.17 – Problème des réflexions multiples.

Ce type de réflexions peut poser des problèmes lors de la phase d'appariement. Par exemple sur la figure 2.18 le capteur 10 a une incidence et une distance mesurée qui placent le point d'impact en une position vérifiant les critères utilisés. Or cette mesure ne peut pas être appariée avec le segment dont l'ellipse englobe son point d'impact car l'axe acoustique croise un mur de l'environnement. Dans ce type de configuration la diaphonie peut-être détectée en relevant une telle invraisemblance.

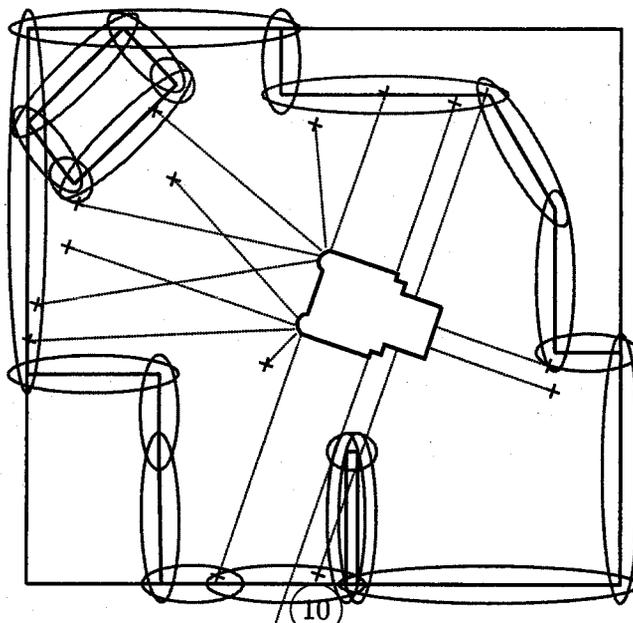


Figure 2.18 – Exemple réel de réflexions multiples provoquant une ambiguïté d'appariement.

(b) **Solution et mise en œuvre** Nous mettons en place un test permettant d'éliminer certaines mesures liées aux phénomènes de diaphonie. Avant d'apparier le point du modèle local au segment du modèle global, nous nous assurons qu'aucun obstacle de l'environnement n'est situé entre le capteur et le point. En présence d'un tel obstacle, l'appariement n'est pas réalisé. Cela nous conduit à une nouvelle

version de la phase d'appariement, qui inclut tous les tests précédemment présentés. Elle est donnée en annexe A par l'algorithme A.3 et la figure 2.19 en donne une représentation sous forme d'organigramme. La réalisation pratique du test de cohérence est donnée en annexe B (page 143). Ce test permet aussi d'éviter les appariements ambigus qui peuvent apparaître lors de la présence d'obstacles de petite taille (figure 2.20).

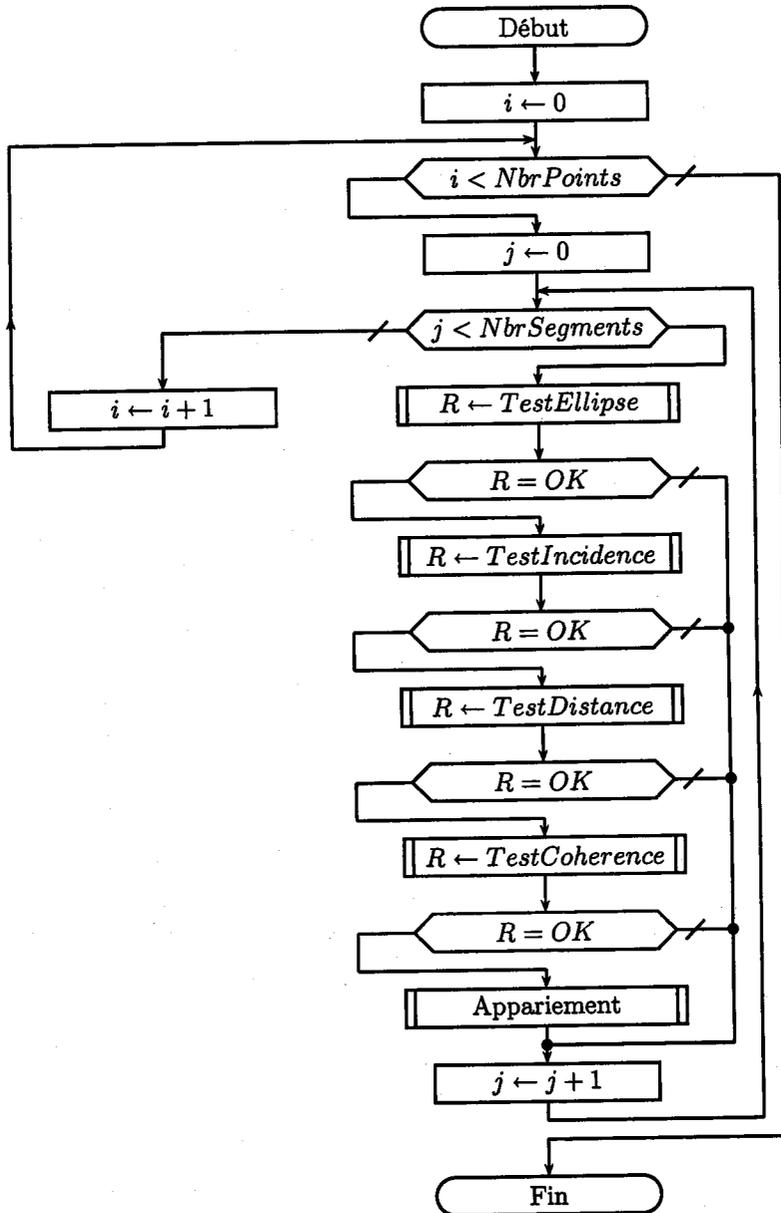


Figure 2.19 – Organigramme de la version complète de l'algorithme d'appariement des point du modèle local avec les segments du modèle global.

Par exemple dans le contexte de la figure 2.20, si nous considérons uniquement la distance qui sépare le point p_1 des segments s_1 et s_3 , alors nous appariersons le point au segment s_3 , ce qui est inexact puisque celui-ci n'est pas perceptible par le capteur étant donnée la position de ce dernier. Le point serait donc apparié avec "la face cachée" de l'obstacle. Ce type de problème n'est pas seulement dû aux erreurs de mesure des capteurs à ultrasons mais provient également de l'imprécision de l'estimation odométrique de la localisation du robot. Comme la position estimée du véhicule est utilisée pour projeter les mesures des capteurs à ultrasons dans le repère global, si celle-ci est erronée, alors la projection est perturbée et peut engendrer des situations comme celles de la figure 2.20.

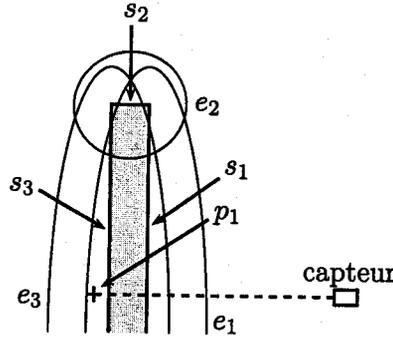


Figure 2.20 – Exemples d'appariements ambigus.

2.6.2 Calcul de la transformation

Dans ce paragraphe nous allons traiter le calcul des transformations en position et en orientation à partir des couples point/segment. La figure 2.21 donne un exemple d'un point P mis en correspondance avec le segment S . La distance l_{PS} qui sépare le point P du segment S est calculée à l'aide de la relation 2.5 du paragraphe 2.6.1.2, que nous pouvons aussi écrire :

$$l_{PS} = [\sin(\alpha_S) \quad -\cos(\alpha_S)] \left[\begin{pmatrix} x_P \\ y_P \end{pmatrix} \right] + \rho_S \quad (2.6)$$

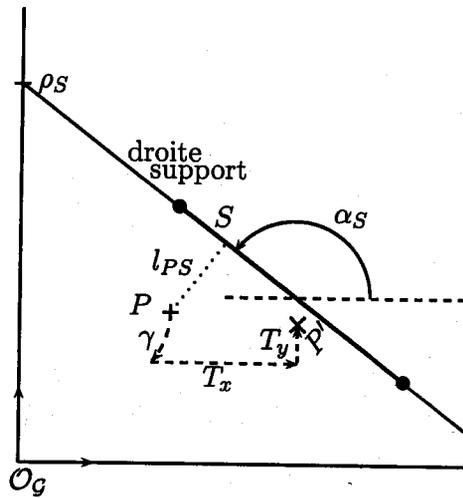


Figure 2.21 – Représentation des Transformations appliquées au point P .

Nous cherchons la transformation qui, appliquée à la position du robot, permet de minimiser la distance l_{PS} . En observant l'équation 2.6, nous remarquons que si $l_{PS} = 0$, le point d'impact est situé exactement sur le segment du modèle et satisfait l'équation 2.1 de la droite.

Comme la position estimée du robot n'est pas exacte, il en est de même pour celle du point d'impact : $l_{PS} \neq 0$. En appliquant une transformation composée d'une rotation d'angle γ et d'une translation T à la localisation estimée du robot, le point P subit la même transformation et se déplace en P' . On peut alors calculer la nouvelle distance $l_{P'S}$ qui sépare le point P' du segment S , elle s'écrit :

$$l_{P'S} = [\sin(\alpha_S) \quad -\cos(\alpha_S)] \left[\begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & T_x \\ \sin(\gamma) & \cos(\gamma) & T_y \end{pmatrix} \begin{pmatrix} x_P \\ y_P \\ 1 \end{pmatrix} \right] + \rho_S \quad (2.7)$$

Pour ramener le point P' sur la droite, nous devons obtenir $l_{P'S} = 0$, dans ce cas en soustrayant l'équation 2.6 à l'équation 2.7, il vient :

$$-l_{PS} = [\sin(\alpha_S) \quad -\cos(\alpha_S)] \left[\begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & T_x \\ \sin(\gamma) & \cos(\gamma) & T_y \end{pmatrix} \begin{pmatrix} x_P \\ y_P \\ 1 \end{pmatrix} - \begin{pmatrix} x_P \\ y_P \end{pmatrix} \right] \quad (2.8)$$

en supposant que la rotation d'angle γ soit de faible amplitude, ce qui signifie que l'erreur en orientation est de faible amplitude, nous pouvons écrire :

$$-l_{PS} = [\sin(\alpha_S) \quad -\cos(\alpha_S)] \left[\begin{pmatrix} 1 & -\gamma & T_x \\ \gamma & 1 & T_y \end{pmatrix} \begin{pmatrix} x_P \\ y_P \\ 1 \end{pmatrix} - \begin{pmatrix} x_P \\ y_P \end{pmatrix} \right] \quad (2.9)$$

$$-l_{PS} = [\sin(\alpha_S) \quad -\cos(\alpha_S)] \left[\begin{pmatrix} 1 & -\gamma \\ \gamma & 1 \end{pmatrix} \begin{pmatrix} x_P \\ y_P \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} - \begin{pmatrix} x_P \\ y_P \end{pmatrix} \right] \quad (2.10)$$

Nous obtenons finalement :

$$-l_{PS} = [\sin(\alpha_S) \quad -\cos(\alpha_S)] \left[\gamma \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_P \\ y_P \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \right] \quad (2.11)$$

L'équation 2.11 permet de calculer la transformation qui fait parcourir la distance l_{PS} au point P afin que celui-ci se place sur le segment S . Une équation de la forme de celle de 2.11 est définie pour chaque couple qui associe un point de la carte locale à un segment de la carte globale. Nous devons donc résoudre un système surdéterminé de $NbApp$ équations à trois inconnues, où $NbApp$ est le nombre d'associations point-/segment. Le système ne possède pas de solution, il est donc impossible de placer tous les points sur leurs segments. Nous cherchons donc, par la méthode des moindres carrés, une estimation de la transformation composée de la rotation d'angle γ et de la translation $T : \{T_x, T_y\}$ qui minimise la somme des nouvelles

distances au carré, soit $\sum_{i=0}^{NbApp} l_{(P'S)_i}^2$.

En définissant :

$$\begin{cases} Y_i = -l_{(PS)_i} \\ X_{i,1} = \sin(\alpha_{S_i}) \\ X_{i,2} = -\cos(\alpha_{S_i}) \\ X_{i,3} = (\cos(\alpha_{S_i}) \quad \sin(\alpha_{S_i})) \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_{P_i} \\ y_{P_i} \end{pmatrix} \\ b = \begin{pmatrix} T_x \\ T_y \\ \gamma \end{pmatrix} \end{cases} \quad \text{Transformation à appliquer} \quad (2.12)$$

où $i \in [1, \dots, NbApp]$, Y est de dimensions $[NbApp \times 1]$ et X de $[NbApp \times 3]$.

Le système surdéterminé, s'exprime donc par :

$$Y = Xb \quad (2.13)$$

et la solution \hat{b} est définie au sens des moindres carrés par $\sum_{i=0}^{NbApp} \varepsilon_i^2$ minimal avec $\varepsilon_i = Y_i - X_i \hat{b}$.

\hat{b} se calcule par :

$$\hat{b} = (X^T X)^{-1} X^T Y \quad (2.14)$$

Une fois le vecteur \hat{b} calculé, nous l'appliquons à la position estimée du robot et à tous les points du modèle local. L'algorithme est relancé (appariement + calcul de la transformation) jusqu'à ce que les trois composantes (T_x, T_y, γ) soient inférieures à leur seuil respectif. Etant donné que trois paramètres doivent être déterminés, un minimum de 3 mises en correspondance est nécessaire pour effectuer le calcul de la transformation. Afin de limiter le temps de calcul, nous fixons le nombre maximal d'itérations à 7 pour le calcul de la transformation globale (somme des transformations calculées à chaque itération).

En pratique, l'inversion de matrice est réalisée en utilisant la méthode de Gréville (donnée en annexe C), qui permet d'obtenir l'inverse de la matrice, ou bien sa pseudo-inverse dans le cas d'une matrice non inversible. Nous pouvons ainsi calculer une transformation quelle que soit la situation.

2.7 Mise en œuvre en simulation

2.7.1 Contexte

Dans cette partie, nous allons évaluer le comportement de la méthode de correction en utilisant des données capteurs simulées pour l'odomètre comme pour les mesures retournées par les capteurs à ultrasons.

Après un déplacement de l'ordre de 50 cm , l'erreur en position est de l'ordre de $\pm 15\text{ cm}$ en abscisse et en ordonnée et l'erreur en orientation est de $\pm 10^\circ$. Comme nous l'avons défini au début du chapitre (§2.3), l'objectif de la correction est d'obtenir une erreur en position inférieure à 5 cm et une erreur en orientation inférieure à 3° . Lorsque ces conditions seront vérifiées la transformation délivrée par l'algorithme de recalage aura ses composantes inférieures à ces seuils. La condition d'arrêt de l'algorithme sera donc : $T_x < 5\text{ cm}$ et $T_y < 5\text{ cm}$ et $\gamma < 3^\circ$.

2.7.2 Exemple de simulation dans un environnement simple

L'environnement utilisé pour la simulation a été choisi simple, pour faciliter l'explication de son déroulement. La figure 2.22 présente la position et l'orientation du robot dans l'environnement lors de la simulation des mesures des capteurs à ultrasons. Le robot est positionné en $(250\text{ cm}, 170\text{ cm})$ et a une orientation de 0° . Dans la suite, nous ne précisons plus l'unité de la position et de l'orientation, et considérons qu'elles sont, respectivement, toujours exprimées en centimètres et en degrés.

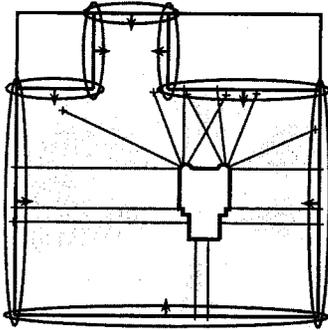


Figure 2.22 – Position réelle du robot dans l'environnement - Position $(250, 170, 0)$

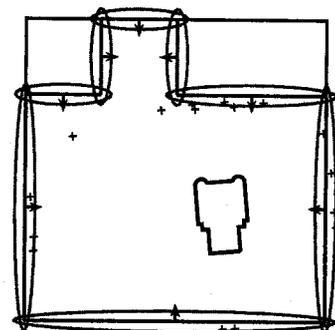


Figure 2.23 – Position odométrique du robot dans l'environnement - Position $(260, 160, 5)$

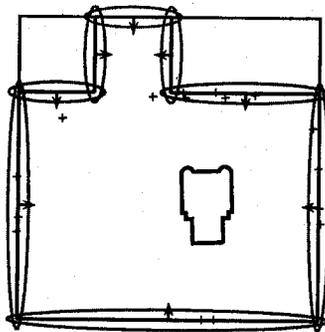


Figure 2.24 – Position corrigée du robot dans l'environnement - Position $(249, 171, 2)$

La figure 2.23 présente la carte locale, établie à partir des données capteurs simulées, projetée autour de la localisation estimée du robot qui est $(260, 160, 5)$.

L'exécution de l'algorithme de correction (appariement et calcul de la transformation), conduit au résultat de la figure 2.24 en 3 itérations. L'estimation de la localisation du robot est alors de $(249, 171, 2)$. Les erreurs sont bien en dessous des valeurs maximales que nous nous sommes fixées. En position l'erreur est de 1 cm sur chaque axe, et l'erreur en orientation est de 2° .

La première itération de l'algorithme apparie 9 points d'impact avec le modèle. La somme des distances points segments au carré normalisée (calculée en divisant la somme par le nombre d'appariements utilisés)

est de 106.5 cm^2 . La transformation calculée est alors de $(-16, 20, -2)$, ce qui donne une nouvelle estimation de la localisation en $(244, 180, 3)$. Lors de la seconde itération, le nombre d'appariements a augmenté et est passé à 11. La somme des distances au carré normalisée a diminué, et vaut maintenant 64.9 cm^2 . Les points du modèle local coïncident donc mieux avec les segments du modèle global. La transformation calculée lors de cette itération est de $(4, -7, -1)$, ce qui donne une nouvelle estimation de la localisation en $(248, 173, 2)$. La troisième et dernière itération apparie toujours 11 points d'impact, la somme des erreurs au carré normalisée à minimiser est de 9.7 cm^2 . L'écart entre les deux modèles est de plus en plus faible. La transformation est maintenant de $(2, -3, 0)$ et la position estimée finale du robot est de $(250, 170, 2)$.

Cet exemple de simulation donne une estimation correcte de la localisation pour une erreur initiale de l'ordre de celles observées lors des déplacements du véhicule. L'erreur résiduelle persistante en orientation est due à l'incertitude inhérente au modèle employé pour représenter les mesures des capteurs à ultrasons (§2.5.2.1).

2.8 Mise en œuvre sur le prototype

Le comportement de la méthode de correction est maintenant évalué en utilisant des mesures prises avec le prototype du VAHM dans un environnement d'intérieur reconstitué. Les positions odométriques seront simulées afin d'évaluer plusieurs corrections avec un même jeu de données. La figure 2.25 présente une vue de dessus de l'environnement dans lequel les mesures ont été réalisées, ainsi que les différentes positions du robot dans cet environnement. Pour une position donnée dans l'environnement, nous enregistrons les seize valeurs retournées par les capteurs à ultrasons ainsi que la position et l'orientation au sol du véhicule. Ces positions et orientations nous serviront de références lors de la phase d'évaluation de la méthode; nous comparerons les localisations corrigées avec ces valeurs.

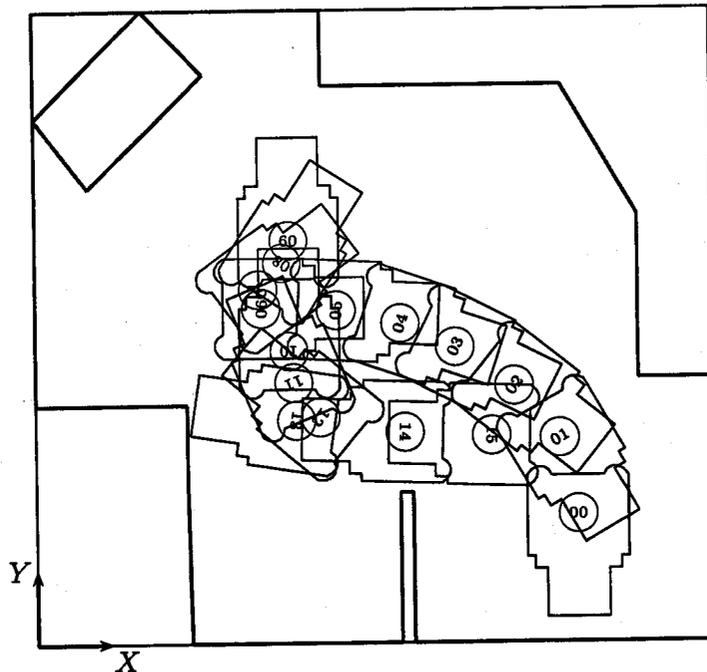


Figure 2.25 – Environnement utilisé pour les essais de la méthode de correction dynamique avec les différentes positions du véhicule.

2.8.1 Conditions d'évaluations

Au paragraphe 2.3, nous avons vu que l'odomètre accumule des erreurs telles que l'estimation de la position peut être située dans un carré d'environ 30 cm de côté autour de la position réelle et que l'orientation se situe dans un cône de 20° d'ouverture centré autour de l'orientation réelle du véhicule.

L'objectif de cette étude est d'évaluer la méthode de correction pour n'importe quelle valeur de l'erreur

d'estimation odométrique située dans la zone déterminée au paragraphe 2.3. Afin de tester l'ensemble des erreurs possibles, nous effectuons le traitement hors ligne et utilisons la position relevée au sol ainsi que le carré définissant la zone des erreurs odométriques afin de définir les positions qui seront utilisées pour initialiser la méthode de correction dynamique. De façon similaire, nous définissons pour l'orientation odométrique une région à partir de l'orientation relevée au sol et du cône de 20° . L'ensemble des triplets utilisés pour simuler la localisation qui devrait être fournie par l'odomètre se situe dans une région conique – pour l'orientation – combinée à une région carrée – pour la position (figure 2.26). Le pas de variation sur x et y est de 1 cm et de 1° pour l'orientation. Sur la figure 2.26 apparaissent également la région carrée de 10 cm de côté et la région conique de 6° , qui définissent la zone dans laquelle une correction est jugée acceptable. Elles représentent les erreurs limites de 5 cm en position et de 3° en orientation que nous nous sommes fixées.

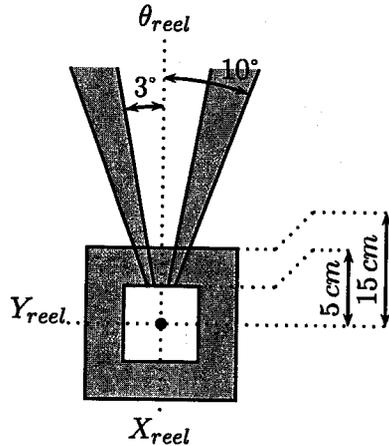


Figure 2.26 – Région d'initialisation odométrique.

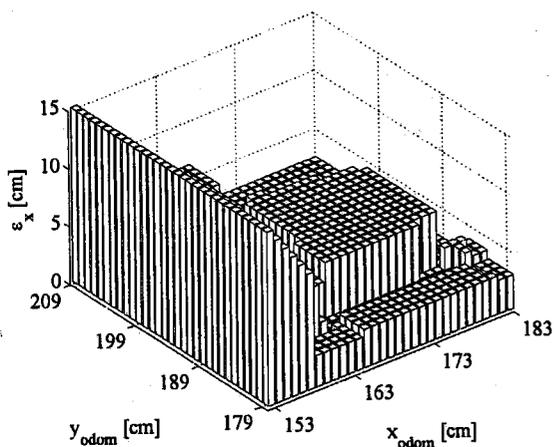
2.8.2 Mode de présentation des résultats

2.8.2.1 Représentation des erreurs de localisation en fonction de la position odométrique

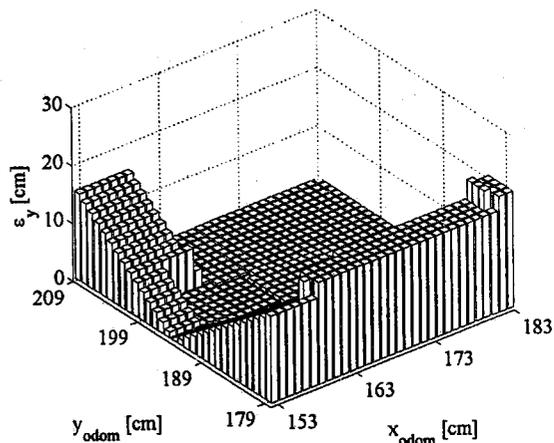
Pour une erreur odométrique en orientation donnée, nous représentons les erreurs absolues sur X , Y et θ après correction en fonction de la position odométrique. C'est-à-dire que nous traçons ε_x en fonction de (X_{odom}, Y_{odom}) (figure 2.27(a)) ainsi que ε_y (figure 2.27(b)) et ε_θ (figure 2.27(c)). Ces erreurs sont calculées en utilisant les valeurs prises comme références, relevées au sol lors des acquisitions, elles sont données par : $\varepsilon_x = |X_{reel} - X_{corrige}|$, $\varepsilon_y = |Y_{reel} - Y_{corrige}|$, $\varepsilon_\theta = |\theta_{reel} - \theta_{corrige}|$. Sur les figures 2.27(a) à 2.27(c) les axes X_{odom} et Y_{odom} indiquent les positions odométriques utilisées pour les essais. La coordonnée centrale du graphique correspond à la position réelle du fauteuil. Ces trois représentations permettent d'apprécier l'efficacité de la méthode de correction en fonction de la position odométrique et ainsi identifier ses limites de fonctionnement dans une configuration donnée. Etant donné que nous faisons varier l'orientation odométrique dans une plage de $\pm 10^\circ$ autour de l'orientation réelle par pas de 1° , pour une localisation réelle donnée, nous possédons vingt et un ensembles de trois courbes.

2.8.2.2 Représentation des erreurs de localisation en fonction de l'orientation odométrique

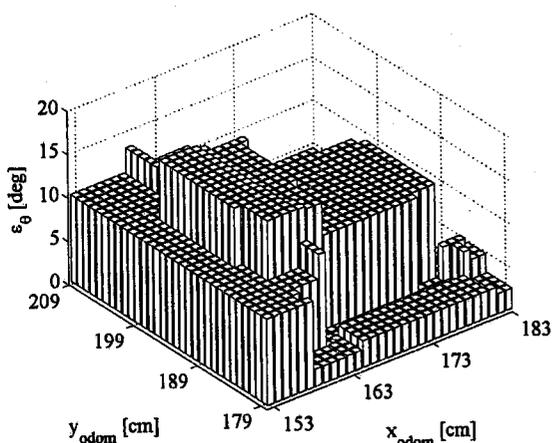
De façon similaire à la représentation précédente, nous affichons pour une position odométrique donnée l'évolution des trois erreurs présentées précédemment en fonction de la valeur de l'orientation odométrique, θ_{odom} (figure 2.27(d)). Comme la région d'initialisation de la position odométrique est un carré de 900 cm^2 et que le pas de variation est de 1 cm , nous obtenons 900 figures possibles pour une localisation réelle du robot dans l'environnement. Sur les représentations de ε_x et ε_y apparaissent les seuils $+5\text{ cm}$ et -5 cm entre lesquels doit se situer l'erreur si la correction de la variable x ou y s'est correctement déroulée. Il en est de même pour la variable ε_θ avec des seuils en $+3^\circ$ et -3° .



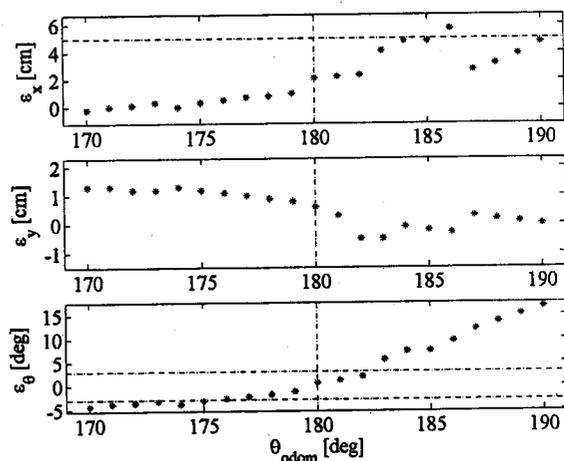
(a) $\epsilon_x = f(x_{odom}, y_{odom})$ pour θ_{odom} donnée.



(b) $\epsilon_y = f(x_{odom}, y_{odom})$ pour θ_{odom} donnée.



(c) $\epsilon_\theta = f(x_{odom}, y_{odom})$ pour θ_{odom} donnée.



(d) $\epsilon_{x,y,\theta} = f(\theta_{odom})$ pour (x_{odom}, y_{odom}) donnée.

Figure 2.27 – Résultats de correction en fonction de la localisation odométrique.

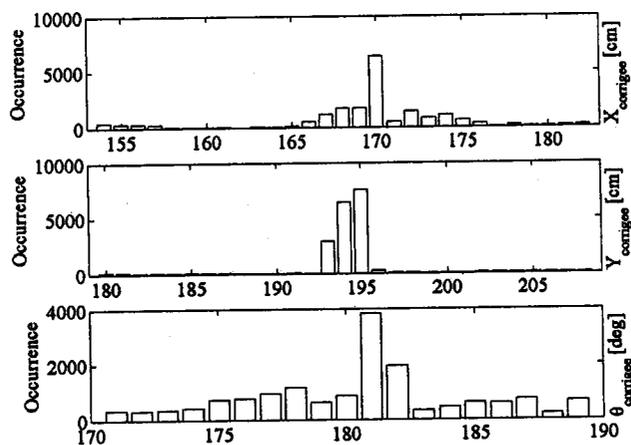


Figure 2.28 – Histogrammes des résultats de la correction pour une localisation réelle.

2.8.2.3 Histogrammes des valeurs corrigées

Nous avons également représenté les histogrammes des valeurs de la localisation corrigée $X_{corrigee}$, $Y_{corrigee}$ et $\theta_{corrigee}$ résultant de toutes les localisations odométriques initiales pour une localisation réelle donnée (figure 2.28). Ces représentations indiquent si les valeurs corrigées convergent toujours vers les mêmes valeurs et si ces valeurs sont proches des valeurs réelles (X_{reel} , Y_{reel} , θ_{reel}). Etant donné que nous utilisons 961 positions odométriques et que pour chaque position odométrique nous simulons 21 orientations odométriques nous obtenons 20181 conditions initiales pour une localisation réelle du véhicule.

2.8.3 Présentation de quelques résultats

2.8.3.1 Etude de la position 00

La localisation 00 de la trajectoire d'essai réalisée avec le véhicule est située en (359, 85, 0). Quelle que soit la valeur choisie pour la localisation odométrique simulée, nous obtenons une correction qui converge en moyenne vers la solution (359, 87, -2) avec une très faible dispersion (figure 2.29). En effet, les trois histogrammes possèdent un seul mode. Pour x , il est centré sur la valeur réelle et a une faible dispersion. En y , ce mode est légèrement biaisé par rapport à la position réelle (+2 cm). En orientation le biais est de -2° , ce qui est proche de la limite tolérée pour l'erreur en orientation et il possède une dispersion plus importante mais qui ne dépasse pas le seuil admis.

Les figures 2.30(a) à 2.30(c) donnent un exemple de répartition de l'erreur sur chaque composante de la localisation après correction en fonction de la position odométrique pour une orientation odométrique simulée initialisée à $+10^\circ$ (l'orientation réelle étant de 0°). Les erreurs en position (ε_x et ε_y) et en orientation (ε_θ) après correction satisfont quasiment toujours les objectifs fixés pour la localisation. Les quelques cas particuliers donnant des erreurs en orientation et en position supérieures aux seuils fixés sont induits par des positions odométriques qui ne permettent pas d'apparier au moins trois points de la carte locale avec les segments de la carte globale lors de la première itération de l'algorithme. Sur les figures 2.30(a) à 2.30(c) ces cas sont situés à la périphérie de la zone d'initialisation (pour $y_{odom} = 100$ cm et $372 \leq x_{odom} \leq 374$ cm). Dans ces cas la position et l'orientation ne sont pas réajustées. Nous remarquons que d'autres positions odométriques ne conduisent pas à une correction satisfaisante en x , il s'agit de configurations dans lesquelles les capteurs ne perçoivent pas de surfaces dans cette direction (comme par exemple dans un couloir). Le même problème est rencontré pour la variable y , lorsque $y_{odom} = 70$ cm. Dans les cas où l'erreur en position est amplifiée, elle reste inférieure à 15 cm. La figure 2.30(d) donne une représentation de l'influence de l'erreur en orientation initiale pour une position odométrique donnée. Les erreurs en position et en orientation après correction restent bien inférieures à leur seuil maximal respectif. Mis à part les quelques cas où la correction n'a pas correctement abouti, l'algorithme a fourni, lors de cette évaluation, des résultats satisfaisants. De plus, la qualité des résultats obtenus dans cet essai est améliorée grâce à la présence de nombreux segments orthogonaux à l'axe des capteurs ce qui induit une grande précision du modèle.

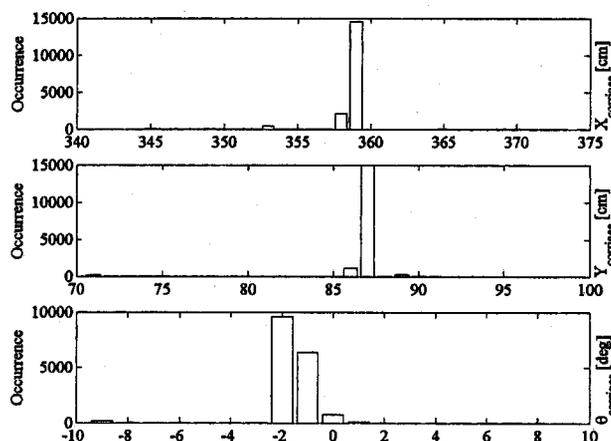
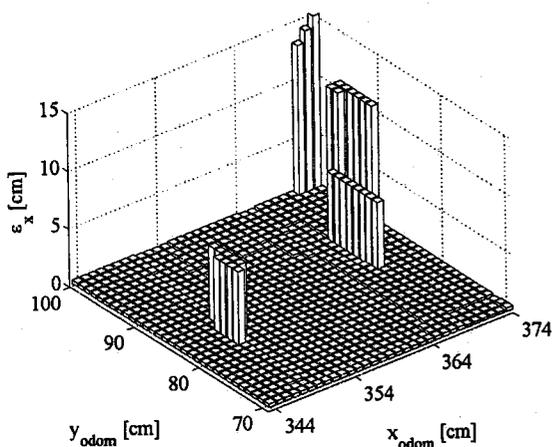
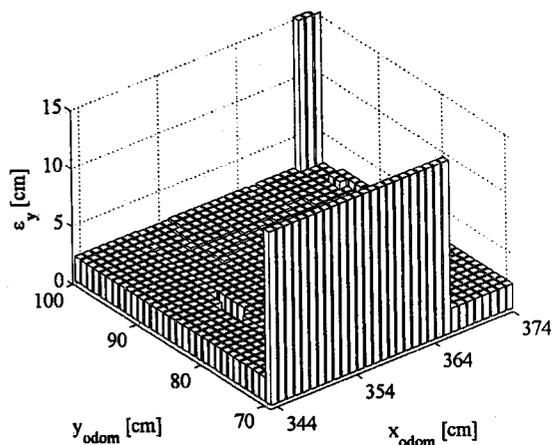


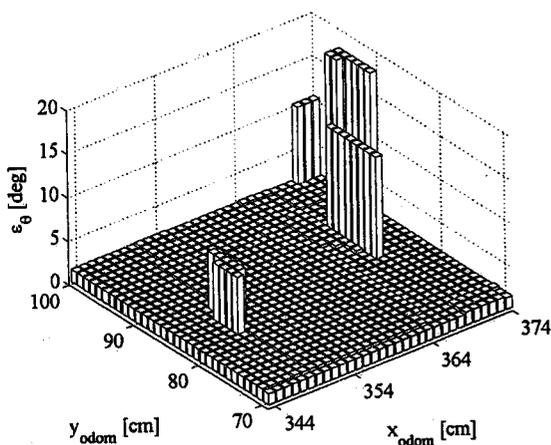
Figure 2.29 – Position réelle 00 (359, 87, -2) : Histogrammes des résultats de la correction.



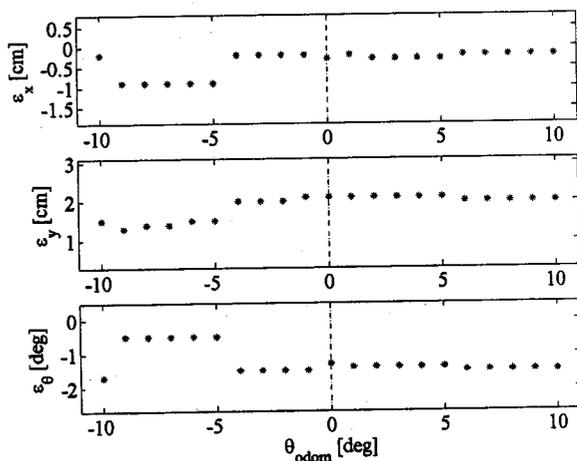
(a) $\varepsilon_x = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = +10^\circ$.



(b) $\varepsilon_y = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = +10^\circ$.



(c) $\varepsilon_\theta = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = +10^\circ$.



(d) $\varepsilon_{x,y,\theta} = f(\theta_{odom})$ pour $(x_{odom}, y_{odom}) = (351, 77)$.

Figure 2.30 – Position réelle 00 (359, 87, -2) : Résultats de correction en fonction de la localisation odométrique.

2.8.3.2 Etude de la position 03

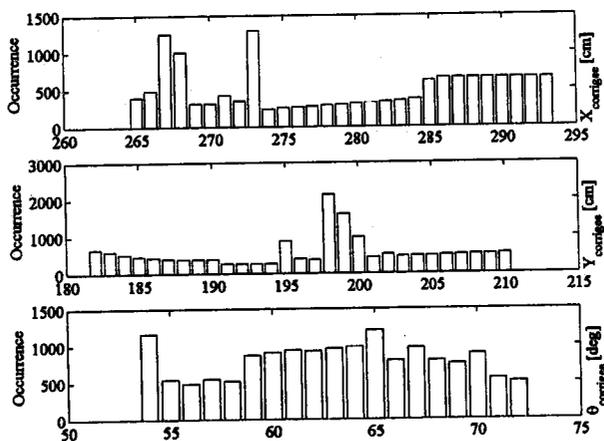
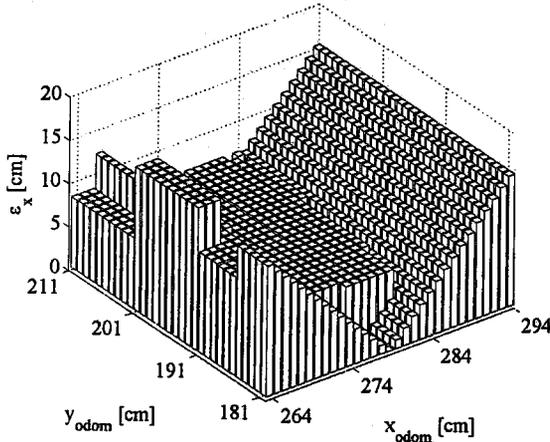
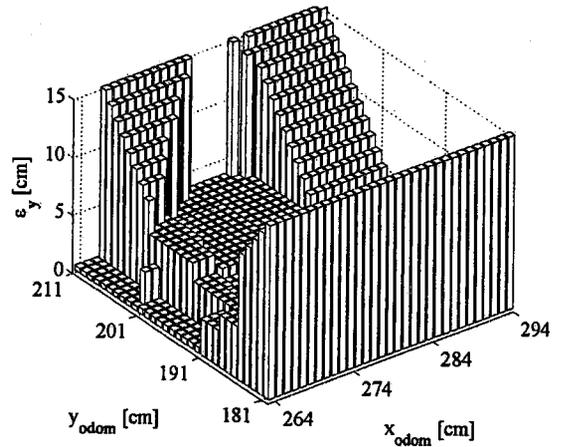


Figure 2.31 – Position réelle 03 (279, 196, 63) : Histogrammes des résultats de la correction.

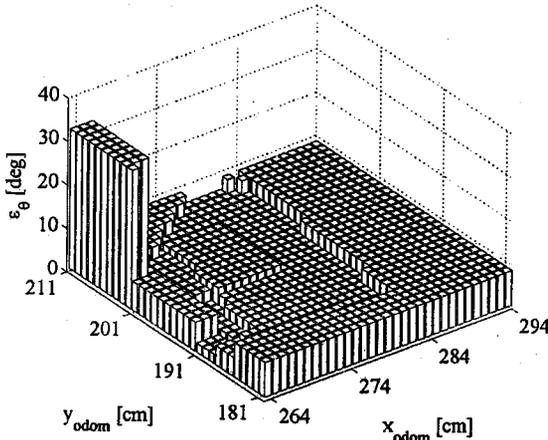
Lors de l'essai 03 le véhicule était situé en (279,196,63). Pour cette localisation réelle, nous remarquons que la méthode de correction a du mal à converger vers une solution unique (figure 2.31), les trois histogrammes sont assez dispersés. La répartition des valeurs de x après correction présente deux modes qui ne sont pas très marqués et dont un possède une erreur de position supérieure aux 5 cm limites (celui situé en 268). La correction de la composante y donne un seul mode, alors que la correction en orientation ne possède pas de mode et est fortement dispersée autour de la valeur réelle.



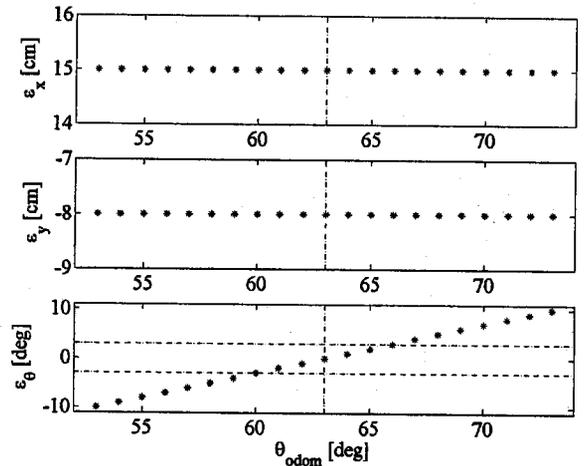
(a) $\varepsilon_x = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 71^\circ$.



(b) $\varepsilon_y = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 71^\circ$.



(c) $\varepsilon_\theta = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 71^\circ$.



(d) $\varepsilon_{x,y,\theta} = f(\theta_{odom})$ pour $(x_{odom}, y_{odom}) = (294, 188)$.

Figure 2.32 – Position réelle 03 (279,196,63) : Résultats de correction en fonction de la localisation odométrique.

Les figures 2.32(a) à 2.32(c) donnent la répartition de l'erreur de localisation après correction en fonction de la position odométrique pour une orientation odométrique simulée initialisée à 71° (l'orientation réelle étant de 63°). Il apparaît qu'un grand nombre de positions odométriques ne permettent pas une exécution de la correction car le nombre de points appariés est insuffisant. Ces cas sont identifiables sur les représentations par les régions où l'erreur sur les variables avant et après correction reste la même, c'est-à-dire les régions donnant l'impression d'une "pente". Pour ces positions odométriques les angles d'incidence des ondes des capteurs à ultrasons sur les surfaces étaient proches de l'angle limite. Alors en raison de l'erreur sur l'orientation ces angles paraissent plus grands qu'ils ne le sont et le test d'incidence de l'algorithme les rejettent. Quelques initialisations permettent tout de même d'obtenir un nombre suffisant d'appariements pour réaliser la correction. Dans ces cas, seule la correction en y parvient à une erreur d'estimation inférieure à 5 cm. L'erreur en x est de l'ordre de 6 cm et l'erreur en θ est réduite sauf pour certaines positions odométriques où elle atteint une valeur supérieure à 30° . Dans ces configurations odo-

métriques le nombre de points appariés était suffisant pour exécuter une correction mais il était insuffisant dans la direction des x . La figure 2.32(d) donne une représentation de l'influence de l'erreur en orientation initiale pour la position odométrique de (294, 188). Nous constatons pour cette position odométrique que, quelle que soit l'orientation odométrique utilisée, la correction est impossible en raison d'un manque de points appariés.

2.8.3.3 Etude de la position 05

Pour l'essai 05 nous avons placé le véhicule en (200, 221, 87). La figure 2.33 montre les répartitions des valeurs pour $X_{corrige}$, $Y_{corrige}$ et $\theta_{corrige}$. Pour cette position réelle, la répartition des solutions pour la valeur de $Y_{corrige}$ laisse apparaître un seul mode en $Y_{corrige} = 220 \text{ cm}$. La répartition de $Y_{corrige}$ possède une dispersion raisonnable puisque le mode reste compris dans un intervalle de $\pm 5 \text{ cm}$ autour de $Y_{reelle} = 221 \text{ cm}$. Pour $X_{corrige}$ nous pouvons observer quatre modes avec une prédominance pour celui situé en $X_{corrige} = 200 \text{ cm}$. La correction ne fournit pas toujours la même solution en fonction de la position odométrique utilisée. Le mode principal est contenu dans une zone de $\pm 5 \text{ cm}$ autour $X_{reelle} = 200 \text{ cm}$. Enfin, pour $\theta_{corrige}$ la répartition des solutions proposées est similaire à celle obtenue à l'essai 03. Dans ce cas, la méthode a du mal à corriger l'orientation du véhicule quelle que soit la localisation odométrique utilisée. Il n'y a aucun mode prédominant et la dispersion est de plus importante.

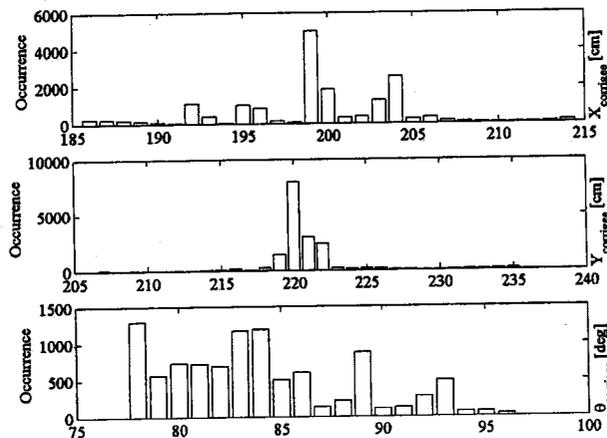


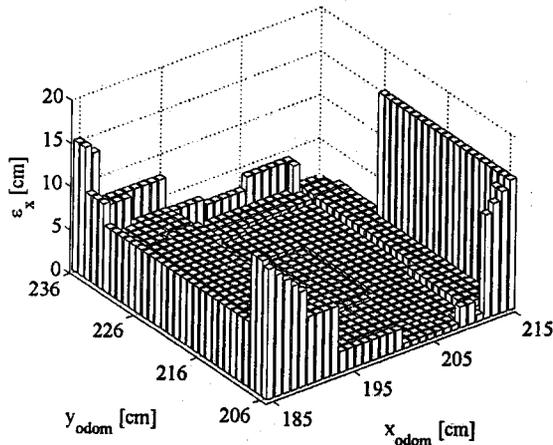
Figure 2.33 – Position réelle 05 (200, 221, 87) : Histogrammes des résultats de la correction.

Les figures 2.34(a) à 2.34(c) donnent la répartition de l'erreur de localisation après correction en fonction de la position odométrique pour une orientation odométrique égale à l'orientation réelle de 87° . Nous pouvons constater que la correction est correctement réalisée tant que l'erreur odométrique sur la position n'atteint pas les limites de la zone des $\pm 15 \text{ cm}$. A proximité de ces limites le nombre d'appariements diminue et est insuffisant dans la direction des x . La correction en orientation est de moins bonne qualité car l'objectif des 3° d'erreur est plus rarement atteint. Les figures 2.35(a) à 2.35(c) considèrent les cas pour une orientation odométrique de 83° . En position, la correction est sensiblement aussi bonne que pour une erreur d'orientation odométrique nulle, nous noterons tout de même une réduction de la zone de fonctionnement de la correction (l'épaisseur de la zone périphérique de correction inefficace a augmenté, surtout en x). Du point de vue de la correction en orientation, les résultats donnent rarement une erreur inférieure à 3° . Les figures 2.35(e) à 2.35(d) considèrent les cas pour une orientation odométrique de 91° . En position l'erreur après correction est plus importante que pour le cas d'une erreur odométrique en orientation nulle, mais n'excède pas la valeur de 5 cm lorsque la correction est correctement effectuée. Du point de vue de l'orientation, les résultats sont très affectés et atteignent des erreurs de l'ordre de 20° .

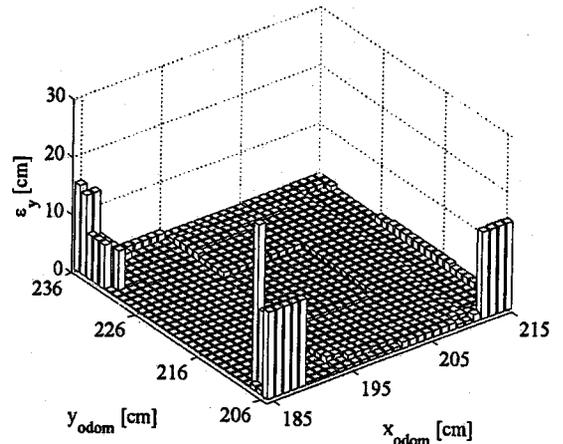
L'influence des configurations de l'environnement apparaît ici puisque pour des erreurs odométriques sur l'orientation d'amplitude égales mais de signes opposés, les résultats de la correction ne sont pas les mêmes et l'erreur sur x après correction sort plus souvent de la limite de validité pour $\theta_{odom} = 83^\circ$ que

pour $\theta_{odom} = 91^\circ$. Il en est de même pour l'erreur sur θ . L'erreur sur y après correction sort plus souvent de la limite de validité pour $\theta_{odom} = 91^\circ$ que pour $\theta_{odom} = 83^\circ$.

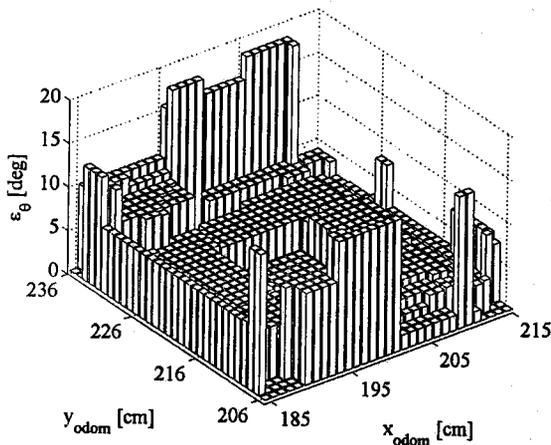
La figure 2.34(d) donne une représentation de l'influence de l'erreur en orientation initiale pour une position odométrique donnée. Nous constatons que l'erreur en y après correction reste bien inférieure au seuil des 5 cm. Pour x , à partir d'une certaine orientation odométrique, la correction n'est plus assurée en raison d'un manque de points appariés dans cette direction. La correction en θ est quant à elle rarement satisfaisante.



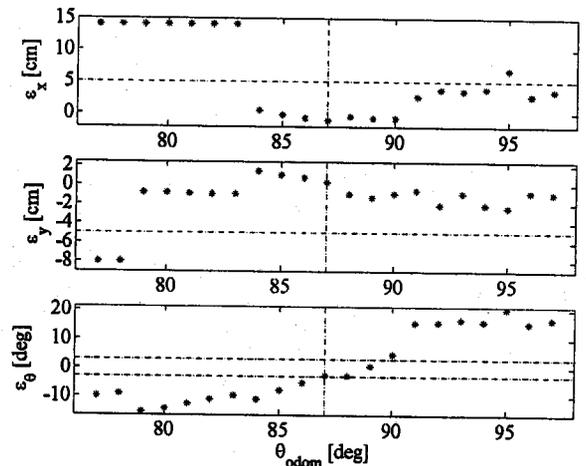
(a) $\varepsilon_x = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 87^\circ$.



(b) $\varepsilon_y = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 87^\circ$.



(c) $\varepsilon_\theta = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 87^\circ$.



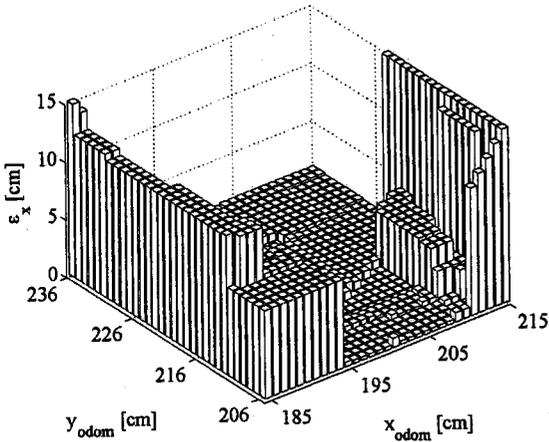
(d) $\varepsilon_{x,y,\theta} = f(\theta_{odom})$ pour $(x_{odom}, y_{odom}) = (214, 213)$.

Figure 2.34 – Position réelle 05 (200, 221, 87) : Résultats de correction en fonction de la localisation odométrique.

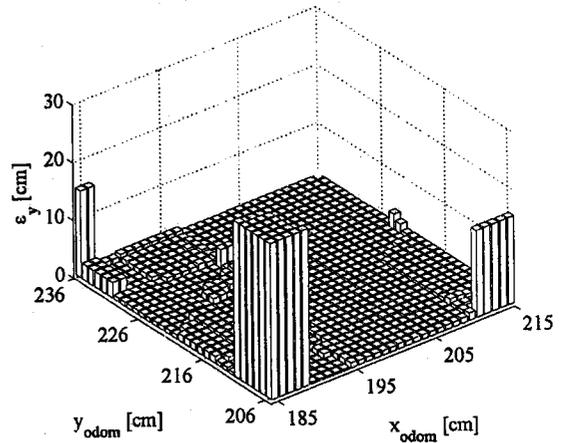
2.8.3.4 Etude de la position 09

La localisation 09 de la trajectoire d'essai effectuée par le robot est située en (168, 267, 180). Sur les histogrammes présentés à la figure 2.36(b), nous pouvons constater que l'algorithme donne une bonne estimation de l'ordonnée $Y_{corrige}$ du véhicule dans l'environnement car la répartition des valeurs corrigées est unimodale et que sa dispersion est faible. La correction de x est légèrement moins bonne, car sa répartition possède une dispersion plus importante que celle de y et elle est biaisée. La majorité des valeurs est située dans la limite des 5 cm d'erreur. La correction en orientation donne une répartition des valeurs corrigées qui ne présente qu'un mode centré à proximité de la valeur réelle, mais avec une dispersion importante signe d'une faible convergence des solutions proposées suivant les différentes localisations odométriques utilisées. Sur la figure 2.36(a), il apparaît que les mesures renvoyées par les capteurs latéraux, qui dans

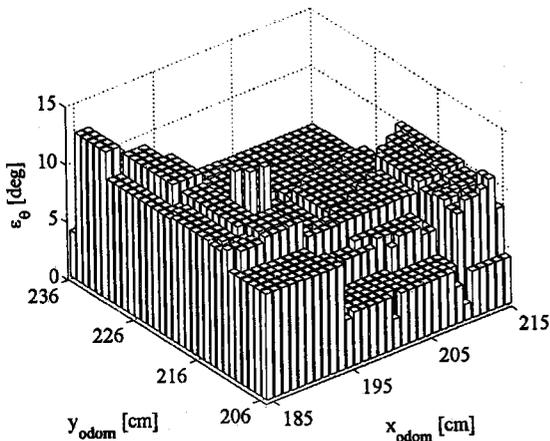
cette configuration sont utiles à la détermination de la correction en X , sont fortement perturbées ou ont un angle d'incidence trop élevé. Elles ne seront donc pas associées aux segments. L'algorithme ne disposera donc pas d'un nombre de mesures suffisant pour corriger la position en X , représentées par des cercles pleins, comparé au nombre de mesures appariées dans la direction des Y , représentées par des cercles vides.



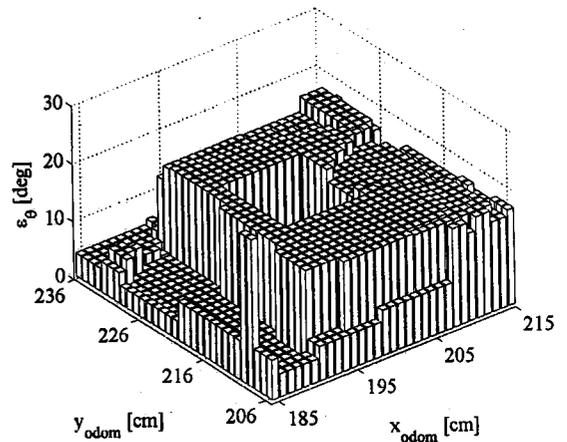
(a) $\varepsilon_x = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 83^\circ$.



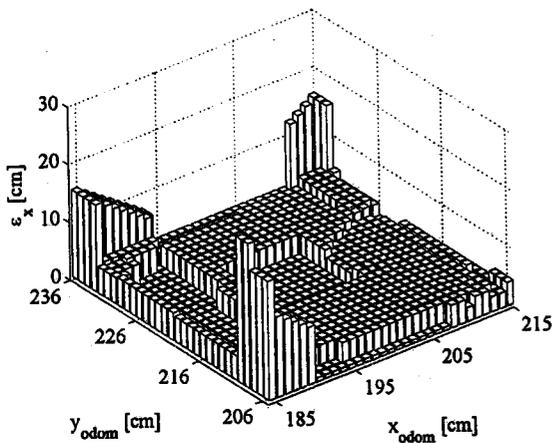
(b) $\varepsilon_y = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 83^\circ$.



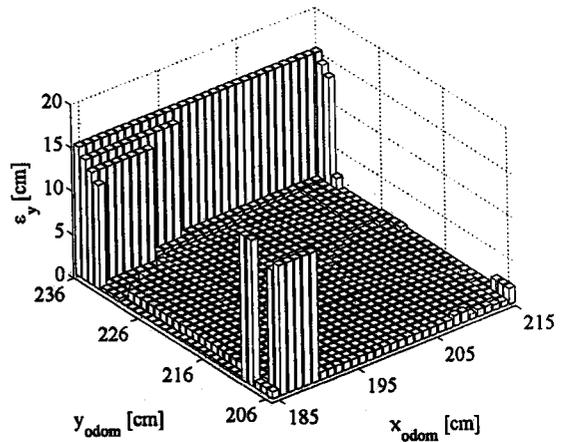
(c) $\varepsilon_\theta = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 83^\circ$.



(d) $\varepsilon_\theta = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 91^\circ$.

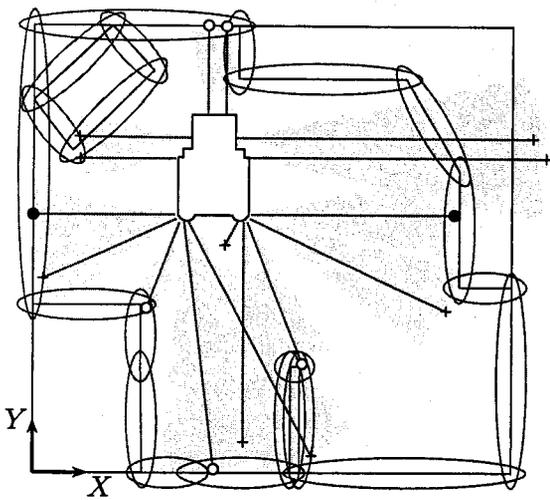


(e) $\varepsilon_x = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 91^\circ$.

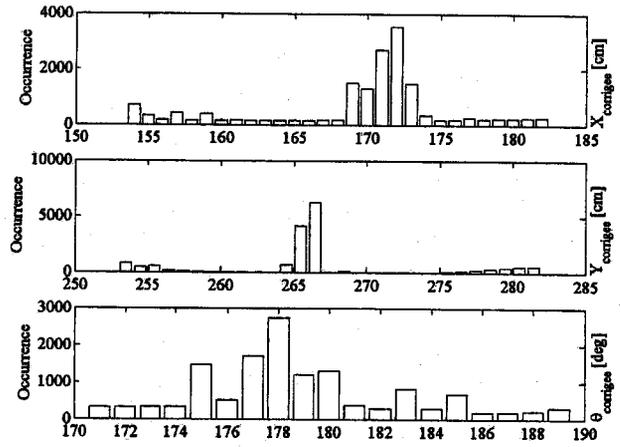


(f) $\varepsilon_y = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 91^\circ$.

Figure 2.35 – Position réelle 05 (200, 221, 87) : Résultats de correction en fonction de la localisation odométrique.

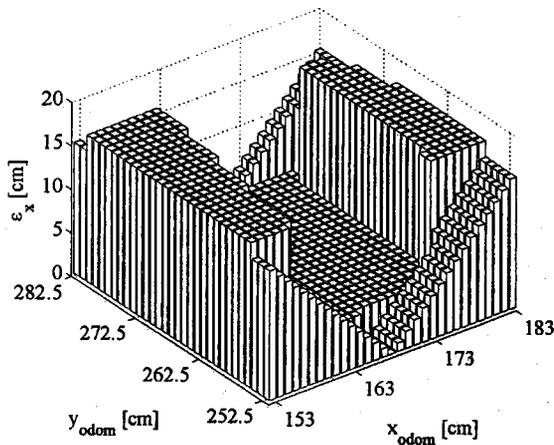


(a) Localisation Odométrique sans erreur (168, 267, 180).

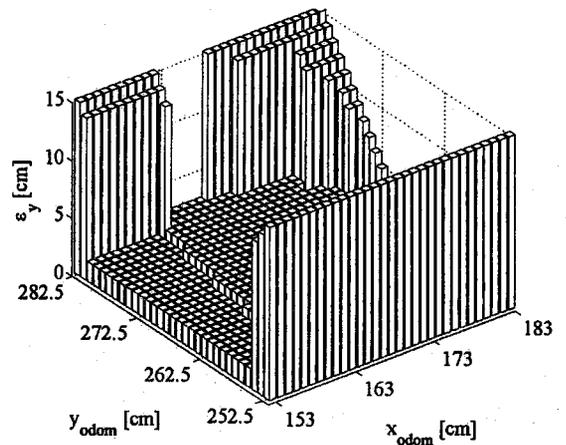


(b) Histogrammes des résultats de la correction.

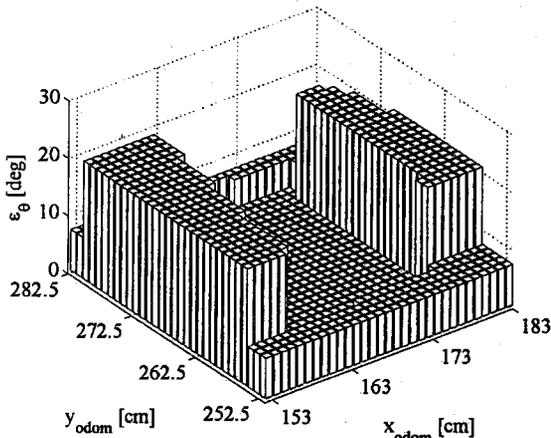
Figure 2.36 – Position réelle 09 (168, 267, 180).



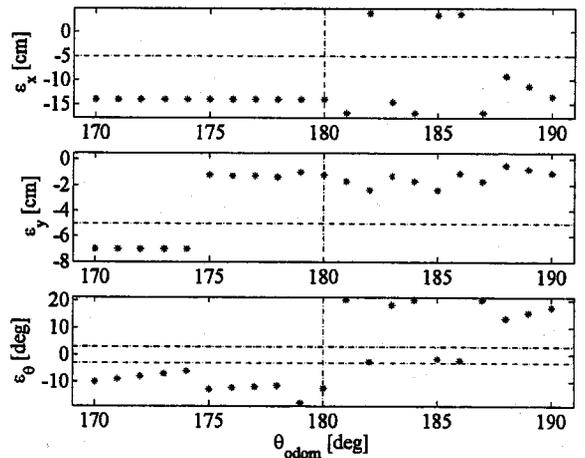
(a) $\varepsilon_x = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 187^\circ$.



(b) $\varepsilon_y = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 187^\circ$.



(c) $\varepsilon_\theta = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 187^\circ$.



(d) $\varepsilon_{x,y,\theta} = f(\theta_{odom})$ pour $(x_{odom}, y_{odom}) = (154, 260)$.

Figure 2.37 – Position réelle 09 (168, 267, 180) : Résultats de correction en fonction de la localisation odométrique.

Les figures 2.37(a) à 2.37(c) montrent que pour cette localisation réelle, et pour l'orientation odométrique de 187° , il existe des positions odométriques pour lesquelles le nombre d'appariements est insuffisant pour réaliser une correction (régions pour $253 \leq y_{odom} \leq 256 \text{ cm}$ et $285 \leq y_{odom} \leq 283 \text{ cm}$). D'autres positions odométriques conduisent à une mauvaise correction en x en raison du faible nombre d'appariements dans cette direction. Dans ces deux régions l'erreur en orientation est supérieure au seuil de 3° . La figure 2.37(d) montre que l'erreur en orientation influence l'appariement des données et ainsi la correction car pour de nombreux cas l'erreur de localisation initiale est égale à l'erreur après correction. Pour toutes les erreurs sur l'orientation odométrique négative, la correction en x n'est pas réalisable.

2.8.3.5 Etude de la position 15

La localisation 15 de la trajectoire place le robot en (302, 137, 268). Les résultats présentés à la figure 2.38 laissent apparaître une correction satisfaisante en position; les répartitions des valeurs corrigées $X_{corrige}$ et $Y_{corrige}$ présentent toutes deux un seul mode qui est respectivement centré sur les valeurs réelles X_{reel} et Y_{reel} . La correction en orientation est par contre de moins bonne qualité; la répartition de $\theta_{corrige}$ ne présente pas de mode dominant et est fortement diffuse.

Les figures 2.39(a) à 2.39(c) montrent que pour cette localisation réelle, et pour l'orientation odométrique de 278° la correction en y est quasiment toujours bien réalisée, sauf pour quelques cas extrêmes situés en périphérie de la zone d'initialisation de la localisation odométrique. En revanche les résultats obtenus en orientation (figure 2.39(c)) sont très différents de la valeur exacte. Ces constatations sont corroborées par la figure 2.39(d), qui présentent, pour une position odométrique donnée, une bonne correction en y et légèrement moins bonne en x , quelle que soit l'orientation odométrique utilisée. Par contre la correction en orientation n'est pas aussi efficace.

essai	$\varepsilon_x [cm]$		$\varepsilon_y [cm]$		$\varepsilon_\theta [^\circ]$	
	μ	σ	μ	σ	μ	σ
00	0.3	0.3	2.0	0.4	1.8	2.8
01	-	-	-	-	-	-
02	-	-	-	-	-	-
03	-	-	2.7	1.0	7.8	6.9
04	3.1	1.5	1.2	0.7	22.5	12.5
05	2.1	1.7	1.0	0.8	9.8	6.5
06	2.5	0.4	1.3	1.3	8.1	6.5
07	1.3	1.1	1.5	0.6	5.1	4.5
08	-	-	-	-	-	-
09	3.3	1.2	1.6	0.7	6.4	6.2
10	2.0	1.2	0.7	0.4	5.5	5.5
11	1.2	0.6	1.7	1.8	2.8	3.6
12	-	-	-	-	-	-
13	2.3	0.7	0.9	0.4	9.4	11.0
14	0.5	0.8	0.4	0.7	9.8	5.5
15	1.5	1.5	0.9	0.4	8.4	6.9

Tableau 2.2 – Valeur moyenne et écart-type de l'erreur de localisation pour les essais réels.

essai	$x[\%]$	$y[\%]$	$\theta[\%]$
00	97.7	97.4	97.7
01	-	-	-
02	-	-	-
03	-	97.9	32.6
04	55.8	9.7	0.4
05	82.0	96.3	18.0
06	72.8	96.0	20.0
07	98.0	97.8	28.8
08	-	-	-
09	81.8	92.6	47.1
10	81.1	98.6	50.1
11	83.7	35.1	56.5
12	-	-	-
13	54.6	94.0	42.4
14	87.5	99.5	7.7
15	81.9	99.7	31.6

Tableau 2.3 – Pourcentage de réussite de correction de chaque paramètre définissant la localisation du véhicule.

2.8.4 Résumé des résultats obtenus pour le reste des essais

Le tableau 2.3 résume les résultats obtenus pour la correction des différentes localisations odométriques simulées pour les positions réelles de la trajectoire de test. Dans ce tableau les lignes laissées vides correspondent à des positions réelles pour lesquelles le recalage n'a pas eu lieu en raison d'un nombre d'appariements insuffisant pour l'exécution de la procédure. La position 04 donne un nombre suffisant de mises

en correspondance, mais la configuration du véhicule dans son environnement place les capteurs de telle façon que l'angle d'incidence de l'onde est élevé et rend l'approximation du modèle ponctuel très grossière d'où un grand nombre de résultats erronés. En règle générale lorsque suffisamment d'appariements sont disponibles et qu'une proportion non négligeable d'entre eux sont favorables au modèle ponctuel (orthogonalité entre l'onde et la surface) l'erreur d'estimation de la position est en moyenne dans la limite des objectifs fixés (5 cm). Par contre, nous constatons que l'algorithme n'obtient pas d'aussi bons résultats pour la correction en orientation. En effet, l'erreur moyenne n'atteint pas l'objectif des 3° que nous nous étions fixés pour le maintien d'une estimation correcte de la localisation du VAHM.

L'algorithme d'ajustement de la localisation par capteurs à ultrasons donne des corrections en position qui permettent d'atteindre l'objectif fixé lorsque les conditions ne sont pas trop défavorables. Par contre, la correction de l'orientation donne de moins bons résultats en raison de la faible information angulaire que fournit le capteur à ultrasons.

2.9 Conclusion sur la méthode

Dans ce chapitre nous avons présenté une méthode de correction de la localisation odométrique à partir d'un modèle de l'environnement connu a priori et de seize mesures fournies par des capteurs à ultrasons. Après avoir présenté le projet VAHM, nous avons rappelé les problèmes liés au système odométrique et ensuite évalué l'évolution de l'erreur de localisation générée par l'odomètre installé sur le prototype. Nous avons pu ainsi fixer la fréquence à laquelle les corrections de la localisation odométrique devront être effectuées lors des déplacements du véhicule. Nous avons ensuite poursuivi par une présentation du modèle global de l'environnement dans lequel évolue le fauteuil et du modèle créé à partir des données fournies par les capteurs à ultrasons. Puis nous avons présenté la méthode de mise en correspondance des points du modèle local avec les segments du modèle global ainsi que le calcul itératif des transformations à appliquer à la localisation estimée, afin de minimiser, aux sens des moindres carrés, la somme des distances points segments.

Les temps de calcul de la méthode de correction ont été évalués de l'ordre de 300 ms ce qui permet une exécution en cours de mouvement.

Lorsque le nombre d'appariement est suffisant, la procédure corrige efficacement la localisation odométrique et l'objectif de réduire l'erreur d'estimation à une valeur inférieure à 5 cm est bien atteint. Par contre, l'approche s'avère inopérante pour la correction de l'orientation; en effet l'erreur d'estimation maximale de 3° que nous nous étions fixés est fréquemment dépassée, et peut atteindre des valeurs de l'ordre de 30°.

Nous avons remarqué lors de cette évaluation qu'il existe une classe de cas où les corrections en position et en orientation conduisent à des erreurs inférieures aux seuils fixés. Ces configurations sont celles où un maximum de capteurs ont leur axe acoustique normal à la surface qu'ils détectent. Dans ce cas l'erreur due au modèle ponctuel de la mesure produite par la détection d'un objet par le capteur à ultrasons est faible. Dans les autres cas si l'erreur en position est généralement acceptable après exécution de la procédure il n'en est pas de même pour l'erreur en orientation. En effet, la faible précision en direction du capteur à ultrasons induit un manque d'information en orientation préjudiciable à toute tentative d'ajustement de cette variable. Or comme nous l'avons fait remarquer au paragraphe 1.3.1.2 (page 39), une erreur sur l'orientation estimée perturbe également l'estimation de la position lors de l'utilisation des équations odométriques; le calcul de la position dépend d'une part de la position antérieure et de l'amplitude du déplacement, mais aussi de la direction de déplacement (orientation du véhicule). Ainsi l'utilisation d'une autre source d'information donnant une mesure d'angle plus précise serait utile pour éviter un "décrochement" de la position estimée. Nous pensons pour cela utiliser un système de vision monoculaire qui présente une toute autre perception de l'environnement. Le chapitre suivant présente cette méthode d'ajustement de l'orientation estimée à partir de l'acquisition d'une image de la scène située devant le véhicule.

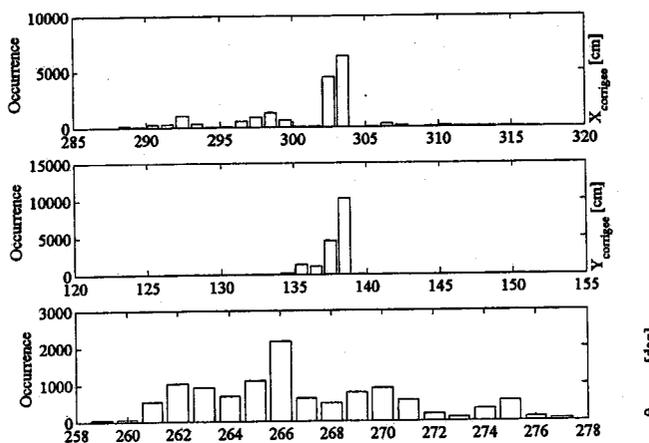
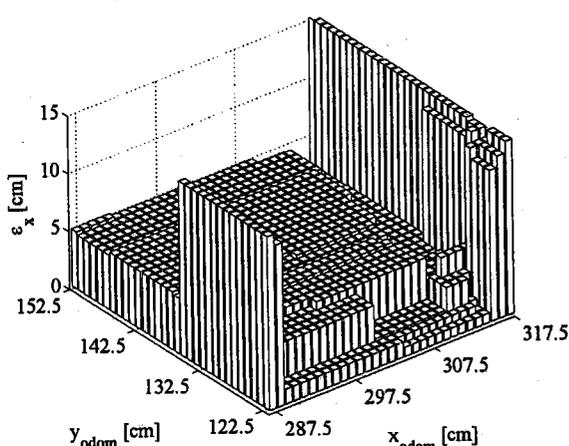
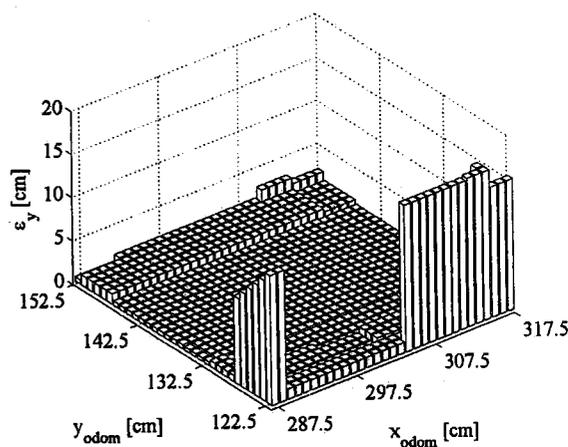


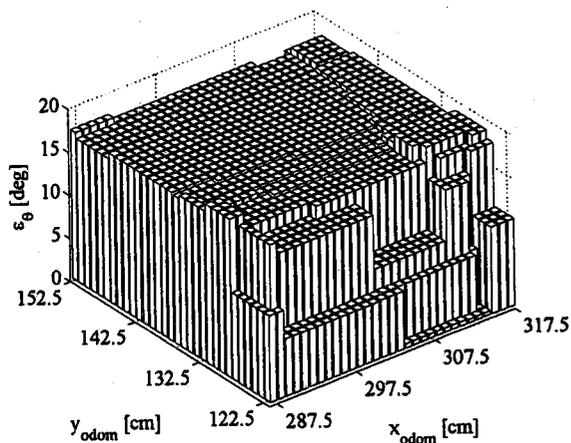
Figure 2.38 – Position réelle 15 (302, 137, 268) : Histogrammes des résultats de la correction.



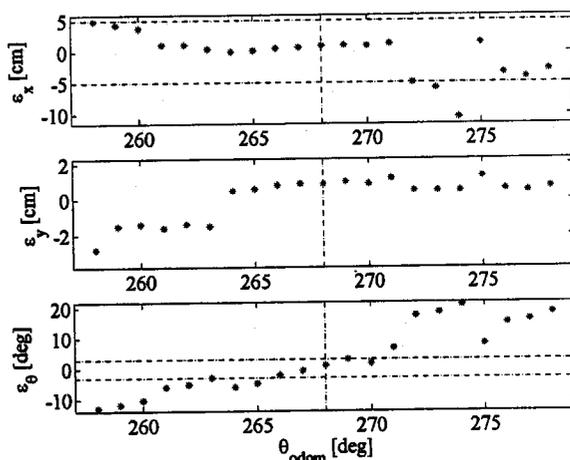
(a) $\epsilon_x = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 278^\circ$.



(b) $\epsilon_y = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 278^\circ$.



(c) $\epsilon_\theta = f(x_{odom}, y_{odom})$ pour $\theta_{odom} = 278^\circ$.



(d) $\epsilon_{x,y,\theta} = f(\theta_{odom})$ pour $(x_{odom}, y_{odom}) = (312, 127)$.

Figure 2.39 – Position réelle 15 (302, 137, 268) : Résultats de correction en fonction de la localisation odométrique.

Cette page est laissée blanche intentionnellement

Chapitre 3

Ajustement de l'orientation par vision monoculaire

3.1 Introduction

Comme nous avons pu le constater au chapitre précédent, la méthode de correction dynamique de la localisation odométrique ne fournit pas la correction en orientation escomptée. Cela est dû aux caractéristiques du capteur à ultrasons et notamment à son cône d'émission qui ne permet pas une bonne précision en orientation. D'où la nécessité d'introduire un capteur complémentaire. Nous avons opté pour un système de vision monoculaire qui permet une toute autre perception de l'environnement et pourra être employé à d'autres tâches sur le prototype comme l'évitement d'obstacles ou d'autres types de navigations réactives. L'objectif de l'ajustement en orientation par vision monoculaire est d'obtenir une meilleure estimation de l'orientation du véhicule, c'est à dire maintenir une erreur moyenne en orientation inférieure à 3°, ce que la première méthode ne réalisait pas.

Comme précédemment, la méthode doit pouvoir s'exécuter suffisamment rapidement et nécessiter peu de ressources système afin qu'elle puisse fonctionner alors que le véhicule est en cours de mouvement sans altérer le reste des fonctionnalités comme la détection de collisions, l'assistance à la commande du véhicule ou l'interface homme/machine.

3.2 Principe de la méthode

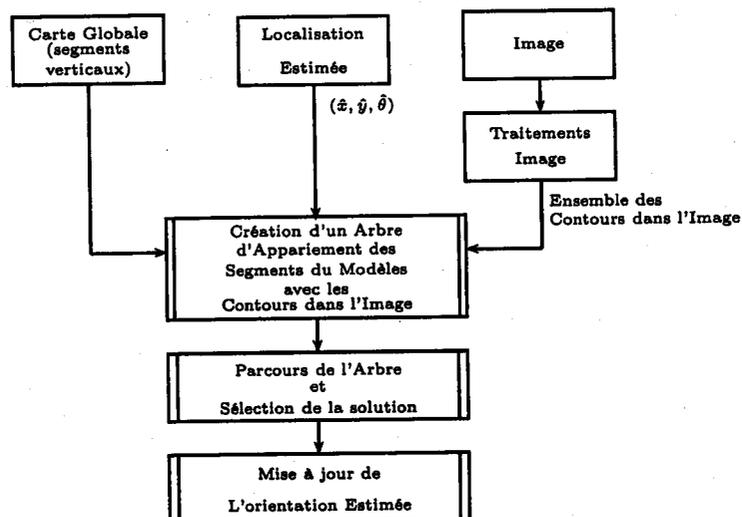


Figure 3.1 – Principe de la méthode d'ajustement de l'orientation par vision monoculaire.

Cette méthode d'ajustement de l'orientation du véhicule utilise les informations suivantes :

- la position et l'orientation estimées par la méthode de localisation dynamique présentée au chapitre 2,
- un modèle de l'environnement mémorisant la position des segments verticaux de l'environnement,
- les positions des contours verticaux dans l'image fournie par la caméra embarquée sur le véhicule.

Le principe de la méthode d'ajustement de l'orientation (figure 3.1) est le suivant : premièrement elle crée un arbre d'appariement qui permet d'obtenir toutes les combinaisons possibles pour apparier les contours de l'image avec les segments du modèle de l'environnement (une solution par chemin). Puis, dans une seconde phase, l'algorithme parcourt l'arbre afin de déterminer le meilleur ajustement en orientation.

3.3 Modèles utilisés par la méthode d'ajustement

3.3.1 Modélisation de l'environnement

Les composantes verticales détectables de l'environnement constituent des balises naturelles relativement aisées à modéliser et à détecter, elles sont par ailleurs des repères fiables pour connaître l'orientation d'un mobile dont la position est connue. Nous les avons utilisées comme caractéristiques de l'environnement à l'instar d'autres projets [Kro89, Bor02a, Cau02].

Les segments verticaux sont caractérisés dans le modèle par :

- le couple (x, y) qui définit la position de la verticale dans le référentiel de l'environnement,
- l'altitude de son point le plus bas z_b , et l'altitude de son point le plus haut z_h .

La figure 3.2 donne une représentation de ces paramètres.

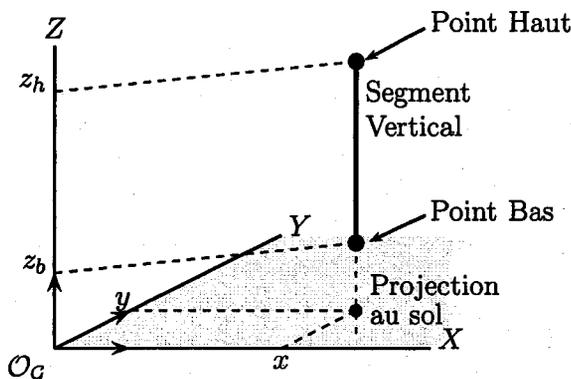


Figure 3.2 – Modèle d'un segment vertical de l'environnement.

Comme beaucoup d'objets présents dans l'environnement sont polygonaux, les composantes verticales du modèle sont situées aux intersections des segments horizontaux du modèle utilisés dans la méthode de localisation dynamique. Mais d'autres verticales peuvent être ajoutées comme celles limitant les côtés d'un poster ou d'une fenêtre qui n'avaient pas été intégrés dans le modèle précédent puisque indétectables par les capteurs à ultrasons.

3.3.2 Informations extraites de l'image

Les informations que nous cherchons à extraire de l'image sont les projections dans le plan image de la caméra des bords verticaux des objets de l'environnement.

Lors de la mise en œuvre de la méthode d'ajustement de l'orientation, la caméra utilisée est montée sur le véhicule de manière à placer l'axe des ordonnées de son plan image parallèle aux bords verticaux des objets de l'environnement (segments verticaux). Ainsi les projections des bords verticaux de l'environnement dans le plan image sont des contours verticaux.

A l'aide de différents outils du domaine du traitement d'image (voir 3.4) nous cherchons à extraire un ensemble de contours verticaux. Chaque contour est caractérisé par son abscisse dans l'image. Cette abscisse est donnée par rapport au centre de l'image, ce qui nous permet de savoir si le bord vertical se situe à droite ou à gauche de l'axe optique de la caméra (figure 3.3).

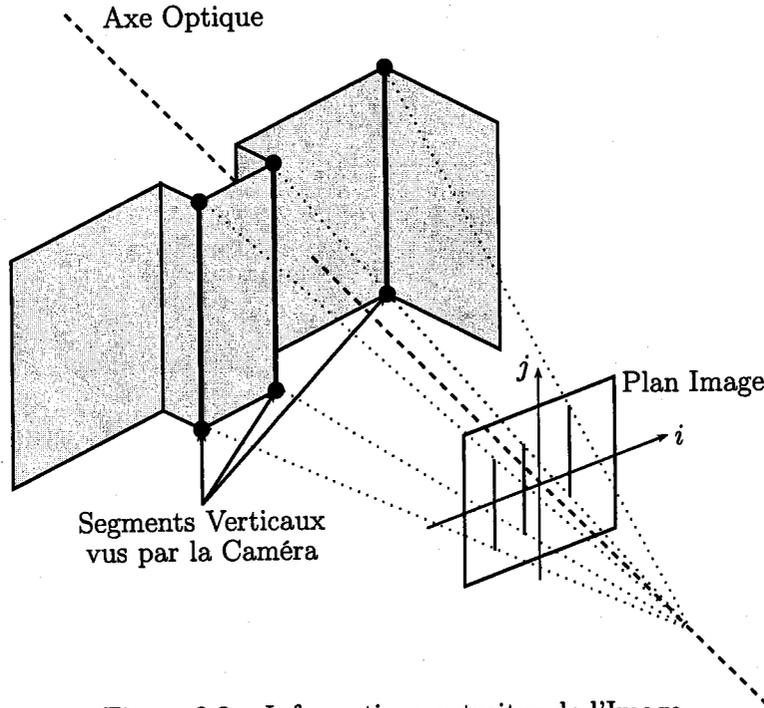
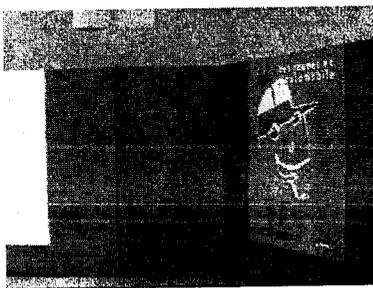
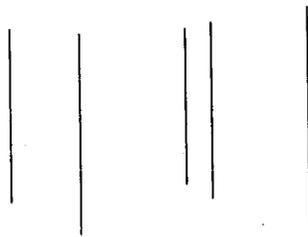


Figure 3.3 – Informations extraites de l'Image.

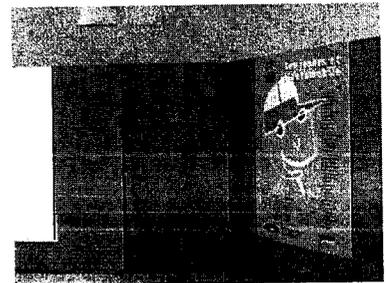
Les figures 3.4(a) et 3.4(b) représentent respectivement une image source et le résultat obtenu après traitement de l'image. La figure 3.4(c) représente l'image source sur laquelle nous avons affiché les contours détectés par les traitements d'image. Cet exemple montre d'une part l'efficacité des outils utilisés que nous avons adaptés à nos besoins (voir figure 3.4) et d'autre part que les projections des bords verticaux des objets de l'environnement dans le plan image sont bien des contours verticaux.



(a) Image avant traitement.



(b) Contours détectés.



(c) Image et contours détectés superposés.

Figure 3.4 – Exemple de détections de contours dans une image

3.4 Mise en œuvre de l'extraction des contours de l'image

3.4.1 Origine des contours

Les contours d'une image sont situés où le niveau d'intensité change brusquement. Une bonne détection des contours rend la détermination des frontières et l'identification d'objet plus facile. Les variations d'intensité peuvent résulter de plusieurs causes physiques (figure 3.5). Dans la scène représentée, les contours présents dans l'image peuvent être dus aux frontières d'occlusions (un des plans composant la frontière n'est pas visible d'un point de vue), aux frontières de plis (tous les plans composant la frontière sont visibles), aux frontières externes (similaires aux limites d'occlusions mais ne résultant pas de l'intersection de plans) ainsi qu'à des marques, des ombres ou des spéclarités.

Il est difficile d'extraire les projections des frontières significatives directement à partir d'une image. En réalisant tout d'abord une transformation de l'image accentuant les discontinuités locales dans l'intensité, puis en regroupant ces points dans des contours, la détection des frontières est plus aisée.

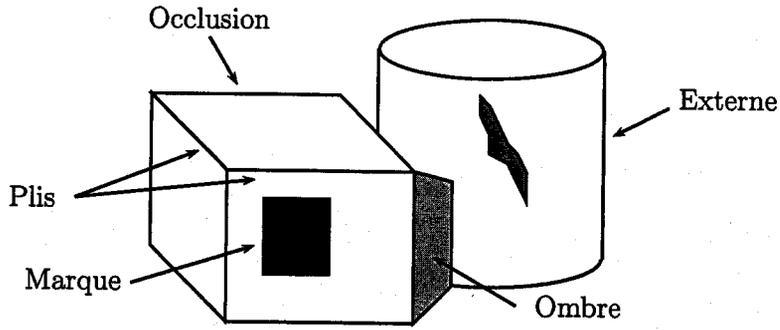


Figure 3.5 – Les origines des contours dans une image.

Dans cette section, nous considérerons uniquement la détection de contours verticaux. Nous présentons successivement les divers outils appliqués à l'image fournie par la caméra afin d'en extraire les informations utiles pour l'ajustement de l'orientation du véhicule.

3.4.2 Amélioration du contraste de l'image

Afin que la détection des variations de l'intensité lumineuse dans l'image soit plus facile, nous effectuons une égalisation d'histogramme qui a pour but d'utiliser toute la plage de variation de la palette des niveaux de gris de l'intensité lumineuse.

L'histogramme H_I d'une image $I(L \times H)$ (L : largeur en pixels, H : hauteur en pixels) donne le nombre d'apparitions de chaque niveau de gris. C'est une représentation de la fréquence d'apparition d'un niveau de gris dans l'image. L'objectif de l'égalisation d'histogramme est de rendre la fréquence d'apparition de chaque niveau de gris constante. L'histogramme devient donc "plat". L'égalisation de l'histogramme est effectuée comme suit : nous calculons l'histogramme cumulé C_I de l'image qui associe à tout niveau de gris n le nombre d'apparitions de niveaux de gris ayant une valeur inférieure ou égale à n , nous avons donc :

$$C_I(n) = H_I(0) + \dots + H_I(n) \quad (3.1)$$

Nous pouvons donc obtenir C_I par l'itération suivante :

$$\begin{aligned} C_I(0) &= H_I(0), \text{ et} \\ C_I(n) &= H_I(n) + C_I(n-1) \text{ pour } n = 1, \dots, M \end{aligned} \quad (3.2)$$

où M est le nombre de niveaux de gris ($M = 255$ par exemple). En particulier $C_I(M) = L \times H$. Nous appliquons alors à l'image I la fonction d'ajustement des niveaux de gris f définie par :

$$f(n) = \frac{M}{L \times H} C_I(n) \text{ arrondi à l'entier le plus proche} \quad (3.3)$$

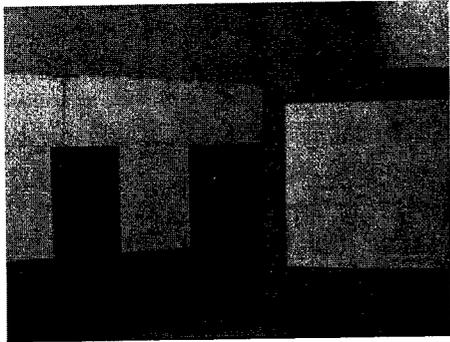
Notons que $\frac{C_I(n)}{L \times H}$ est la proportion de points de l'image ayant un niveau de gris inférieur ou égal à n dans l'image I , donc le niveau de gris $f(n)$ sera la fraction de M correspondant à cette proportion. Nous avons :

$$\begin{aligned} f(0) &= \frac{M}{L \times H} H_I(0) \text{ et} \\ f(n) - f(n-1) &= \frac{M}{L \times H} H_I(n) \text{ pour } n = 1, \dots, M \end{aligned} \quad (3.4)$$

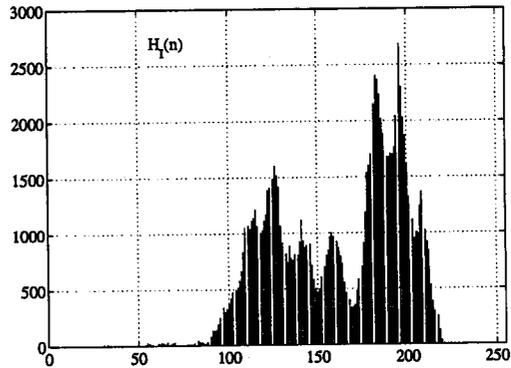
et en particulier $f(M) = M$.

L'effet de cette fonction f est de "mettre à gauche" de chaque niveau de gris un écart dont la taille est proportionnelle au nombre de points ayant ce niveau de gris dans l'image I . Donc tout niveau de gris fortement représenté dans l'image non égalisée se verra écarté des niveaux de gris plus sombres. En pratique l'image aura un aspect où les zones claires, moyennes et sombres s'équilibreront.

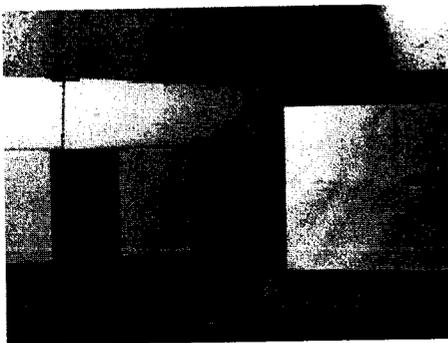
La figure 3.6(c) donne le résultat du traitement de l'image initiale (figure 3.6(a)). Leurs histogrammes respectifs (figures 3.6(b) et 3.6(d)) font apparaître pour l'image initiale une concentration autour des niveaux de gris moyens, c'est à dire une image qui manque de contraste et une utilisation partielle de la plage des niveaux de gris disponible. L'histogramme cumulé de l'image égalisée (figure 3.6(e)) est une droite dont la pente est constante et égale au nombre d'apparitions du premier niveau de gris¹. L'histogramme cumulé étant l'intégration de l'histogramme des niveaux de gris, si le premier est une droite alors le second correspond à une constante. L'objectif d'une fréquence d'apparition constante pour chaque niveau de gris est donc satisfait. Nous remarquons que l'image égalisée (figure 3.6(c)) est plus contrastée que l'image d'origine (figure 3.6(a)).



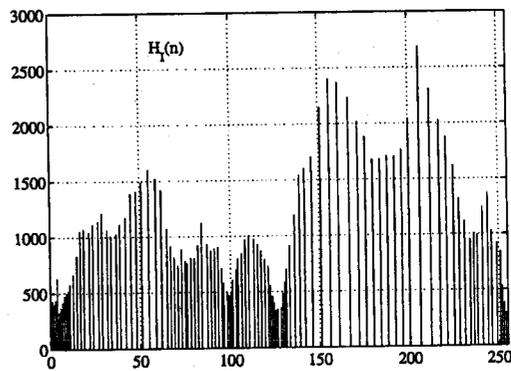
(a) Image source.



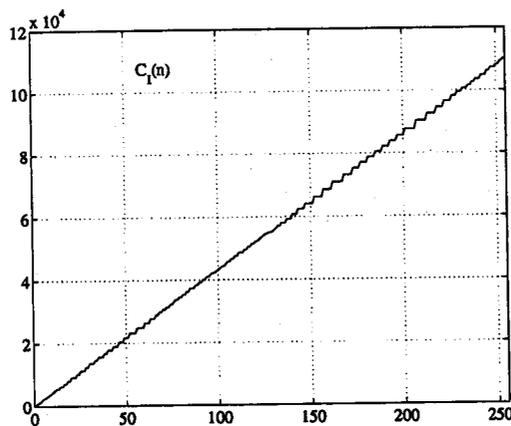
(b) Histogramme de l'image source.



(c) Image égalisée.



(d) Histogramme de l'image égalisée.



(e) Histogramme cumulé de l'image égalisée.

Figure 3.6 – Egalisation d'histogramme.

¹qui est également $(L \times H)/N$

3.4.3 Détection des contours dans l'image

La façon la plus courante pour détecter les contours est l'utilisation du gradient de l'image. Pour une image $I(i, j)$, la norme du gradient (également appelé amplitude) $g(i, j)$ et sa direction $\theta(i, j)$, sont calculées à l'aide des relations suivantes :

$$g(i, j) \approx (\Delta i^2 + \Delta j^2)^{\frac{1}{2}} \quad (3.5)$$

et

$$\theta(i, j) \approx \arctan \left(\frac{\Delta j}{\Delta i} \right) \quad (3.6)$$

où Δi (respectivement Δj) est la composante en i (respectivement en j) du gradient $g(i, j)$ de l'image.

Afin de déterminer les composantes Δi et Δj du gradient, plusieurs opérateurs sont utilisables [Hor95, Par96]. Nous avons choisi d'utiliser l'opérateur de Deriche [Der87], car celui-ci est actuellement le plus performant en terme de rapidité d'exécution et de qualité de détection des contours. Il est conçu pour minimiser l'erreur de position et maximiser le rapport signal sur bruit

Cet opérateur est une combinaison de deux filtres itératifs (figure 3.7) qui permettent de calculer les images Δi et Δj . Nous pouvons alors déterminer la norme et la direction du gradient de l'image.

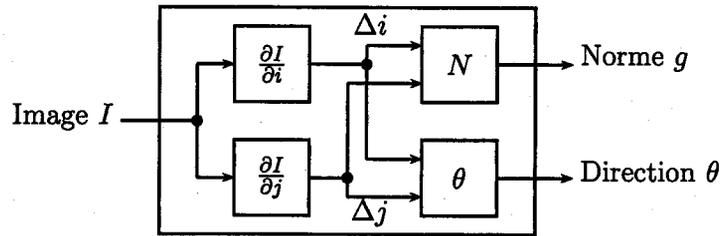


Figure 3.7 – Schéma de principe du calcul de la norme et de la direction du gradient de l'image.

Ces deux filtres sont :

– un filtre de lissage l , donné par :

$$l(x) = b(\alpha|x| + 1)e^{-\alpha|x|} \quad (3.7)$$

où x représente soit la variable i , soit la variable j .

– Un filtre de dérivation d , donné par :

$$d(x) = kxe^{-\alpha|x|} \quad (3.8)$$

Lors des essais nous avons utilisé le filtre avec $\alpha = 1.0$.

La décomposition de l'opérateur en deux filtres permet une mise en œuvre efficace du filtre en deux dimensions.

Nous posons :

- D_i et D_j les filtres de dérivation selon les axes i et j du plan image,
- L_i et L_j les filtres de lissage selon les axes i et j du plan image,
- $I(i, j)$ l'image d'entrée de l'opérateur de Deriche.

L'image résultante, ou image des contours, $O(i, j)$, est donnée par la norme et la direction du gradient :

$$g(i, j) = \sqrt{((I * L_j) * D_i)^2 + ((I * L_i) * D_j)^2} \quad (3.9)$$

$$\theta(i, j) = \arctan \frac{(I * L_i) * D_j}{(I * L_j) * D_i} = \arctan \frac{\Delta j}{\Delta i} \quad (3.10)$$

où Δi et Δj sont les composantes du gradient suivant les axes i et j .

3.4.3.1 Calcul du gradient

Le filtrage récursif permet de mettre en œuvre les filtres à réponse impulsionnelle infinie avec un coût algorithmique faible. Cette technique est employée pour programmer les filtres de lissage et dérivation de l'opérateur de Deriche. La détermination des coefficients des filtres est réalisée en utilisant un signal connu en entrée du filtre et en se fixant le type de signal à obtenir en sortie. Comme l'objectif est de détecter des contours, c'est-à-dire une variation brutale de l'intensité lumineuse, nous allons "injecter" une variation brutale de l'intensité lumineuse en entrée du filtre et rechercher le moyen d'obtenir une impulsion à la position du contour. Idéalement un contour peut être représenté par une fonction échelon d'amplitude A (figure 3.8):

$$I(x) = AH(x), \text{ où } H(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases} \quad (3.11)$$

D'autres profils sont cependant utilisables. Nous avons choisi un profil en forme de droite (figure 3.9) car dans le cas d'utilisation d'images réelles les contours sont rarement des contours idéaux. Dans la littérature les coefficients donnés pour les filtres de lissage et de dérivation sont déterminés à partir de la réponse à un contour idéal. Nous donnons en annexe D le détail de la détermination des paramètres des filtres, que nous résumons au paragraphe suivant.

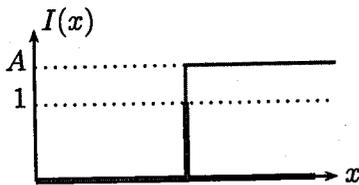


Figure 3.8 - Profil d'un contour idéal.

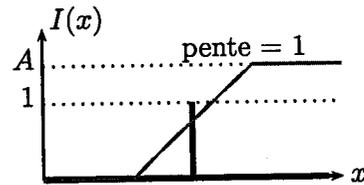


Figure 3.9 - Profil utilisé pour le contour.

(a) **Résumé des équations de filtrage** En utilisant comme variable d'entrée $I(x)$ et comme variable de sortie $O(x)$, nous pouvons écrire pour le filtre de dérivation $d(x)$:

$$\begin{aligned} O^+(x) &= AI(x-1) - B_1O^+(x-1) - B_2O^+(i-2), & x = 2, \dots, N-2 & \quad (ER1) \\ O^-(x) &= -AI(x+1) - B_1O^-(x+1) - B_2O^-(i+2), & x = N-2, \dots, 2 & \quad (ER2) \\ O(x) &= O^+(x) + O^-(x), & x = 2, \dots, N-2 & \quad (ER3) \end{aligned}$$

avec :

$$\begin{aligned} A &= -k\alpha^2 e^{-\alpha} \\ k &= \frac{-(1 - e^{-\alpha})^3}{2\alpha e^{-\alpha}(1 + e^{-\alpha})} \\ B_1 &= -2e^{-\alpha} \\ B_2 &= e^{-2\alpha} \end{aligned}$$

Le filtre de lissage $l(x)$ s'écrit quant à lui :

$$\begin{aligned} O^+(x) &= A_0I(x) + A_1I(x-1) - B_1O^+(x-1) - B_2O^+(i-2), & x = 2, \dots, N-2 & \quad (ER4) \\ O^-(x) &= A_2I(x+1) + A_3I(x+2) - B_1O^-(x+1) - B_2O^-(i+2), & x = N-2, \dots, 2 & \quad (ER5) \\ O(x) &= O^+(x) + O^-(x), & x = 2, \dots, N-2 & \quad (ER6) \end{aligned}$$

avec :

$$\begin{aligned}
 A_0 &= b \\
 A_1 &= b(\alpha - 1)e^{-\alpha} \\
 A_2 &= b(\alpha + 1)e^{-\alpha} \\
 A_3 &= -be^{-2\alpha} \\
 b &= \frac{-(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}} \\
 B_1 &= -2e^{-\alpha} \\
 B_2 &= e^{-2\alpha}
 \end{aligned}$$

(b) **Principe de calcul du gradient** La figure 3.10 résume l'utilisation des équations récurrentes (ER1) à (ER6) qui permettent de déterminer les composantes en i et en j du gradient de l'image. Le bloc l_j effectue un lissage de l'image I à l'aide des relations (ER4) à (ER6) donnant lieu à l'image I_j qui est dérivée lors de son passage dans le bloc d_i . Nous obtenons alors Δi la composante en i du gradient de l'image. La détermination de la composante en j du gradient de l'image s'effectue de manière semblable ; le bloc l_i lisse l'image I donnant lieu à l'image I_i qui est ensuite dérivée par le bloc d_j , utilisant les équations (ER4) à (ER6), et donne ainsi Δj la composante en j du gradient de l'image. Δi et Δj sont ensuite utilisés pour déterminer la norme $g(i, j)$ et la direction $\theta(i, j)$ du gradient en chaque point de l'image.

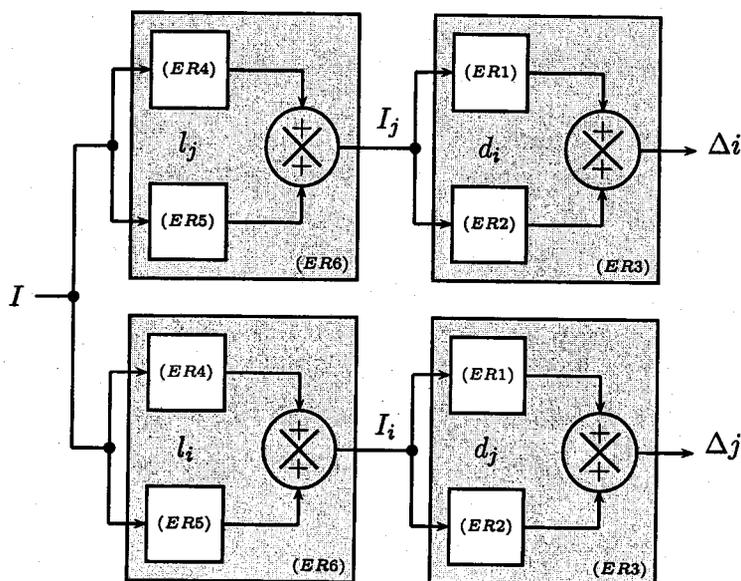


Figure 3.10 – Schéma de principe de mise en œuvre du calcul des composantes en i et en j du gradient.

3.4.3.2 Suppression des non maxima locaux

Ayant calculé le gradient en chaque point de l'image, il faut sélectionner ses maxima. C'est-à-dire déterminer pour un pixel s'il fait partie d'un contour en utilisant les valeurs de la norme et de la direction de son gradient. La méthode utilisée est la suivante :

- A l'aide de la direction du gradient nous sélectionnons deux pixels dans le voisinage des huit pixels situés autour du pixel courant. Dans l'exemple donné sur la figure 3.11 la direction du gradient est indiquée par la flèche, et les pixels 'g' et 'h' sont sélectionnés.
- En utilisant la direction et la norme du gradient nous calculons la norme du gradient pour le pixel fictif situé entre les deux pixels sélectionnés à l'étape précédente. Sur la figure 3.11 ce pixel est représenté en gris clair.
- Si la norme du gradient au point fictif est supérieure à la norme des deux autres pixels ('g' et 'h' dans notre exemple) alors le pixel courant est un pixel contour.

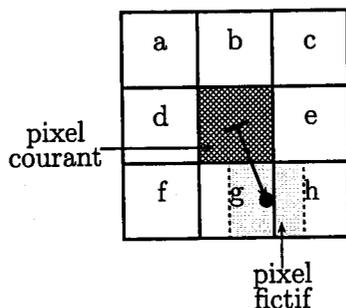


Figure 3.11 – Interpolation du gradient en un point du voisinage du point central.

Nous profitons également de cette étape pour éliminer directement les pixels contours dont la direction du gradient n'est pas proche de $\pm 90^\circ$, étant donné que seuls les contours verticaux nous intéressent.

3.4.3.3 Seuillage par hystérésis

Afin de supprimer les contours parasites correspondant à des perturbations dans l'image, nous effectuons un seuillage par hystérésis sur l'image résultant de la suppression des non maxima locaux. Le principe du seuillage par hystérésis utilise deux seuils, un seuil haut et un seuil bas (figure 3.12). Ils sont fixés en fonction de l'amplitude du gradient que l'on veut détecter, c'est-à-dire l'amplitude du changement de niveau de gris dans l'image. Lorsqu'un pixel a la norme de son gradient supérieure au seuil haut, alors il est directement considéré comme un point d'un contour, si elle est inférieure au seuil bas alors il est éliminé. Dans le cas où elle se situe entre le seuil bas et le seuil haut et que le pixel est connecté à un pixel d'un contour alors il est gardé comme élément d'un contour, sinon il est également rejeté.

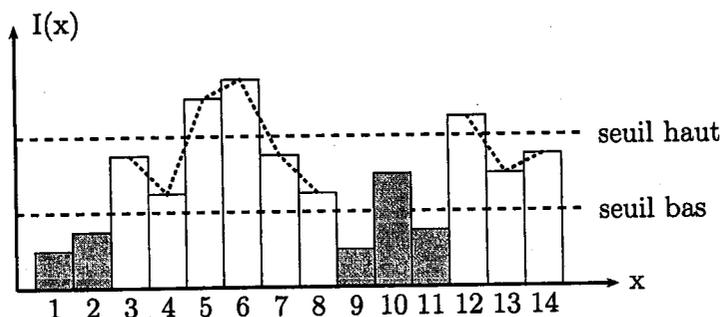


Figure 3.12 – Principe du seuillage par hystérésis.

Dans l'exemple de la figure 3.12, les échantillons marqués en gris sont éliminés car la norme de leur gradient est inférieure au seuil bas (échantillons 1, 2, 9 et 11) ou, comme pour l'échantillon 10, est comprise entre les deux seuils le pixel n'étant pas connecté à un échantillon d'un contour. Les échantillons restants, marqués en blanc, font partie d'un contour car la norme de leur gradient est supérieure au seuil haut (échantillons 5, 6, et 12) ou comprise entre les deux seuils ces échantillons étant connexes à un pixel faisant partie d'un contour (échantillons 3, 4, 7, 8, 13 et 14).

Le résultat de ce traitement est une image des pixels appartenant aux contours de l'image. Nous savons quels pixels de l'image font partie des contours verticaux. Il reste maintenant à déterminer quels ensembles de pixels appartiennent au même contour (projection d'une verticale de l'environnement) afin de définir de façon unique la position dans l'image de chaque contour. Ceci est réalisé lors de l'étape de suivi de contour.

3.4.3.4 Suivi de contour

Le suivi de contour opère en trois phases : dans un premier temps, il regroupe les pixels ayant une norme et une direction du gradient semblables dans un voisinage de 1 pixel, c'est le *chaînage de pixel*. Puis, une fois les petites chaînes construites, il effectue un regroupement des chaînes dans un voisinage

supérieur à un pixel, c'est le *chaînage des chaînes*. Une fois les chaînes regroupées, les chaînes de trop petite taille sont supprimées.

3.4.3.5 Mise en œuvre

Ces différents outils ont été mise en œuvre en langage "C". L'annexe E donne un exemple des traitements réalisés sur une des images de la séquence d'essai de la méthode d'ajustement. Cette annexe présente la transformation de l'image au fur et à mesure de l'application des différents outils jusqu'à l'obtention de la position de chaque contour dans l'image.

3.4.4 Paramètres du capteur de vision

Notre méthode d'ajustement de l'orientation par vision monoculaire nécessite la connaissance de certains paramètres de la caméra. En supposant que les verticales observées sont suffisamment éloignées de la caméra, et pour des raisons de simplicité, nous avons opté pour un modèle sténopé. Nous allons voir dans cette partie les paramètres le définissant et la manière de les déterminer expérimentalement, c'est-à-dire de calibrer la caméra.

3.4.4.1 Présentation des paramètres utilisés

La figure 3.13 représente les différents paramètres caractérisant notre capteur de vision, ainsi que les repères liés au robot et à la caméra.

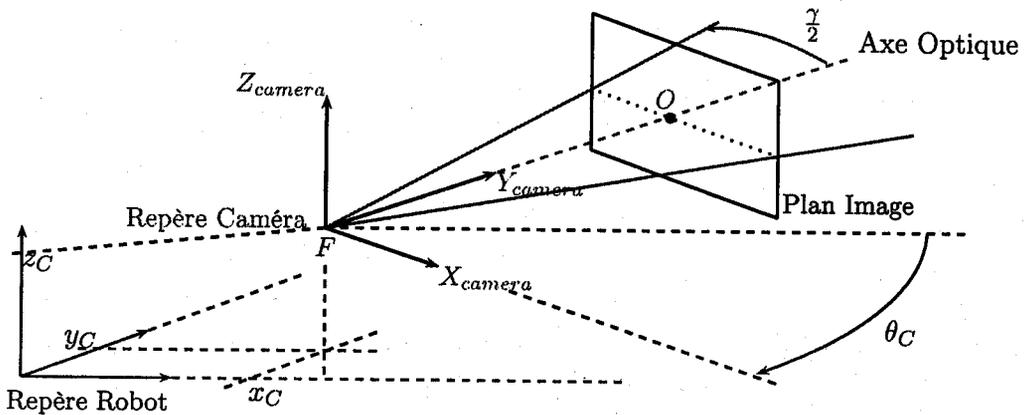


Figure 3.13 – Repères et paramètres utilisés pour définir la caméra

Les coordonnées (x_C, y_C, z_C) représentent la position et θ_C l'orientation de la caméra dans le repère lié au robot. La distance \overline{OF} est la distance focale de la caméra et l'angle $\gamma/2$ est le demi-angle d'ouverture de la caméra par rapport à l'axe optique de celle-ci. Le repère lié à la caméra est positionné en F (point focal) et l'axe Y_{camera} est placé le long de l'axe optique, la partie positive orientée vers le plan image, dont le centre est identifié par le point O .

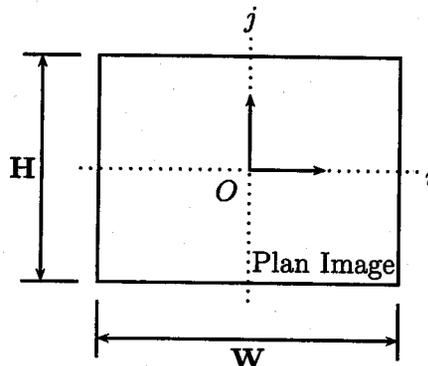


Figure 3.14 – Repère du plan image et paramètres utilisés pour définir l'image

L'image (figure 3.14) est caractérisée par sa largeur W ainsi que sa hauteur H exprimées en pixels. Le repère image est placé au centre du plan, l'axe des abscisses i positives est orienté vers la droite, et l'axe des ordonnées j positives vers le haut.

Les paramètres intrinsèques de la caméra, \overline{OF} et $\gamma/2$, sont déterminés expérimentalement à partir de l'observation d'une mire. Connaissant la position relative de la caméra par rapport à la mire, nous déterminons les paramètres \overline{OF} et $\gamma/2$ de la caméra en effectuant l'observation de la mire (figure 3.15(b)) dont les caractéristiques sont données sur la figure 3.15(a). La figure 3.16 montre une représentation vue de dessus du système de calibration. La caméra est placée devant la mire de façon à ce que son axe optique soit normal à la cible. Nous relevons alors la distance D qui sépare la mire de la caméra.

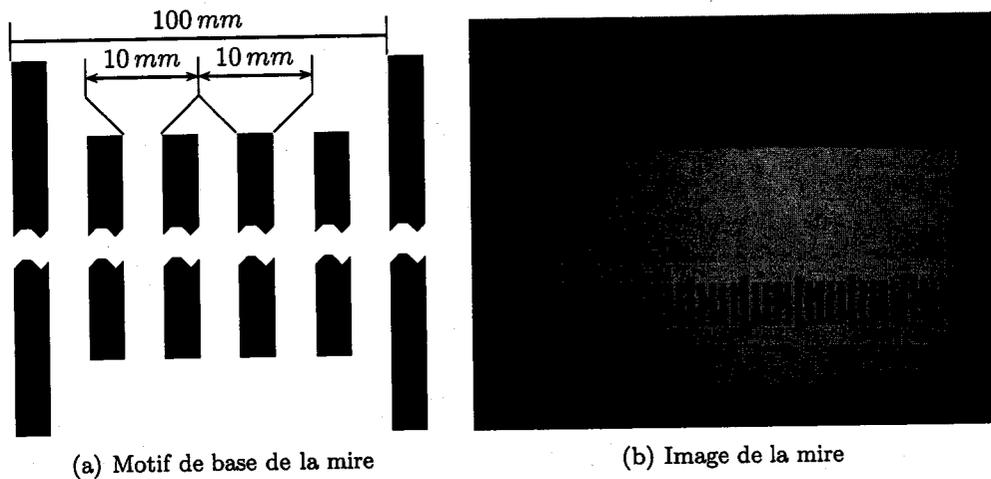


Figure 3.15 – Observation de la mire de calibration.

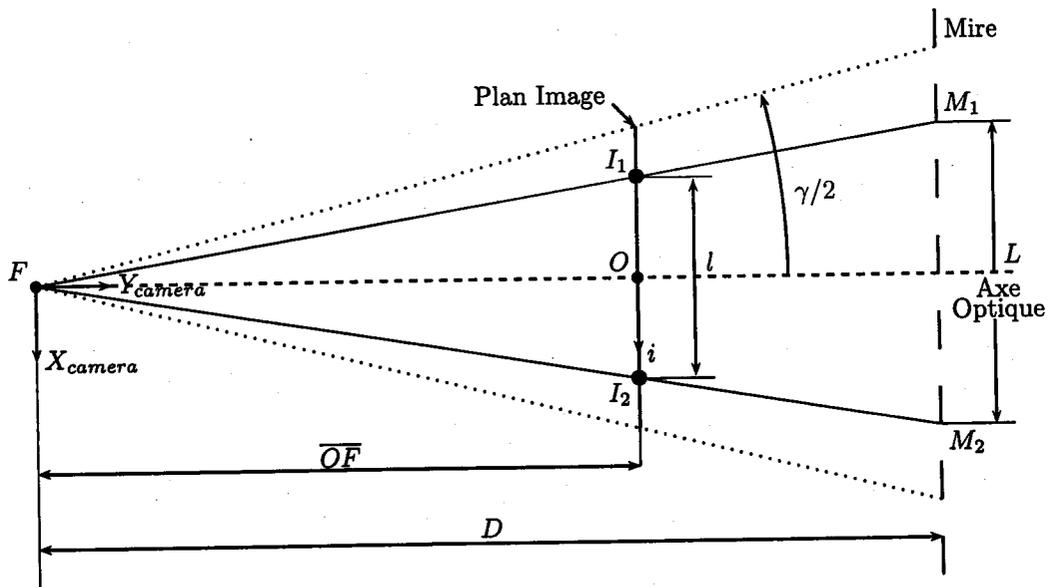


Figure 3.16 – Vue de dessus du système de calibration.

Plus cette distance sera grande, plus la calibration de la caméra sera insensible aux éventuelles erreurs de mesures. Nous identifions ensuite à partir de l'image les deux verticales de la mire les plus éloignées (dans l'exemple de la figure 3.15(b) toute la cible est visible, on prend dans ce cas les verticales extérieures à la cible). Nous relevons la distance L qui correspond à la distance entre ces deux verticales de la mire, soient M_1 et M_2 ces deux verticales (figure 3.16). Celles-ci se projettent respectivement dans le plan image en I_1 et I_2 . Nous pouvons ainsi mesurer la distance l exprimée en *pixels* qui sépare les deux contours I_1 et I_2 dans l'image. Puis, en appliquant le théorème de Thalès, nous retrouvons la longueur \overline{OF} . Elle sera

exprimée en *pixels*.

$$\frac{\overline{OF}}{D} = \frac{l}{L}$$

$$\overline{OF} = l \cdot \frac{L}{D} \quad (3.12)$$

En utilisant la distance focale \overline{OF} précédemment estimée, et la demi-largeur de l'image $W/2$, nous déterminons le demi-angle d'ouverture :

$$\frac{\gamma}{2} = \arctan \left(\frac{W}{2 \cdot \overline{OF}} \right) \quad (3.13)$$

Nous avons ainsi déterminé les deux paramètres intrinsèques de la caméra utiles à la méthode d'ajustement de l'orientation. Nous avons déterminé la focale de la caméra uniquement dans la direction des x du repère caméra car nous utilisons uniquement l'abscisse des contours verticaux dans l'image. La détermination des paramètres extrinsèques qui concernent la position et l'orientation de la caméra dans le repère lié au fauteuil est plus facile car nous choisissons la position et l'orientation. Nous avons choisi de fixer l'orientation de la caméra à 0° , c'est-à-dire que son axe optique est parallèle à l'axe de déplacement vers l'avant du robot. La caméra est placée sous l'accoudoir opposé à celui où est situé le manche de contrôle du véhicule afin de ne pas gêner l'utilisateur du véhicule. La détermination des paramètres extrinsèques se limite à un relevé de la position de la caméra sur le robot et une vérification de son orientation dans le repère lié au robot.

3.5 Détermination de l'orientation du robot à partir de l'observation d'un unique segment

Dans ce paragraphe, nous présentons le calcul de l'orientation du robot en se basant sur l'observation d'un segment vertical de l'environnement. Les paramètres connus sont la position du segment dans le repère lié à l'environnement, la position du robot dans ce même repère, ainsi que les paramètres de la caméra. Dans une première partie, nous poserons les bases pour le calcul de l'orientation du robot, puis dans une seconde partie, nous appliquerons le calcul de l'orientation au calcul de l'ajustement d'une orientation estimée du robot.

Les paramètres extrinsèques de la caméra étant connus, calculer l'orientation de la caméra est équivalent à calculer l'orientation du robot. Pour des raisons de simplicité, nous n'évoquerons dans cette partie que le calcul de l'orientation de la caméra.

3.5.1 Calcul de l'orientation

La figure 3.17 donne une représentation vue du dessus de la scène. La caméra est située au point de coordonnées (x_C, y_C) et observe une verticale située au point de coordonnées (x_V, y_V) . Ces coordonnées sont données dans le même repère. La position du contour P dans le plan image, correspondant à la verticale V , est supposée connue. Nous allons présenter dans cette section le calcul de l'orientation θ_C de la caméra à partir de l'observation P de la verticale V du modèle de l'environnement.

Nous déterminons les angles suivants :

- l'angle θ_V entre l'axe des abscisses du repère lié à l'environnement et la droite \overline{FV} . Il est donné par la relation :

$$\theta_V = \arctan \left(\frac{y_V - y_C}{x_V - x_C} \right) \quad (3.14)$$

- l'angle θ_P calculé à partir de la position du contour P dans le plan image. Cet angle est repéré par rapport à l'axe des abscisses de la caméra (X_{camera}), et est déterminé par :

$$\theta_P = 90^\circ - \arctan \left(\frac{i_P}{\overline{OF}} \right) \quad (3.15)$$

Les contours dans l'image des verticales situées à gauche (respectivement à droite) de l'axe optique auront un angle θ_P supérieur (respectivement inférieur) à 90° .

Sachant que les points V , F , et P sont alignés et que l'orientation de la caméra θ_C est définie par l'angle formé entre l'axe des abscisses du repère global et celui du repère de la caméra (figure 3.17), nous pouvons écrire :

$$\theta_V = \theta_P + \theta_C \quad (3.16)$$

Nous pouvons ainsi établir une estimation de l'orientation de la caméra $\hat{\theta}_C$ dans le repère lié à l'environnement à partir d'un appariement connu d'un contour avec un segment à l'aide de la relation suivante :

$$\hat{\theta}_C = \theta_V - \theta_P \quad (3.17)$$

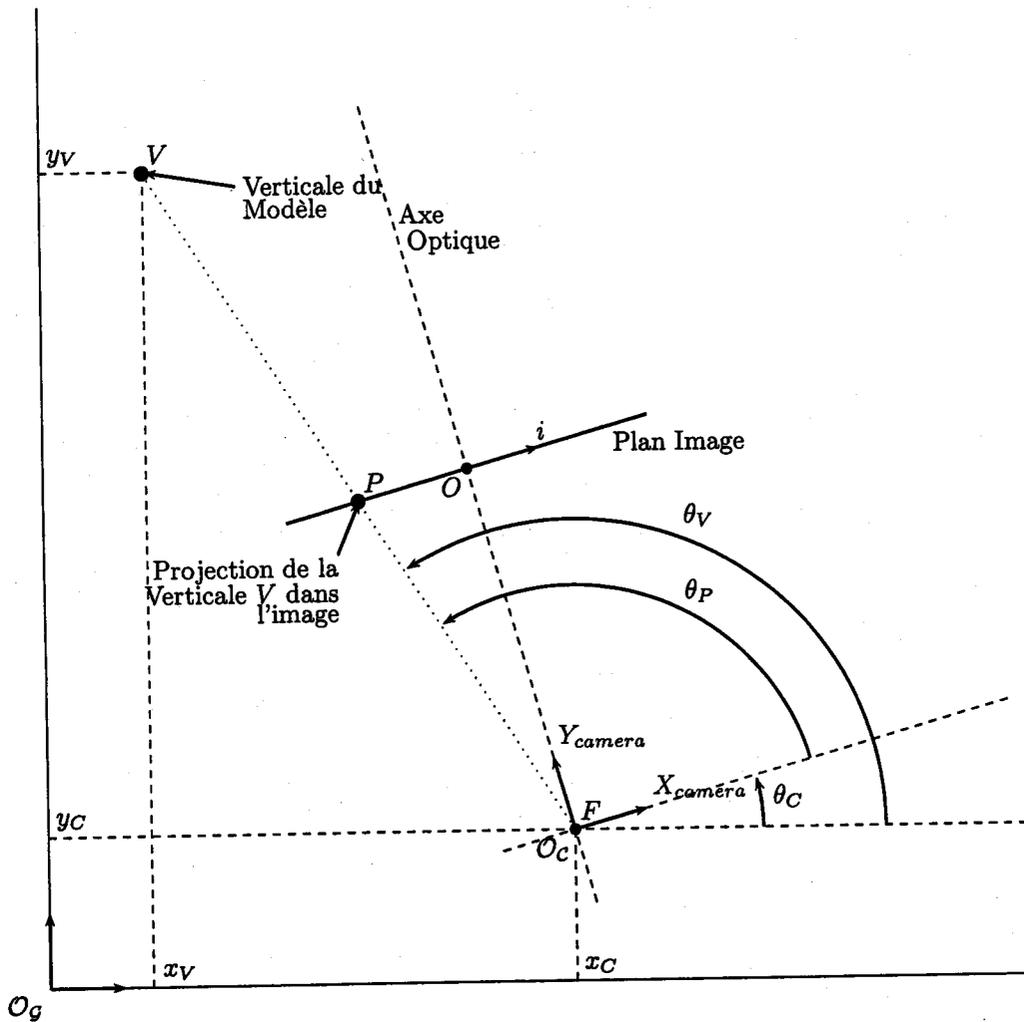


Figure 3.17 – Calcul de l'orientation de la caméra à partir d'une observation d'une seule verticale du modèle.

3.5.2 Ajustement de l'orientation

La détermination de l'orientation de la caméra vue au paragraphe précédent est maintenant adaptée pour ajuster une estimation approximative de celle-ci. Nous supposons disposer de l'orientation fournie par la méthode de localisation dynamique présentée au chapitre 2, soit $\hat{\theta}_{C_{us}}$ cette orientation. La figure 3.18 présente la scène vue de dessus.

Si l'orientation de la caméra était connue avec exactitude, alors les points V , P et F devraient être alignés. Or, si l'on observe la figure 3.18, nous pouvons remarquer que la droite \overline{FV} coupe le plan image en P' , alors que le contour détecté dans l'image se situe en P . Comme précédemment, nous pouvons déterminer les angles θ_V et θ_P . Mais $\hat{\theta}_{C_{us}}$ n'étant pas connue avec exactitude, alors l'égalité de la relation 3.16 n'est plus respectée. Cette différence angulaire, représentée par la variable τ sur la figure 3.18, se

mesure par le non alignement des points V , P et F . L'orientation estimée $\hat{\theta}_{Cvis}$ est un ajustement de $\hat{\theta}_{Cus}$ qui ramène le point P à l'abscisse du point P' dans le plan image, afin que les trois points V , P et F soient de nouveau alignés (figure 3.19). C'est-à-dire que le repère caméra tourne de la quantité τ autour de F . Nous obtenons la configuration de la figure 3.18. La nouvelle orientation estimée de la caméra est donc :

$$\hat{\theta}_{Cvis} = \hat{\theta}_{Cus} + \tau \quad (3.18)$$

Un couple contour/segment permet ainsi d'ajuster l'orientation de la caméra.

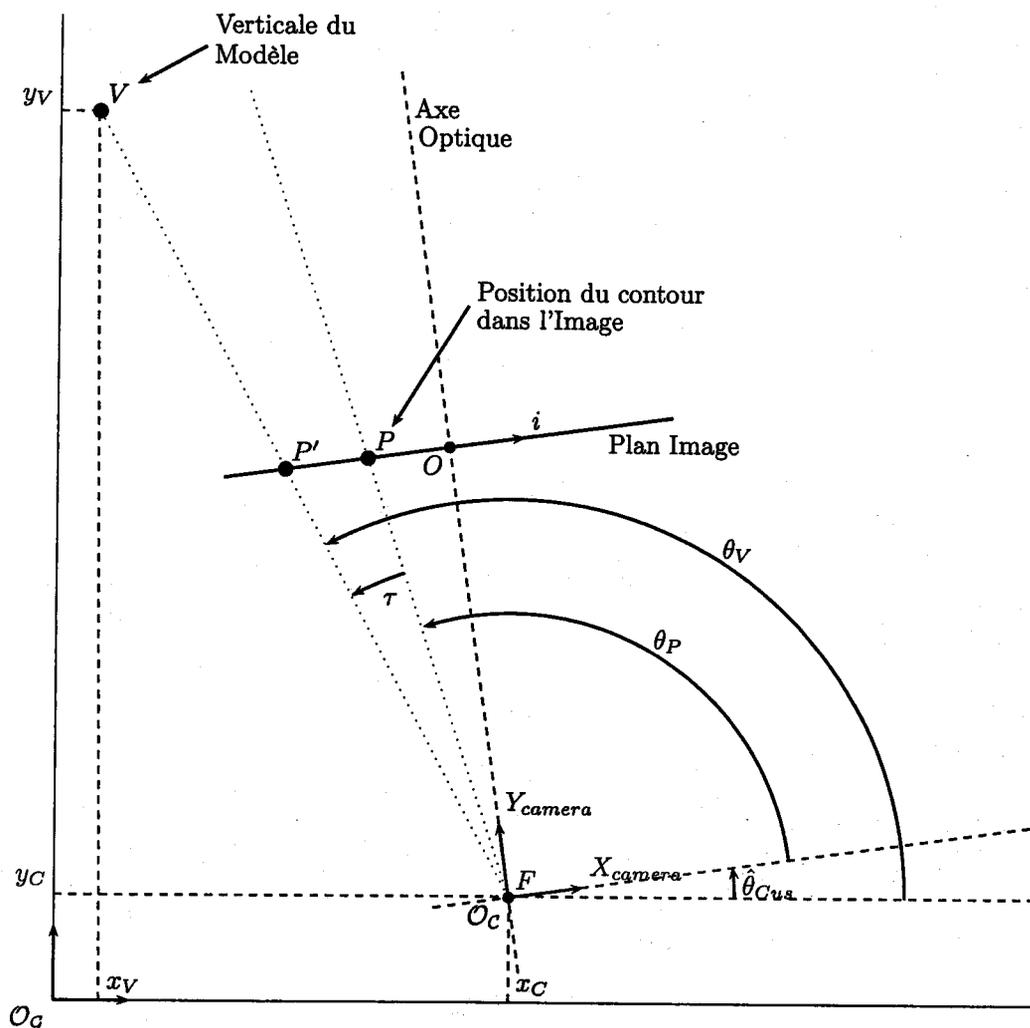


Figure 3.18 – Ajustement de l'orientation estimée de la caméra.

Lors de l'acquisition d'une image et après son traitement nous possédons un ensemble de contours dans l'image, mais nous n'avons aucune information concernant les correspondances entre les contours de l'image et les segments du modèle. Nous devons considérer toutes les combinaisons possibles des contours de l'image avec les segments du modèle, cette analyse est effectuée au moyen d'un arbre d'interprétation [Gri87, Kro89, Aid02].

3.6 Ajustement en orientation à partir de plusieurs observations

Cette partie concerne l'application des outils vus précédemment afin d'ajuster l'orientation estimée en utilisant l'observation de plusieurs verticales de l'environnement. Plusieurs contours sont présents dans l'image, il faut alors déterminer quelles sont les verticales du modèle correspondant à ces contours. Pour cela, nous mettons en œuvre un arbre d'interprétation que nous utilisons afin de retrouver, parmi toutes les combinaisons possibles d'appariements des données, celle qui permet de calculer l'ajustement le plus approprié.

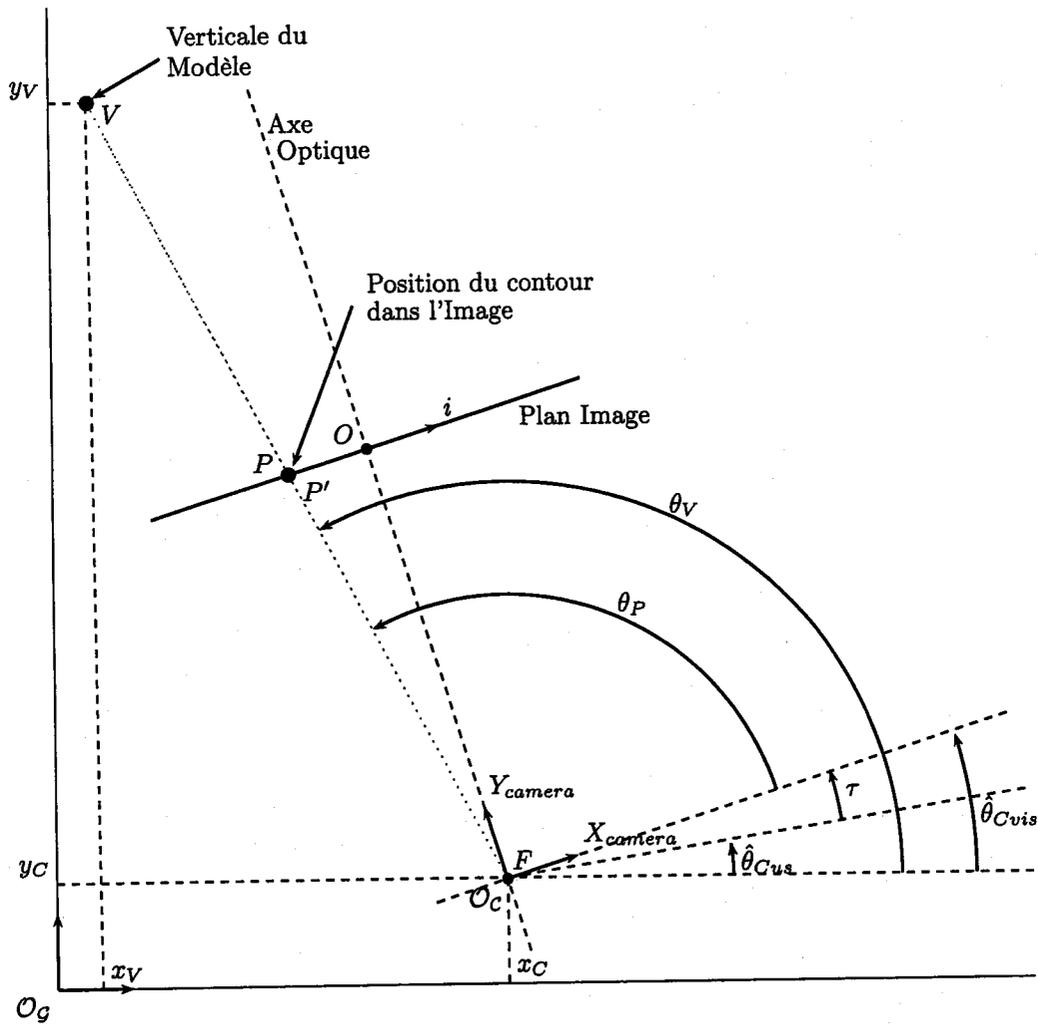


Figure 3.19 – Orientation estimée de la caméra après ajustement.

3.6.1 L'arbre d'interprétation

3.6.1.1 Présentation de l'arbre

L'arbre d'interprétation (figure 3.20) se compose de nœuds. Chaque nœud représente une association possible d'un des M contours (C_m) avec une des N verticales (V_n) du modèle, permettant de calculer un ajustement $\tau_{m,n}$ comme nous l'avons vu au paragraphe 3.5.2 traitant de l'ajustement de l'orientation à partir d'un appariement connu.

L'arbre débute par la création d'un nœud "racine" qui ne contient aucune information de mise en correspondance. Chaque nœud comporte $N - k$ descendants, où k est le niveau de l'arbre variant de 0 (racine) à M . A chaque niveau de l'arbre correspond un contour de l'image. Lorsque $k = M$, nous sommes sur un nœud terminal qui ne possède pas de nœuds fils.

Le chemin partant d'un des nœuds fils du nœud racine et atteignant un nœud terminal est une solution possible d'appariement des contours de l'image avec les segments du modèle. Lors de la création de l'arbre, chaque verticale du modèle n'est utilisée qu'une fois pour un chemin donné car une verticale du modèle ne peut donner lieu qu'à un contour dans l'image. Dans l'exemple de la figure 3.20 le chemin en gris qui propose les appariements (C_1, V_4) , (C_2, V_2) , (C_3, V_6) , (C_4, V_1) est une solution possible.

L'arbre propose autant de solutions possibles pour l'appariement des données des deux ensembles qu'il possède de chemins.

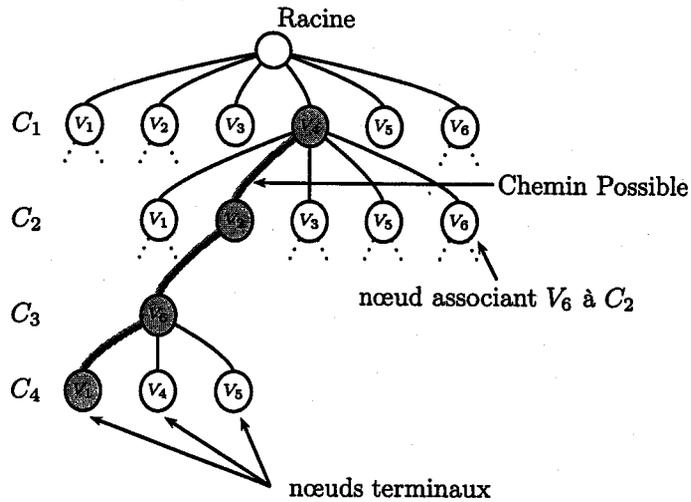


Figure 3.20 – Exemple d'arbre pour 4 contours (C_1 à C_4) et 6 verticales (V_1 à V_6).

3.6.1.2 Construction de l'arbre

La création de l'arbre est un processus récursif, qui s'exécute tant qu'il est possible de créer des nœuds fils. Pour réaliser, la récursivité du processus de création de l'arbre, une fonction appelée à partir d'un nœud courant permet la création de ses descendants. Cette même fonction est ensuite appelée par chacun des enfants qui viennent d'être créés. A mesure que la fonction est appelée, la profondeur (initialement nulle : nœud racine) de l'arbre croît. La création de l'arbre prend fin lorsque tous les nœuds situés à la profondeur M ont été créés (M étant le nombre de contours détectés dans l'image). Les nœuds situés à un même niveau de profondeur qui descendent d'un même nœud parent sont appelés des nœuds frères. Les N nœuds enfants de la racine de l'arbre étant créés, leurs descendants peuvent être créés. Compte tenu de l'aspect hiérarchique de l'arbre, les descendants d'un nœud courant sont créés en utilisant uniquement les verticales associées à ses frères. De cette manière nous garantissons qu'une verticale du modèle n'apparaît qu'une seule fois le long d'un chemin donné de l'arbre.

L'algorithme de la figure 3.21 décrit la fonction de création des nœuds enfants d'un nœud quelconque. Les paramètres de la fonction sont :

- N_{parent} : le nœud pour lequel nous allons créer les descendants,
- k : la profondeur du nœud N_{parent} dans l'arbre,
- V : la liste des verticales à utiliser pour la création des nœuds enfant, c'est-à-dire ceux qui ne sont pas encore utilisés dans le chemin menant jusqu'au nœud N_{parent} .

```

0 : Creer_Noeuds_Enfants( Nparent, k, V )
1 :     k = k + 1
2 :     SI k <= N ALORS,
3 :         Ccourant = Ck
4 :         POUR tous les Vi de V FAIRE
5 :             Vcourant = Vi
6 :             Ncourant = Creer_Noeud( Ccourant, Vcourant )
7 :             Lier_Noeud( Nparent, Ncourant )
8 :             Enlever_Verticale( V, Vcourant )
9 :             Creer_Noeuds_Enfants( Ncourant, k, V )
10 :        FIN POUR
11 :    SINON
12 :        SORTIR
13 :    FIN SI
14 : FIN Creer_Noeuds_Enfants
    
```

Figure 3.21 – Procédure : Creer_Noeuds_Enfants.

La liste V est initialement constituée de l'ensemble des verticales du modèle, k vaut 0 (profondeur du nœud racine) et $N_{parent} = Racine$. A la ligne 1 l'algorithme se place à la profondeur suivante dans l'arbre. Si la totalité des contours de l'image a été déjà utilisée ($k > N$), alors l'algorithme arrête l'évolution du chemin sur ce nœud (ligne 12). Sinon, il crée (ligne 6) à partir de toutes les verticales de V et du contour $C_{courant}$ correspondant à la profondeur k le nouveau nœud $N_{courant}$ qu'il lie (ligne 7) avec le nœud N_{parent} situé à la profondeur $(k - 1)$. Puis il supprime la verticale courante de la liste des verticales V (ligne 8) avant de lancer la procédure de création des nœuds enfants du nœud courant : $N_{courant}$ (ligne 9).

3.6.1.3 Interprétation de l'arbre

Sur la figure 3.22, sont représentés en traits continus les rayons partant du point focal de la caméra et passant par les abscisses des contours dans l'image. Ces faisceaux sont projetés autour de la position et de l'orientation exactes de la caméra. Sur cette même figure sont représentés en tirets les mêmes faisceaux mais projetés en utilisant une orientation estimée erronée. Dans ce cas nous pouvons voir que les trois ajustements sont similaires. Donc il nous faut déterminer le chemin de l'arbre qui donne des ajustements "semblables" pour ces différents nœuds ($\tau_1 = \tau_2 = \tau_3$). Sous cette condition, nous aurons alors la bonne correspondance des contours avec les verticales. Il ne restera plus qu'à ajuster l'orientation estimée de la caméra avec la valeur moyenne de l'ajustement calculée pour le chemin, soit :

$$\hat{\theta}_{Cvis} = \hat{\theta}_{Cus} + \bar{\tau} \quad (3.19)$$

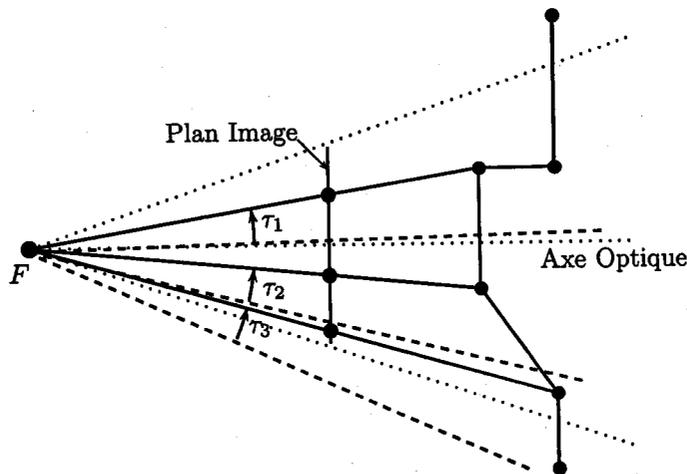


Figure 3.22 – Exemple où l'ajustement τ est identique pour différents appariements.

Au paragraphe 3.5.2, nous avons vu qu'il est possible d'ajuster l'orientation de la caméra à partir d'un couple segment/contour correctement apparié. L'arbre permet de donner toute les combinaisons possibles d'appariement des verticales du modèle avec les contours de l'image. Or, parmi toutes ces propositions, une seule est la bonne. En supposant que l'on connaisse la bonne combinaison représentée par un chemin, en calculant pour chaque nœud de ce chemin la valeur de l'ajustement à appliquer à l'orientation estimée, nous devrions trouver une valeur sensiblement identique. Ainsi, pour choisir dans l'arbre les bons appariements, nous allons chercher le chemin qui fournit sensiblement le même ajustement pour chacun de ses nœuds. Le critère que nous avons choisi est l'écart-type des ajustements le long d'un chemin i calculé par :

$$\sigma_i^2 = \frac{1}{M} \sum_{m=1}^M (\bar{\tau}_i - \tau_m)^2 \quad (3.20)$$

où i est l'indexe du chemin, M le nombre de contours dans l'image et $\bar{\tau}_i$ la valeur moyenne de l'ajustement pour le chemin i déterminée comme suit :

$$\bar{\tau}_i = \frac{1}{M} \sum_{m=1}^M \tau_m \quad (3.21)$$

3.6.1.4 Limitation de l'espace des segments

Lorsque le nombre de verticales dans le modèle est faible, l'utilisation de la totalité de celles-ci pour la recherche de leur correspondant dans le plan image (les contours détectés) ne pose pas de problème. Mais en pratique leur nombre peut être élevé. Le traitement décrit ici vise à limiter le nombre de verticales de l'environnement susceptibles d'être mises en correspondance avec les contours de l'image. Connaissant une estimation de la position du véhicule ainsi que de son orientation nous sélectionnons les verticales du modèle situées dans le champ de vision de la caméra. Cette zone de visibilité est définie à l'aide des paramètres de la caméra fixée sur le véhicule et de l'incertitude sur son orientation. Par ailleurs, les verticales masquées par des obstacles de l'environnement ne sont pas répertoriées (figure 3.23). La sélection des verticales du modèle s'effectue en deux étapes :

Etape 1 : Définition d'une zone géographique. Lors de cette étape, nous définissons une région de visibilité augmentée construite à partir du demi-angle d'ouverture de la caméra $\gamma/2$ augmenté de la quantité ρ afin de tenir compte des erreurs sur l'estimation de l'orientation et ainsi être sûr de sélectionner les verticales du modèle correspondant aux contours détectés dans l'image. Dans les essais résultant du paragraphe 2.8, nous avons constaté que l'erreur absolue en orientation n'excède pas (en moyenne) les 10° . Pour cela nous avons choisi d'augmenter le demi angle d'ouverture de la caméra ($\gamma/2$) de la valeur $\rho = 10^\circ$. Les verticales du modèle ayant leurs coordonnées (x, y) incluses dans cette zone sont sélectionnées afin de constituer une liste L des verticales susceptibles d'être vues par la caméra. Dans l'exemple de la figure 3.23 les verticales V_0 à V_{12} sont sélectionnées et intégrées dans la liste L.

Etape 2 : Suppression des verticales masquées. Lors de cette seconde étape, nous vérifions que chaque verticale du modèle présente dans la liste L est effectivement visible par la caméra. Pour cela, nous utilisons les segments horizontaux définissant les limites de l'espace libre (c'est-à-dire les contours des obstacles) du modèle de l'environnement utilisé au chapitre 2. Pour chaque verticale V_i de la liste L, nous vérifions que la droite $\overline{FV_i}$ ne croise aucun obstacle, c'est-à-dire qu'il n'existe aucun point d'intersection entre cette droite et les limites des obstacles. S'il existe une intersection la verticale V_i est retirée de la liste L. Dans l'exemple de la figure 3.23 les segments V_2 et V_3 sont supprimés car les droites $\overline{FV_2}$ et $\overline{FV_3}$, représentées en pointillés sur la figure 3.23, coupent des frontières de l'espace libre.

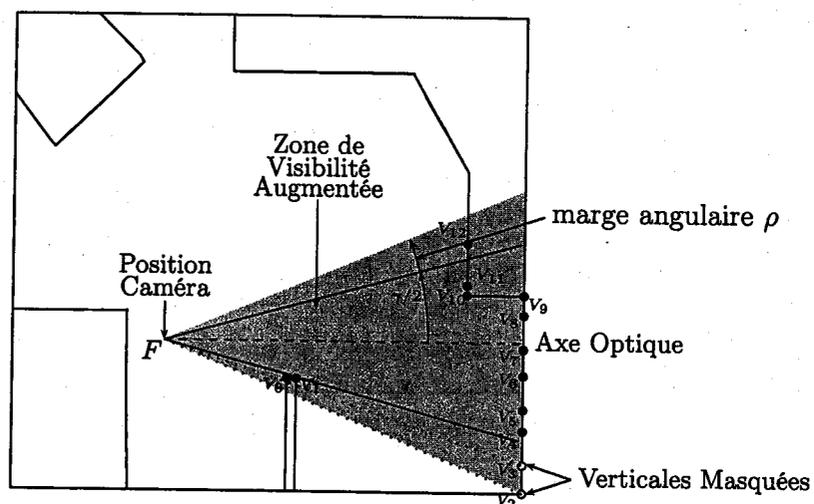


Figure 3.23 – Exemple de sélection de verticales du modèle – définition de la zone de visibilité en fonction des paramètres connus de la caméra, de sa position et de son orientation estimées dans le repère global.

Cette technique permet de réduire le nombre de mises en correspondance possibles en réduisant le nombre N de verticales du modèle. L'étape suivante consiste à apparier les contours de l'image avec les verticales sélectionnées dans le modèle selon la méthode exposée au paragraphe 3.6.1.1.

3.6.1.5 Elagage de l'arbre

En construisant l'arbre comme décrit au paragraphe 3.6.1.1, nous obtenons toutes les combinaisons possibles des M contours de l'image avec les N verticales du modèle. Or, parmi les solutions proposées, certaines ne sont pas cohérentes dans le sens où elles "croisent" des projections. Autrement dit, en considérant une scène située devant la caméra où la verticale V_1 se projetterait en C_1 et la verticale V_2 en C_2 , un appariement incohérent proposerait d'associer V_1 avec C_2 et V_2 avec C_1 (figure 3.24). Ces associations ne sont pas réalisables et ne doivent pas figurer dans l'arbre.

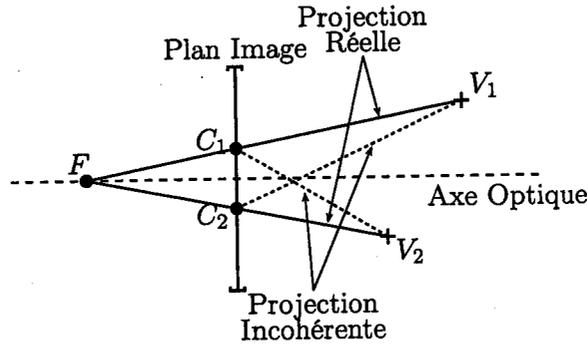


Figure 3.24 – Exemple d'incohérence d'un appariement.

La solution que nous proposons ici consiste à introduire un test de cohérence de l'appariement proposé par un nœud fils compte tenu de l'appariement proposé par son nœud parent. Pour cela, nous apportons une modification à la procédure de création de l'arbre d'interprétation (figure 3.21) au niveau de l'étape qui crée un nouveau nœud (ligne 6) : nous vérifions la cohérence des appariements représentés dans le nœud courant (N_{courant}) et dans le nœud parent (N_{parent}) avant d'ajouter le nouveau nœud dans l'arbre.

Nous présentons le principe de ce test sur un exemple. Considérons la configuration proposée par la figure 3.25. Supposons qu'au niveau m , le contour image C_B soit associé à la verticale V_2 , alors au niveau $m + 1$, a priori, les contours C_A et C_C peuvent être appariés à V_1 ou V_3 .

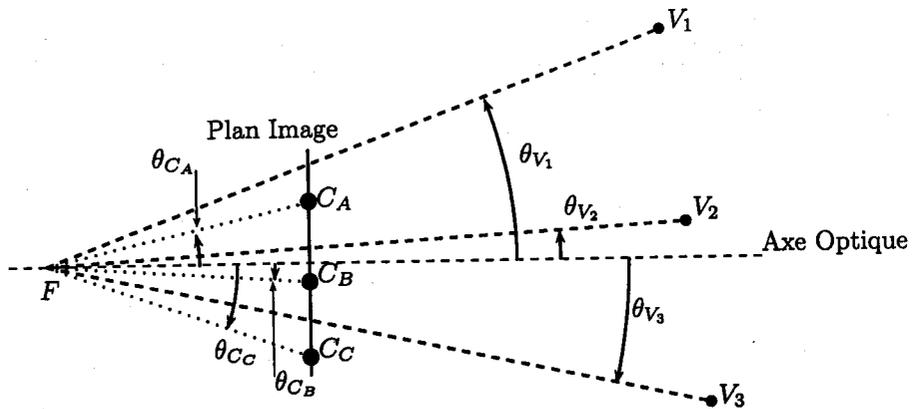


Figure 3.25 – Principe de la recherche de la cohérence de deux appariements successifs d'un chemin.

Afin de déterminer la cohérence des appariements successifs le long d'un chemin, nous déterminons les angles entre les contours des niveaux successifs et les angles entre les deux verticales que nous voulons leur apparié. Dans le cas de la figure 3.25, si nous voulons évaluer la cohérence des appariements avec le contour C_A , sachant que son nœud parent est (C_B, V_2) , nous définissons les angles suivants :

$$\begin{aligned}\beta_{BA} &= \theta_{CB} - \theta_{CA} \\ \beta_{21} &= \theta_{V1} - \theta_{V2} \\ \beta_{23} &= \theta_{V3} - \theta_{V2}\end{aligned}$$

β_{BA} et β_{21} étant de même signe, le nœud appariant le contour C_A et la verticale V_1 est créé et lié au nœud parent (C_B, V_2). Par contre β_{BA} et β_{23} sont de signes opposés. Dans ce cas, le nœud appariant le contour C_A et la verticale V_3 n'est pas créé et l'évolution du chemin dans cette direction est stoppée. Graphiquement, nous obtenons le résultat de la figure 3.26.

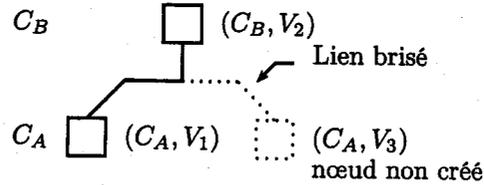


Figure 3.26 – Résultat sur la création de l'arbre.

Cette technique de construction de l'arbre permet d'éliminer les solutions incohérentes, qu'il est inutile de créer et d'évaluer. La taille de l'arbre est diminuée ainsi que le temps de traitement nécessaire pour la recherche du meilleur ajustement en orientation. En effectuant cet élagage, certains chemins seront réduits, c'est-à-dire qu'ils n'utiliseront pas la totalité des contours détectés dans l'image. Les chemins peuvent être de différentes tailles et la sélection du chemin proposant le meilleur ajustement doit en tenir compte. En effet le critère de l'écart type des ajustements, que nous considérons pour établir le bon appariement des contours et des verticales, n'est plus significatif lorsque le chemin est de petite taille (inférieure ou égale à deux nœuds). Nous imposerons donc un seuil de trois nœuds minimum pour les chemins qui seront examinés afin de choisir celui présentant le plus petit écart type.

Nous avons établi la procédure de détermination de l'orientation du fauteuil à partir des informations visuelles. Le paragraphe suivant analyse la complexité de son algorithme.

3.7 Etude de la complexité de l'algorithme

Comme nous l'avons vu précédemment, un nœud de l'arbre permet le calcul de l'ajustement en orientation pour un couple (C_m, V_n) donné. L'algorithme doit donc déterminer $\tau(m, n)$ pour chaque nœud de l'arbre. Ainsi la complexité de l'algorithme dépend directement du nombre de contours M détectés dans l'image et du nombre de verticales N sélectionnées. Nous allons voir dans les deux parties suivantes comment déterminer le nombre de nœuds pour un arbre total puis pour un arbre élagué constitués à partir de N verticales du modèle et M contours de l'image.

3.7.1 Nombre de nœuds d'un arbre total

L'arbre classique ne tient pas compte du lien existant entre l'ordre d'apparition des contours dans l'image et l'ordre des verticales du modèle de la scène située devant le robot. Il considère donc toutes les associations des M contours avec les N verticales comme possibles.

A un niveau² k de l'arbre ($k \in [1 \dots M]$) le nombre de nœuds créés correspond au nombre d'associations de k verticales prises parmi les N avec les k premiers contours de l'arbre. Autrement dit, il s'agit du nombre de suites ordonnées de k verticales prises parmi N , soit en utilisant la formule des arrangements :

$$A_N^k = \frac{N!}{(N-k)!} \quad (3.22)$$

Le nombre de nœuds obtenus pour l'arbre développé jusqu'au niveau k est la somme du nombre de nœuds créés à chaque niveau inférieur à k :

$$S_t(k) = \sum_{i=1}^k \frac{N!}{(N-i)!} \quad (3.23)$$

²Nous rappelons que la profondeur de l'arbre est définie par le nombre de contours dans l'image et que le niveau 0 correspond à la racine de l'arbre

Le nombre total de nœuds dans l'arbre est alors donné pour $k = M$:

$$S_t(M) = \sum_{i=1}^M \frac{N!}{(N-i)!} \quad (3.24)$$

3.7.2 Nombre de nœuds de l'arbre élagué

Quand l'ordre des contours dans l'image impose l'ordre des verticales dans le modèle, le nombre de nœuds créés au niveau k correspond au nombre de manières de prendre k verticales parmi N . En effet, les verticales étant sélectionnées, les contours qui leur sont affectés sont imposés par la physionomie du lieu, c'est-à-dire que l'ordre dans lequel elles sont choisies n'importe plus. Le nombre de nœuds au niveau k correspond donc au nombre de sous ensemble de k éléments pris parmi N , soit en utilisant la formule des combinaisons :

$$C_N^k = \frac{N!}{k!(N-k)!} \quad (3.25)$$

Alors le nombre total de nœuds dans l'arbre élagué s'écrit :

$$S_e(M) = \sum_{i=1}^M \frac{N!}{i!(N-i)!} \quad (3.26)$$

3.7.3 Comparaison et conclusion

Afin d'évaluer l'apport de l'élagage sur la complexité de la méthode d'ajustement nous allons comparer l'évolution du nombre de nœuds contenus dans l'arbre pour le cas d'un arbre total et d'un arbre élagué. Les tableaux 3.1 et 3.2 donnent les valeurs obtenues pour les deux types d'arbre. Les colonnes correspondent au nombre n de verticales sélectionnées et les lignes correspondent au nombre m de contours détectés dans l'image.

Il apparaît que le nombre de nœuds est considérablement réduit lors de l'utilisation d'un arbre élagué, ce qui permet d'utiliser un maximum d'information sans accroître de façon démesurée la taille de l'arbre. De plus le fait d'élaguer l'arbre en fonction de l'ordre d'apparition des contours et des verticales ne peut pas éliminer le bon appariement de l'arbre mais devrait au contraire éviter des situations ambiguës causées par des entités proches (contours de l'image ou verticales du modèle).

La figure 3.27 donne l'évolution du nombre de nœuds contenus dans un arbre total et dans un arbre élagué construits en utilisant sept contours de l'image. En abscisse est donné le nombre de verticales sélectionnées dans le modèle et en ordonnée le nombre de nœuds contenus dans l'arbre (en échelle log). Nous pouvons noter que l'explosion combinatoire provoquée par l'utilisation d'un arbre total est évitée avec un arbre élagué.

$m \backslash n$	1	2	3	4	5	6	7	8	9	10	11
1	1	2	3	4	5	6	7	8	9	10	11
2	0	4	9	16	25	36	49	64	81	100	121
3	0	0	15	40	85	156	259	400	585	820	1111
4	0	0	0	64	205	516	1099	2080	3609	5860	9031
5	0	0	0	0	325	1236	3619	8800	18729	36100	64471
6	0	0	0	0	0	1956	8659	28960	79209	187300	397111
7	0	0	0	0	0	0	13699	69280	260649	792100	2060311
8	0	0	0	0	0	0	0	109600	623529	2606500	8713111
9	0	0	0	0	0	0	0	0	986409	6235300	28671511

Tableau 3.1 – Nombre de nœuds de l'arbre total en fonction du nombre m de contours et du nombre n de verticales

$m \backslash n$	1	2	3	4	5	6	7	8	9	10	11
1	1	2	3	4	5	6	7	8	9	10	11
2	0	3	6	10	15	21	28	36	45	55	66
3	0	0	7	14	25	41	63	92	129	175	231
4	0	0	0	15	30	56	98	162	255	385	561
5	0	0	0	0	31	62	119	218	381	637	1023
6	0	0	0	0	0	63	126	246	465	847	1485
7	0	0	0	0	0	0	127	254	501	967	1815
8	0	0	0	0	0	0	0	255	510	1012	1980
9	0	0	0	0	0	0	0	0	511	1022	2035

Tableau 3.2 – Nombre de nœuds de l'arbre élagué en fonction du nombre m de contours et du nombre n de verticales

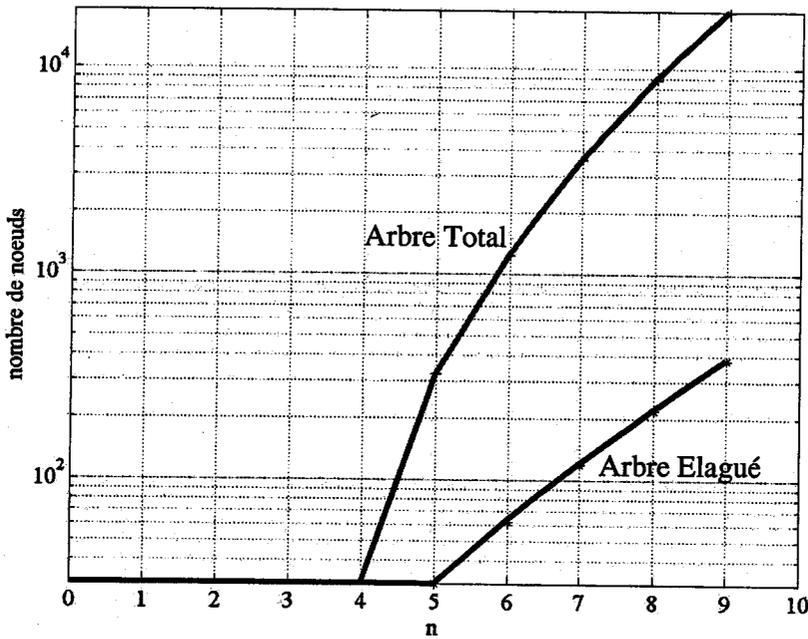


Figure 3.27 – Evolution du nombre de nœuds pour 5 contours dans l'image avec un arbre total et un arbre élagué.

3.8 Influence d'une erreur en position

Comme nous l'avons vu précédemment au paragraphe 3.5.2, le calcul de l'ajustement en orientation dépend de la position estimée (\hat{x}_C, \hat{y}_C) de la caméra dans le repère global. C'est lors du calcul de l'angle θ_V , donné par la relation 3.14, qu'une erreur en position peut perturber la détermination de l'ajustement. Nous rappelons ici son calcul :

$$\hat{\theta}_V = \arctan \left(\frac{y_V - \hat{y}_C}{x_V - \hat{x}_C} \right) \quad (3.27)$$

où (x_V, y_V) sont les coordonnées de la verticale dans le repère global, supposées connues sans erreur. La position de la caméra n'étant pas connue avec exactitude, nous pouvons écrire :

$$\begin{cases} \hat{x}_C = x_C + \varepsilon_x \\ \hat{y}_C = y_C + \varepsilon_y \end{cases} \quad (3.28)$$

où (x_C, y_C) sont les coordonnées réelles de la caméra dans le référentiel global et ε_x et ε_y sont respectivement l'erreur sur l'abscisse et l'erreur sur l'ordonnée.

En remplaçant \hat{x}_C et \hat{y}_C par leur équation respective dans la relation 3.27, nous obtenons :

$$\hat{\theta}_V = \arctan \left(\frac{y_V - y_C - \varepsilon_y}{x_V - x_C - \varepsilon_x} \right) \quad (3.29)$$

Si la position de la caméra n'était pas perturbée, alors nous aurions l'angle θ_V de référence :

$$\theta_V = \arctan \left(\frac{y_V - y_C}{x_V - x_C} \right)$$

qui correspond exactement à l'équation 3.14. Nous pouvons ainsi déterminer l'erreur absolue :

$$E = \left| \theta_V - \hat{\theta}_V \right| \quad (3.30)$$

$$E = \left| \arctan \left(\frac{y_V - y_C}{x_V - x_C} \right) - \arctan \left(\frac{y_V - y_C - \varepsilon_y}{x_V - x_C - \varepsilon_x} \right) \right| \quad (3.31)$$

qui peut aussi s'exprimer par :

$$E = \left| \varepsilon_x \frac{\partial \theta_V}{\partial x_C} + \varepsilon_y \frac{\partial \theta_V}{\partial y_C} \right| \quad (3.32)$$

où $\frac{\partial \theta_V}{\partial x_C}$ (respectivement $\frac{\partial \theta_V}{\partial y_C}$) est la dérivée partielle de θ_V par rapport à x_C (respectivement y_C) :

$$\frac{\partial \theta_V}{\partial x_C} = \frac{y_V - y_C}{(x_V - x_C)^2 \left(1 + \frac{(y_V - y_C)^2}{(x_V - x_C)^2} \right)} \quad (3.33)$$

$$\frac{\partial \theta_V}{\partial y_C} = - \frac{x_V - x_C}{(x_V - x_C)^2 \left(1 + \frac{(y_V - y_C)^2}{(x_V - x_C)^2} \right)} \quad (3.34)$$

Ce qui donne pour E :

$$E = \left| \frac{\varepsilon_x (y_V - y_C) - \varepsilon_y (x_V - x_C)}{(x_V - x_C)^2 \left(1 + \frac{(y_V - y_C)^2}{(x_V - x_C)^2} \right)} \right| \quad (3.35)$$

En posant $\Delta x = x_V - x_C$ et $\Delta y = y_V - y_C$ dans l'équation 3.35, nous obtenons :

$$E = \left| \frac{\varepsilon_x \Delta y - \varepsilon_y \Delta x}{\Delta x^2 \left(1 + \frac{\Delta y^2}{\Delta x^2} \right)} \right| \quad (3.36)$$

qui se simplifie en :

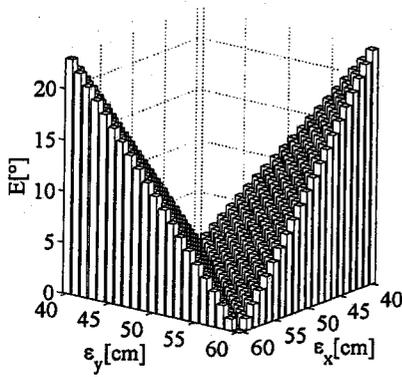
$$E = \left| \frac{\varepsilon_x \Delta y - \varepsilon_y \Delta x}{\Delta x^2 + \Delta y^2} \right| \quad (3.37)$$

En observant l'équation 3.37, nous remarquons que plus la distance entre la verticale du modèle et la caméra est importante, moins l'estimation de $\hat{\theta}_V$ sera sensible aux erreurs de position. De plus, tant que $|\Delta y| \ll |\Delta x|$ l'erreur sur y_C prédomine devant l'erreur sur x_C . Réciproquement si $|\Delta y| \gg |\Delta x|$ alors c'est l'erreur sur x_C qui prédomine devant l'erreur sur y_C .

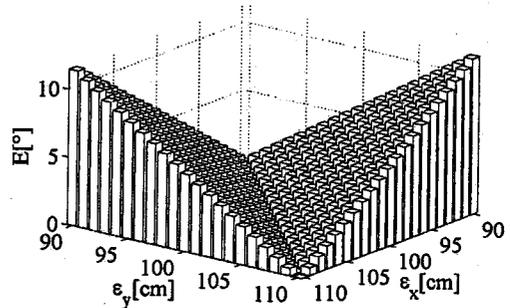
Lorsque la position relative entre la verticale observée et la caméra est telle que $\Delta x = \Delta y$, alors l'erreur d'ajustement reste faible tant que les erreurs ε_x et ε_y restent de même amplitude, mais croît rapidement dans le cas contraire. Elle atteint les 3° lorsque ces erreurs sont de l'ordre de 5 cm, sauf si elles sont de même signe. Par exemple si $\Delta x = \Delta y = 100$ cm, et $\varepsilon_y = -\varepsilon_x = 5$ cm, l'erreur E est de 2.9°. Etant donné que l'erreur en position résiduelle après application de la méthode de correction par ultrasons ne devrait pas excéder les 5 cm, nous aurions théoriquement atteint l'objectif fixé permettant de maintenir l'erreur de localisation en dessous des 3° si les verticales observées sont situées à 1 m de la caméra.

Les figures 3.28(a) à 3.28(d) donnent des exemples d'évolution de l'erreur en fonction de l'erreur en position pour différentes positions relatives entre la caméra et le segment vertical du modèle. Sur ces figures, l'erreur E est exprimée en degrés.

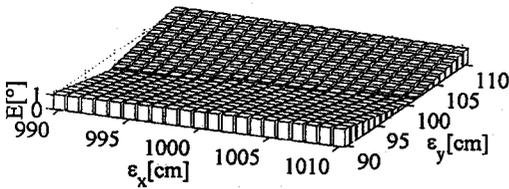
La figure 3.28(c) montre que lorsque $\Delta x \gg \Delta y$, alors l'erreur sur la coordonnée y_C prédomine sur l'erreur sur la coordonnée x_C : une variation de ε_y engendre une variation de E alors qu'une variation de ε_x a peu d'influence sur l'erreur d'ajustement E . Si la coordonnée relative Δx (respectivement Δy) est nulle alors l'erreur d'ajustement ne dépend plus que de ε_x (respectivement ε_y) (figure 3.28(d)).



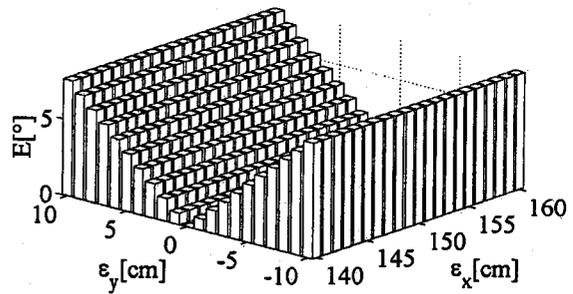
(a) $\Delta x = 0.5 m$ et $\Delta y = 0.5 m$



(b) $\Delta x = 1 m$ et $\Delta y = 1 m$



(c) $\Delta x = 10 m$ et $\Delta y = 1 m$



(d) $\Delta x = 1.5 m$ et $\Delta y = 0 m$

Figure 3.28 – Evolution de l'erreur en fonction de l'erreur en position pour différentes positions relatives caméra/segment vertical.

3.9 Etude en simulation

3.9.1 Objectif et contexte de la simulation

L'objectif de la simulation est d'évaluer les performances de la méthode d'ajustement de l'orientation. Lors de cette étude nous maîtrisons toutes les variables expérimentales, c'est-à-dire que nous connaissons la position et l'orientation exacte du fauteuil dans son environnement. Connaissant la position exacte de la caméra, l'image qu'elle devrait fournir est simulée en utilisant les informations contenues dans le modèle de l'environnement. Dans une première partie nous réalisons une étude qualitative de l'ajustement en orientation. Nous y présentons les résultats pour différentes localisations estimées simulées (celles que donneraient la correction par capteurs à ultrasons), obtenues en ajoutant une erreur sur la position réelle de l'ordre de $5 cm$ et une erreur sur l'orientation réelle de l'ordre de 10° . Dans une seconde partie nous évaluons la plage d'erreur en orientation admissible compensable par la méthode d'ajustement, en considérant une erreur de position non nulle. Enfin dans une dernière étape, nous évaluons l'influence d'une erreur en position sur l'ajustement en orientation.

La figure 3.29 donne une représentation vue de dessus de l'environnement utilisé pour la simulation ainsi que les positions "réelles" du fauteuil lors des simulations. Les petits cercles représentent les positions

cartésiennes des segments verticaux. Ces segments représentent soit l'intersection de deux plans limites de l'espace libre soit des motifs présents sur ces plans (limite d'un panneau, bord d'un poster...). Ce second type de caractéristique explique la présence de plusieurs segments de long du mur droit. La taille de l'espace de travail est $4 \times 4 m^2$ et il est peu encombré.

Comme nous faisons l'hypothèse d'une erreur en orientation initiale de l'ordre de 10° , la sélection des verticales est réalisée en augmentant le demi angle d'ouverture de la caméra de 10° (voir paragraphe 3.6.1.4).

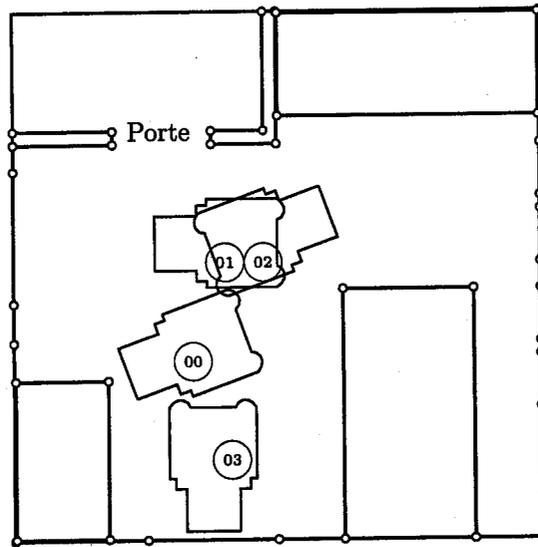


Figure 3.29 – Positions du robot dans l'environnement utilisé pour l'étude en simulation de la méthode d'ajustement.

3.9.2 Etude qualitative de la méthode d'ajustement

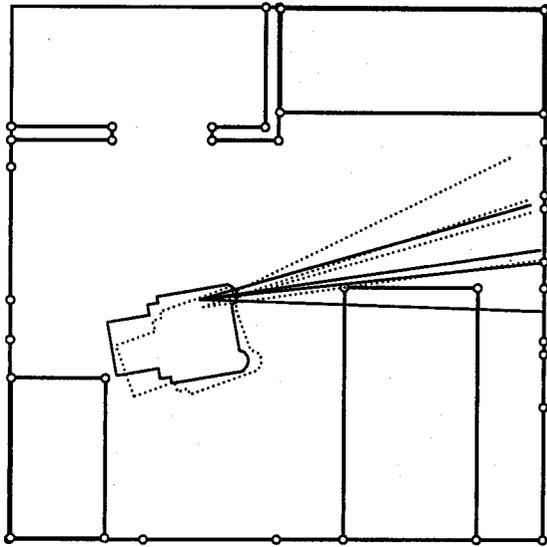
Nous décrivons dans ce paragraphe les résultats obtenus en simulation pour quatre positions différentes du véhicule dans l'environnement (figure 3.29). La localisation estimée prise pour initialiser la procédure est définie en ajoutant à la position réelle une erreur de l'ordre de celle obtenue après traitement des données des capteurs à ultrasons. Le tableau 3.3 donne les localisations "réelles" du véhicule ainsi que les localisations estimées et l'orientation ajustée.

essai	Localisation Réelle			Localisation Estimée			Ajustement	
	x_{re} [cm]	y_{re} [cm]	θ_{re} [°]	x_{est} [cm]	y_{est} [cm]	θ_{est} [°]	θ_{ajt} [°]	$ \epsilon_\theta $ [°]
00	150	150	290	143	155	280	288.5	1.5
01	175	225	110	180	220	100	108.5	1.5
02	175	225	270	170	230	280	268.9	1.1
03	150	75	0	160	70	-10	2.6	2.6

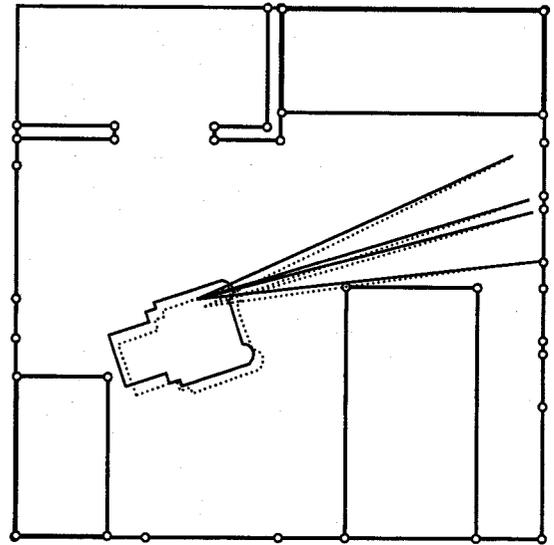
Tableau 3.3 – Localisations réelles et estimées utilisées pour l'évaluation en simulation de l'ajustement de l'orientation.

Les figures 3.30 à 3.33 illustrent les résultats obtenus ; sur ces figures les droites en pointillés sont celles joignant le centre optique de la caméra (position réelle) aux segments de l'environnement. Les traits pleins représentent ces mêmes droites mais pour la caméra en position estimée avant correction (a) ou après correction (b). L'objectif de l'ajustement est de faire correspondre au mieux les droites en pointillés et les droites continues. Les figures 3.30(b) à 3.33(b) donnent les résultats des ajustements en orientation. Dans les quatre essais présentés, nous voyons qu'après ajustement la direction d'observation est retrouvée car

les droites coïncident malgré une erreur en position. Lors de ces essais en simulations l'erreur résiduelle en orientation, après ajustement, est en moyenne de 1.7° avec un maximum de 2.6° pour le dernier essai. Dans celui-ci on peut remarquer la présence de deux contours très proches dans l'image que l'algorithme arrive à distinguer et à apparier correctement à leurs correspondants dans le modèle de l'environnement. Le temps de calcul moyen nécessaire à l'ajustement est de 250 ms .

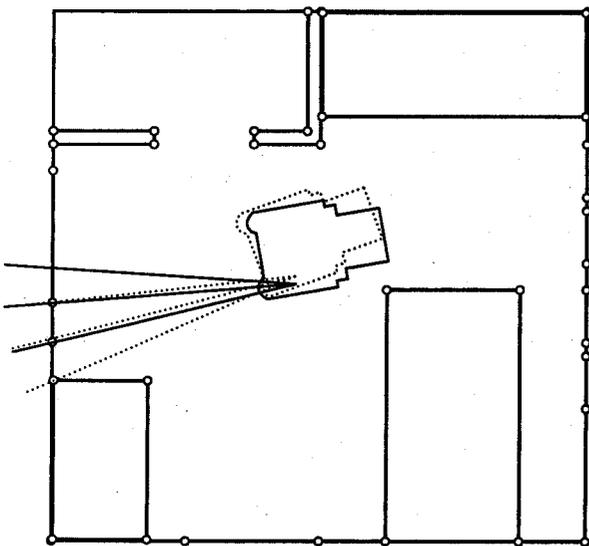


(a) Localisation réelle et estimée.

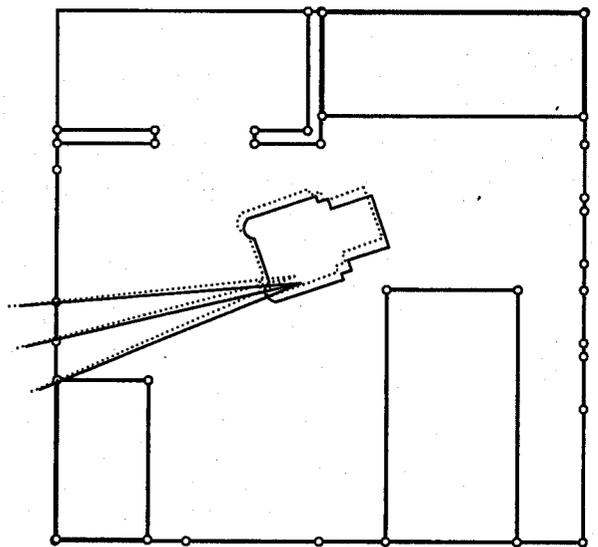


(b) Localisation réelle et corrigée.

Figure 3.30 – Simulation 00.



(a) Localisation réelle et estimée.



(b) Localisation réelle et corrigée.

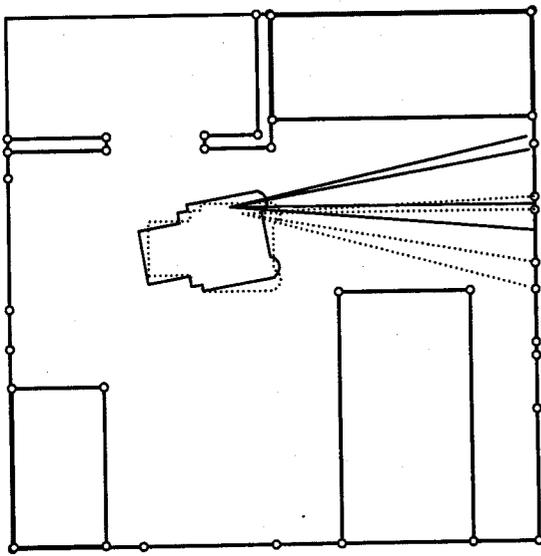
Figure 3.31 – Simulation 01.

3.9.3 Synthèse

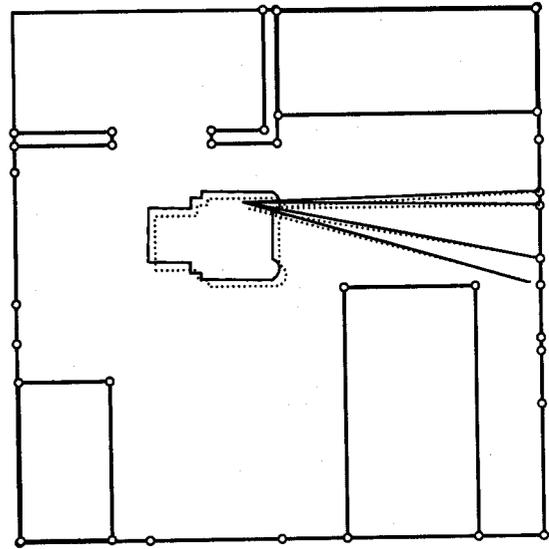
Les essais réalisés en simulation ont tous conduit à un ajustement permettant d'atteindre l'objectif fixé qui était de réduire l'erreur d'estimation en orientation à une valeur inférieure à 3° , et cela même en présence d'une erreur sur l'estimation de la position. Cette erreur en position ne devrait pas dépasser les 5 cm lorsque la correction de la position odométrique par capteur à ultrasons se déroule correctement.

Nous avons tout de même dépassé cette limite dans le dernier exemple, ce qui n'a pas fait échouer la méthode d'ajustement.

Ces essais mettent en évidence les potentialités de l'approche qui à partir de données simulées proches des conditions réelles permet d'obtenir les résultats voulus.

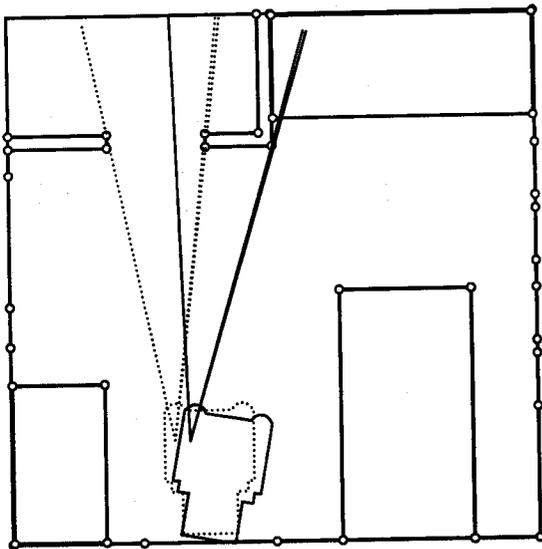


(a) Localisation réelle et estimée.

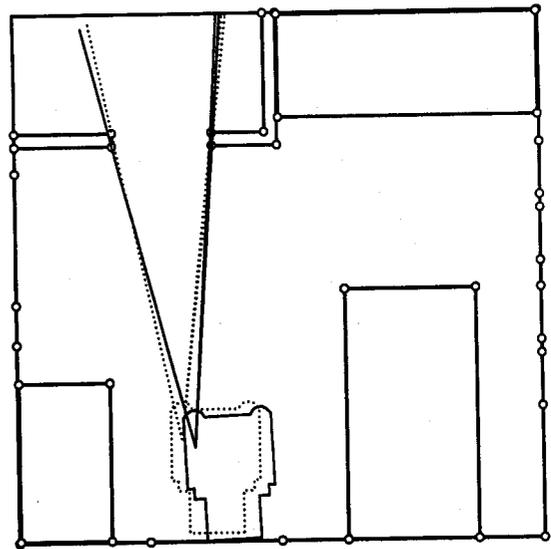


(b) Localisation réelle et corrigée.

Figure 3.32 – Simulation 02.



(a) Localisation réelle et estimée.



(b) Localisation réelle et corrigée.

Figure 3.33 – Simulation 03.

3.9.4 Estimation de la plage d'erreur admissible sur l'orientation estimée

Nous allons évaluer dans ce paragraphe la plage de variation de l'erreur initiale sur l'orientation que la méthode d'ajustement arrive à compenser. Nous allons considérer les mêmes cas que dans le paragraphe précédent en ce qui concerne la position estimée, par contre nous allons faire varier l'orientation estimée autour de l'orientation réelle tant que la méthode d'ajustement reste opérante.

Le tableau 3.4 résume les résultats obtenus pour chacune des quatre simulations. Les colonnes deux à quatre donnent la localisation réelle du fauteuil, les colonnes cinq et six donnent la position estimée du fauteuil et les colonnes sept et huit donnent les bornes de la plage de variation de l'orientation estimée pour laquelle l'erreur en orientation après ajustement est inférieure à 3°.

essai	Localisation Réelle			Position Estimée		$\Delta\theta_{min}$ [deg]	$\Delta\theta_{max}$ [deg]
	x_{re}	y_{re}	θ_{re}	x_{est}	y_{est}		
00	150	150	290	143	155	-22	16
01	175	225	110	180	220	-24	12
02	175	225	270	170	230	-25	11
03	150	75	0	160	70	-12	5

Tableau 3.4 – Evaluation de la plage de variation de l'erreur admissible sur l'orientation estimée avant l'ajustement par vision monoculaire.

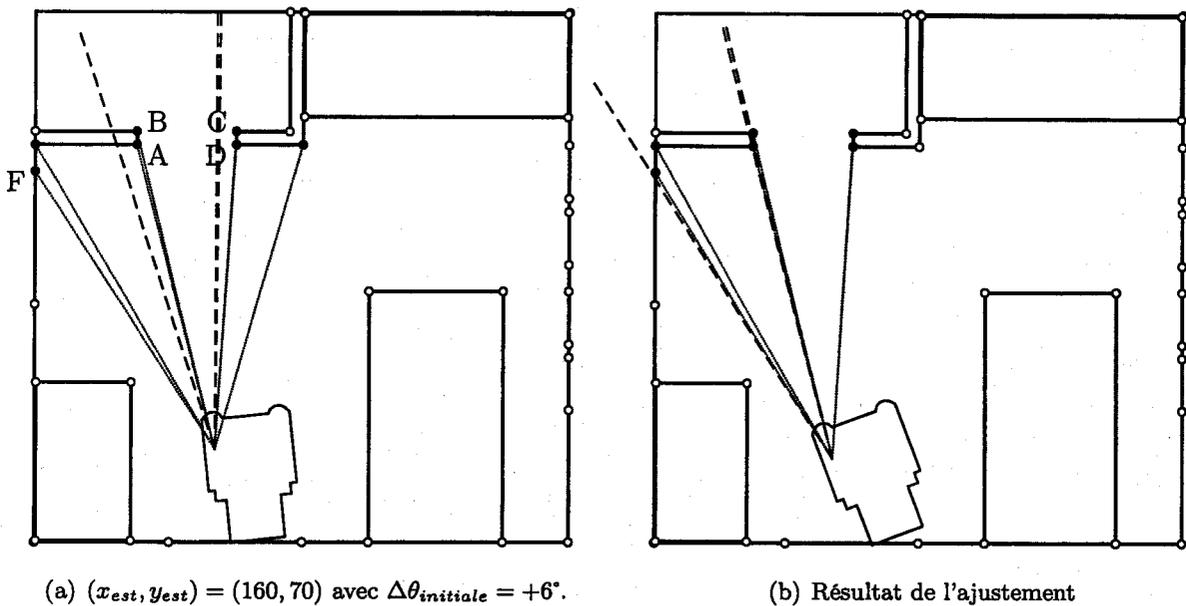


Figure 3.34 – Simulation 03 : Limite de la plage de variation de l'erreur initiale en orientation.

Pour les trois premiers essais la plage d'erreur admissible est supérieure à $\pm 10^\circ$ bien que la position estimée soit perturbée, en revanche dans le cas 03, lorsque $\theta_{us} \geq 6^\circ$, l'ajustement ne fonctionne plus. La figure 3.34(a) donne la configuration estimée du fauteuil avant l'ajustement, et la figure 3.34(b) donne le résultat de l'ajustement. Sur la figure 3.34(b) nous constatons que les trois rayons générés par les contours détectés dans l'image correspondent bien avec trois des verticales du modèle, mais ce ne sont pas les bonnes. Lors de la simulation de l'image, l'abscisse du véhicule était de 150 cm, or de cette position les points C et D sont plus différenciables que les points A et B, c'est-à-dire qu'ils n'apparaissent pas sur le même axe. Tandis que depuis la position estimée ($x_{est} = 160$ cm) les points C et D paraissent alignés et les points A et B sont différenciables, ce qui induit une erreur de mise en correspondance. Ceci est confirmé par le fait que la configuration du triplet de verticales sélectionnées (A,B,F) dans le mauvais appariement est semblable à la configuration des trois verticales (D,C,A) qui correspondent réellement aux contours projetés dans le plan image. Il s'agit donc d'une particularité de cet environnement dont on peut supposer qu'elle ne se reproduira pas fréquemment en environnement réel.

3.9.5 Evaluation de l'influence d'une erreur en position

Dans ce paragraphe, nous testons sur les exemples de simulation la validité des résultats obtenus au paragraphe 3.8. Pour cela nous initialisons l'orientation estimée avec une erreur de $+10^\circ$ et -10° . La position estimée est initialisée avec différentes valeurs comprises dans une région carrée de 20 cm de côté centrée autour de la position réelle simulée.

Chacune des figures 3.35(a) à 3.38(b) donne une représentation de l'évolution de l'erreur après ajustement en fonction de l'erreur d'estimation sur la position pour une erreur initiale sur l'orientation connue et fixée. Sur les axes X_{us} et Y_{us} des figures sont représentées les coordonnées de la position estimée. Elles sont centrées autour de la position réelle. L'axe ε_θ représente la valeur absolue de l'erreur en orientation après ajustement.

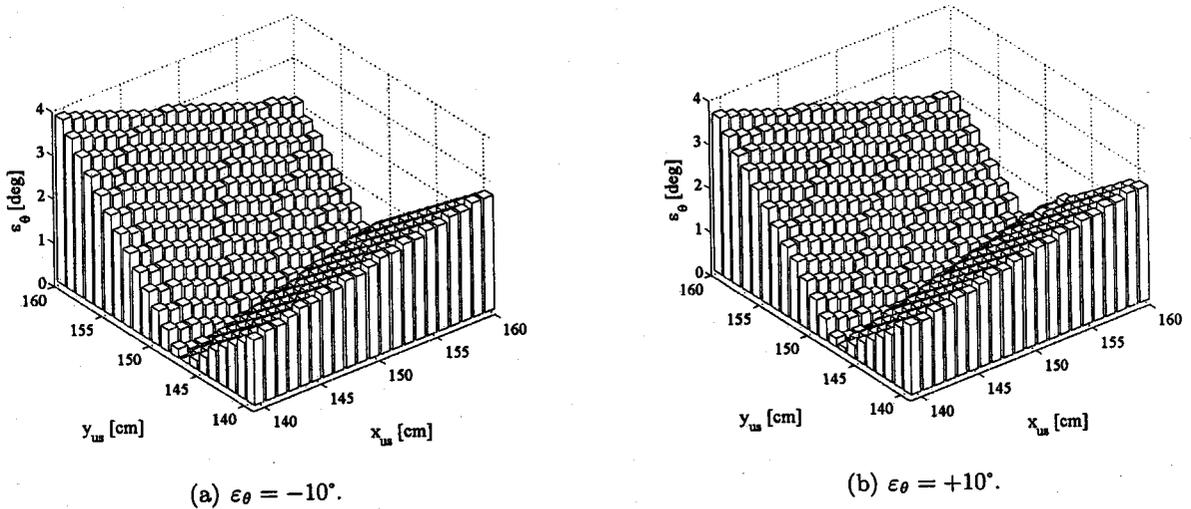


Figure 3.35 – Simulation 00 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.

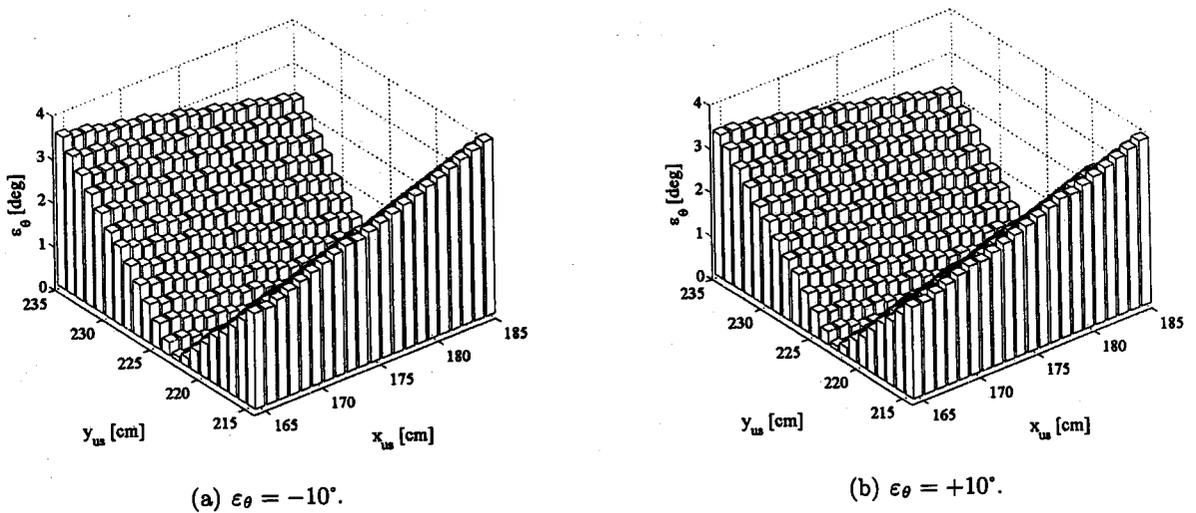


Figure 3.36 – Simulation 01 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.

En général nous retrouvons l'allure en forme de "V" de l'évolution de l'erreur, telle que cela avait été calculé au paragraphe 3.8. L'axe de l'arête basse du "V" est ici moins net que pour les courbes théoriques car nous considérons plusieurs verticales contre une seule par le calcul (§3.8). Dans la plupart des cas l'erreur ne dépasse le seuil des trois degrés qu'au voisinage des erreurs en position de l'ordre de 10 cm .

Mais tant que l'erreur en position est inférieure à 5 cm, l'erreur en orientation après ajustement n'est pas supérieure à 3°, ce qui est en accord avec les objectifs fixés pour le maintien d'une estimation correcte de la localisation. De plus, quelque soit l'erreur initiale en orientation pour une configuration réelle donnée, le résultat est quasiment le même ; les courbes de répartition de l'erreur en orientation après ajustement sont similaires (courbes (a) et (b) d'un même essai). La différence d'apparence entre les courbes 3.37 (a) et (b) et 3.38 (a) et (b) est due à la présence de valeurs élevées de ε_θ pour les courbes 3.37(b) et 3.38(b) qui nous ont amenés à adopter des échelles différentes.

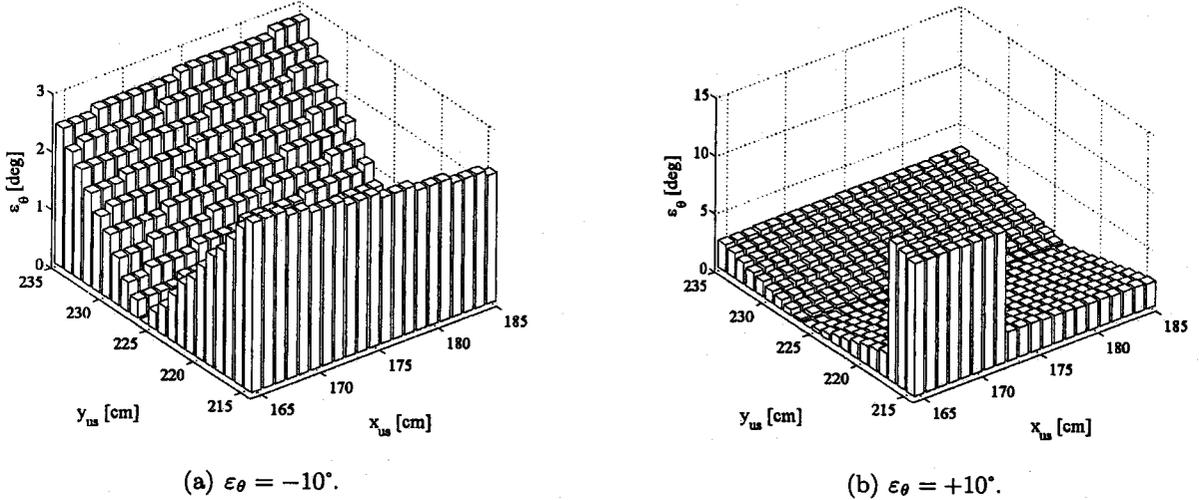


Figure 3.37 – Simulation 02 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.

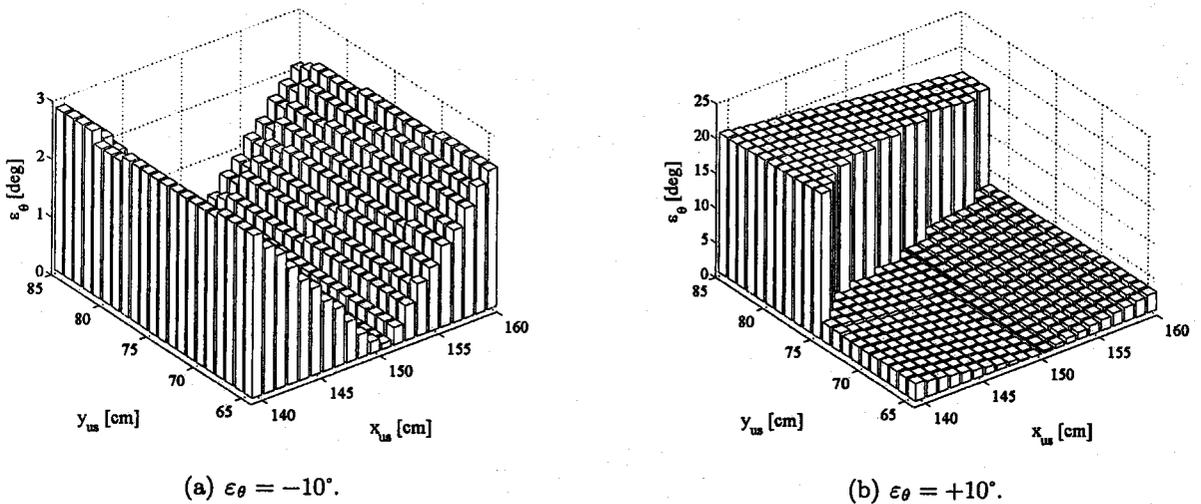


Figure 3.38 – Simulation 03 : évolution de l'erreur d'ajustement en fonction de l'erreur en position.

Les erreurs importantes visibles sur la figure 3.37(b) sont situées en marge de la zone d'initialisation de la position estimée. Dans cette situation, une des verticales de l'environnement correspondant à un des contours de l'image n'est plus sélectionnée, rendant alors impossible la détermination de l'appariement correct. En effet, en raison de la nature de l'arbre dont la profondeur est indexée par les contours de l'image, le long d'un chemin, le contour ne possédant pas de correspondantes dans la liste des verticales sera associé à un élément quelconque ce qui provoquera le début des mauvaises mises en correspondance. Nous nommons ce type de contour un "contour orphelin". Nous proposerons par la suite une méthode permettant de prendre en compte la présence de ce type de contour. Pour la courbe 3.38(b) nous retrouvons

le même problème des deux contours presque confondus que nous avons détaillé au paragraphe 3.9.4. Il ne se produit que lorsque ε_θ est positif et induit l'impossibilité d'ajuster l'orientation. Il est la cause des nombreuses valeurs de $\varepsilon_\theta > 15^\circ$.

L'allure des courbes montre que conformément aux formules (§3.8) lorsque $\Delta x \gg \Delta y$ (essais 00 et 02) l'erreur en y a plus d'influence et lorsque $\Delta y \gg \Delta x$ c'est l'erreur en x qui domine (essai 03). Dans un essai tel que le cas 01 où les conditions sont moins nettes l'allure de la courbe montre moins de prédominance d'un axe sur l'autre.

3.9.6 Synthèse

Cette étude en simulation de la méthode d'ajustement en orientation par vision monoculaire donne des résultats encourageants. D'une part l'ajustement en orientation permet de compenser des erreurs de l'ordre de celles fournies par la méthode employant les capteurs à ultrasons, d'autre part elle admet une erreur sur l'estimation de la position.

La suite de ce chapitre va être consacrée à l'étude de cas réels et la présentation d'une solution au problème posé par les contours orphelins.

3.10 Ajustement à partir d'images réelles

Dans cette partie nous allons évaluer le comportement de la méthode d'ajustement en orientation à partir d'essais en conditions réelles. Les acquisitions que nous allons utiliser ont été enregistrées pour les mêmes positions que celles considérées au paragraphe 2.8, pour les essais de la méthode de localisation à partir de capteurs à ultrasons du VAHM. La figure 3.39 donne une vue de dessus de l'environnement et des 15 positions successives du véhicule, les cercles représentent les positions des verticales modélisées pour l'ajustement en orientation. L'environnement d'essai est composé de panneaux de bois de 1 m ou de 0.5 m, ce qui explique la régularité de l'espacement des verticales et leur nombre important le long d'un même mur. L'évaluation de la méthode d'ajustement est réalisée indépendamment de la méthode de localisation en cours de mouvement par capteur à ultrasons. Nous initialiserons manuellement la position et l'orientation estimée à fournir à la méthode d'ajustement. Nous allons, comme pour l'étude en simulation, séparer cette étude en trois étapes. Dans une première étape, nous réalisons quelques essais pour différentes initialisations et différentes configurations réelles. Une seconde étape sera destinée à estimer la plage de variation de l'orientation initiale admissible autour de l'orientation réelle. Enfin dans une dernière étape, nous évaluerons l'influence d'une erreur en position sur l'ajustement en orientation.

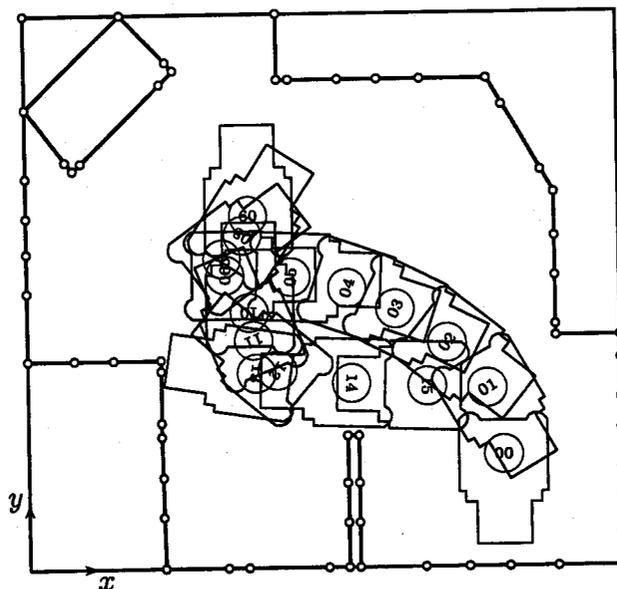


Figure 3.39 – Position des segments verticaux et trajectoire du fauteuil dans l'environnement de test.

3.10.1 Etudes des résultats obtenus pour différentes configurations

Dans cette première partie nous présentons des essais réalisés pour lesquels nous avons choisi comme position estimée la position réelle du véhicule à laquelle nous avons ajouté une erreur quelconque d'environ 5 cm et comme orientation estimée l'orientation réelle perturbée avec une erreur maximale de 10°. Nous voulons durant cette première évaluation nous rendre compte du comportement de l'algorithme lors de l'utilisation d'images réelles, c'est-à-dire vérifier que la méthode d'ajustement permet d'atteindre une erreur en orientation inférieure à 3°.

Le tableau 3.5 résume les différents essais réalisés. Les colonnes 2 à 4 donnent la localisation réelle du fauteuil, et les colonnes 5 à 7 donnent la localisation estimée simulée du fauteuil (normalement fournie par la méthode de correction par capteur à ultrasons). La colonne 8 donne la valeur de l'erreur en orientation après ajustement. Enfin les colonnes 9 à 12 donnent respectivement le nombre de verticales sélectionnées, le nombre de contours détectés, le nombre de nœuds contenus dans l'arbre élagué et dans l'arbre total. Dans la plupart des cas l'erreur d'orientation après correction est inférieure à 3°. Il est à noter que les essais 01, 03 et 07 ne donnent pas le bon appariement car un contour orphelin, provoqué par une texture, est présent dans l'image. Dans le cas 01 le contour parasite proche d'un contour réel apparaît "au milieu" du chemin de l'arbre, c'est à dire qu'il y a d'autres contours à associer lorsqu'on parcourt le chemin du nœud racine à un nœud terminal et il perturbe d'autres mises en correspondance (dans ce cas 4 contours réels). Dans les cas 03 et 07 la présence des contours orphelins (parasites) ne perturbent la mise en correspondance que d'un seul contour car ils se présentent à un niveau plus bas dans l'arbre. Ainsi l'ajustement en orientation est moins altéré. De plus le contour étant très proche d'une vraie caractéristique l'appariement n'est que faiblement perturbé.

essai	Localisation Réelle			Localisation Estimée			Ajt. $ \varepsilon_\theta $ [°]	nombre			
	x [cm]	y [cm]	θ [°]	x [cm]	y [cm]	θ [°]		seg.	ctr.	nbr. nœuds	
										elg.	ttl.
00	359	85	0	354	90	8	1.6	8	5	219	8801
01	347.5	135.5	30	343	139	23	4.5	7	7	128	13700
02	318	170	51	325	165	58	0	8	5	219	8801
03	279	196	63	288	186	55	0.4	8	6	247	28961
04	242.5	212	71	250	217	80	1.8	11	6	1489	397112
05	200	221	87	195	210	97	6.8	7	4	99	1100
06	150	222.5	90	160	218	100	1.8	7	3	64	260
07	148	235	126	142	240	120	1	5	5	32	326
08	164	252.5	147	170	260	137	3	9	3	130	586
09	168.5	267.5	180	160	260	190	2	11	3	232	1112
10	168.5	194	180	1 seul contour dans l'image							
11	171.5	173	202	180	180	212	0.7	11	5	1024	64472
12	189	150	230	180	160	220	0.8	11	6	1486	397112
13	173	148	261	180	140	270	3	10	3	176	821
14	245	140	268	240	148	258	3	9	6	466	79210
15	302.5	137.5	268	310	130	258	3	7	4	99	1100

Tableau 3.5 – Résultats des ajustements réalisés.

Par contre, le cas 05 ne possède pas de contours perturbateurs comme nous pouvons le voir sur la figure 3.40(a). Mais compte tenu de l'erreur de position et d'orientation le contour situé à l'extrémité droite devient orphelin (figure 3.40(b)). Il correspond à la verticale absente de la figure 3.40(a). Le contour sera donc associé à une autre verticale de l'image perturbant l'appariement des données. Nous rencontrons ici le même problème que celui évoqué en simulation : lorsque l'erreur sur la localisation est telle que la phase de sélection des verticales du modèle ne permet plus de considérer toutes les verticales qui se sont projetées dans le plan image. Nous verrons ultérieurement un moyen de prendre en compte ce type de

problème

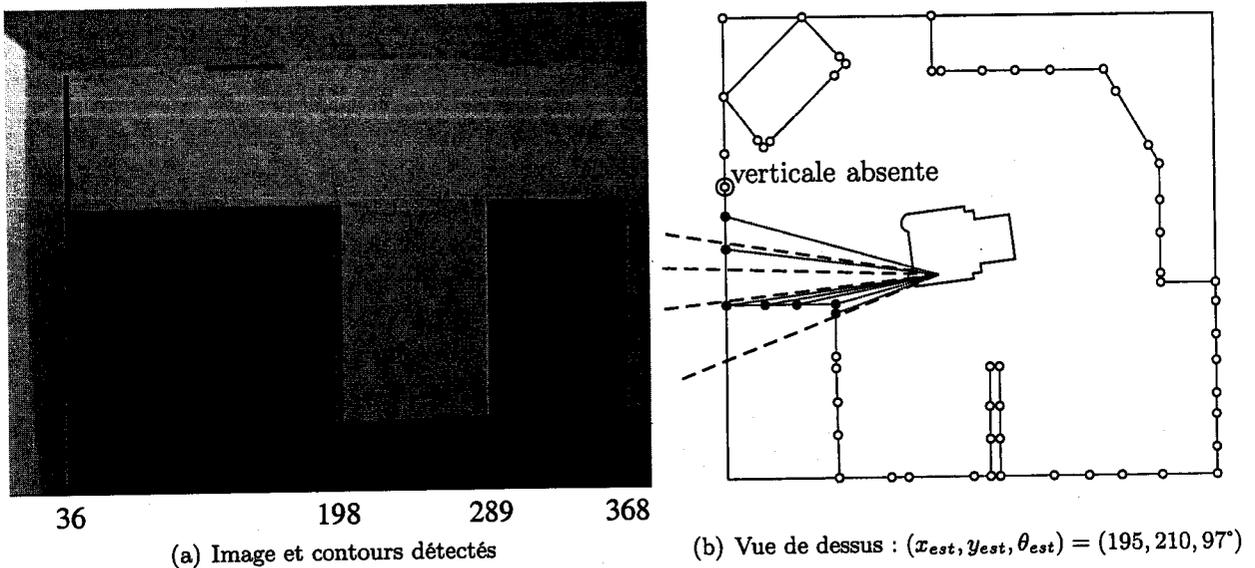


Figure 3.40 – Résultat de ajustement pour le cas 05 en conditions réelles.

Nous remarquons également l'avantage de l'utilisation d'un arbre élagué qui permet d'obtenir le même résultat que l'arbre total mais pour une complexité moindre. L'exécution de la méthode d'ajustement utilisant un arbre élagué sur un ordinateur architecturé sur un PIII 1GHz avec 512 Mo de RAM nécessite un temps de calcul moyen de l'ordre de 400 ms.

3.10.2 Estimation de la plage d'erreur admissible

Dans ce paragraphe nous allons reprendre les positions estimées utilisées précédemment et faire varier l'orientation estimée autour de l'orientation réelle jusqu'à atteindre les limites de la méthode d'ajustement pour lesquelles l'erreur sur l'orientation ajustée n'excède pas 3°. Les résultats résumés dans le tableau 3.6 montrent qu'une erreur initiale de $\pm 10^\circ$ sur l'orientation estimée est admissible pour la plupart des positions d'essais. Précédemment nous avons noté que dans le cas 05 l'algorithme présentait des difficultés à retrouver l'orientation correcte du fauteuil en raison d'une erreur en position importante. L'évaluation de sa plage d'erreur n'a pas permis d'obtenir une erreur après ajustement inférieure à 3° car rapidement les contours situés aux extrémités droite et gauche (respectivement 368 et 36 - figure 3.40(a)) n'ont plus de correspondants dans la liste des verticales sélectionnées. Les cas où il y avait un contour orphelin permanent dans l'image (contours provoqués par une perturbation dans l'image : texture dans l'image, ombre, etc.) présentent une plage de variation réduite. Pour le cas 01, la plage est inexistante, néanmoins l'erreur reste proche de 4.5°. Bien que possédant un contours orphelin, le cas 03 donne tout de même un résultat satisfaisant $\varepsilon_\theta \in [-9, 23]$, ceci est dû au fait qu'il s'est associé à un contour image proche de lui.

Sur le reste des positions de test, malgré une erreur non négligeable sur la position estimée, l'algorithme d'ajustement permet la correction d'une erreur absolue en orientation qui dépasse les 10°, atteignant même dans certains cas la valeur de 20°. Ces compensations d'erreurs importantes sont possibles lorsque les verticales observées sont situées loin de la caméra.

L'apparition de contours orphelins est une des causes d'échec de la méthode. L'étude du cas 11 met en évidence ce problème. La figure 3.41 représente la configuration du fauteuil autour de la position estimée (180, 180, 219°). Sur cette figure sont marquées les verticales V_1 à V_5 qui correspondent respectivement aux contours 276 à 24 de la figure 3.42. Pour cette localisation estimée, les contours de l'image ont tous leurs correspondants dans la liste des verticales sélectionnées. L'algorithme peut donc déterminer le bon appariement et ajuster correctement l'orientation estimée.

En augmentant l'erreur en orientation et en utilisant comme localisation estimée (180, 180, 220°), la verticale V_1 n'est plus sélectionnée (figure 3.43). L'algorithme ne peut plus trouver le bon appariement et

la qualité de l'ajustement en orientation est de moins bonne qualité car le contour 276 est devenu orphelin. Dans ce cas particulier il suffirait d'augmenter ρ , mais ce problème survenant dans d'autres situations, nous avons été amené à le considérer. Il apparaît donc nécessaire de mettre en œuvre une démarche pour prendre en compte les contours orphelins. Cela permettra d'augmenter les capacités d'ajustement de la méthode.

Une idée suggérée dans [Gri87] consiste à utiliser des nœuds nuls qui permettent d'éviter d'associer des mesures perturbées avec les composantes d'un modèle. Le paragraphe 3.11 présentera l'utilisation que nous en avons faite pour tenir compte des contours orphelins présents dans une image.

Lors de la phase d'estimation de la plage de variation nous avons noté que l'erreur négative maximale que pouvait corriger l'algorithme était de -11° (au delà de cette valeur l'ajustement en orientation ne garantit plus une erreur d'estimation inférieure à 3°). Or, en augmentant l'erreur d'estimation avant ajustement à plus de -11° , nous nous apercevons que les verticales V_1 à V_4 sont toutes sélectionnées et l'appariement retenu est également le bon appariement. Cette limitation est due à l'erreur en position. Le paragraphe suivant propose donc une étude de l'influence de l'erreur en position sur le calcul de l'ajustement.

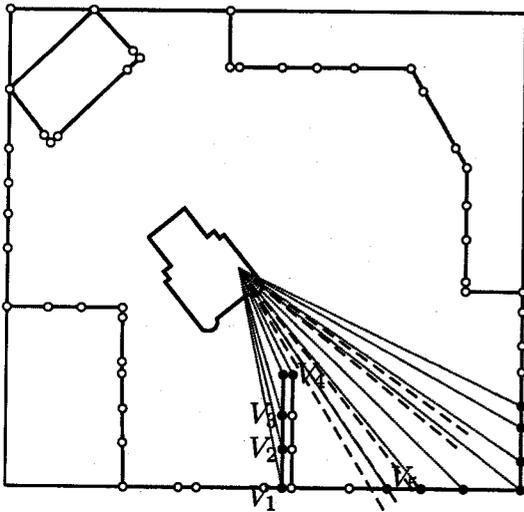


Figure 3.41 – Vue de dessus de la position 11 du véhicule avant ajustement de l'orientation.

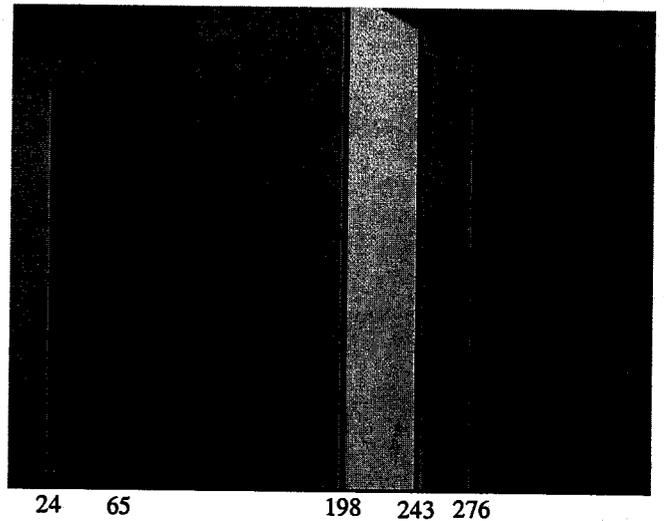


Figure 3.42 – Image acquise par la caméra avec mise en évidence des contours.

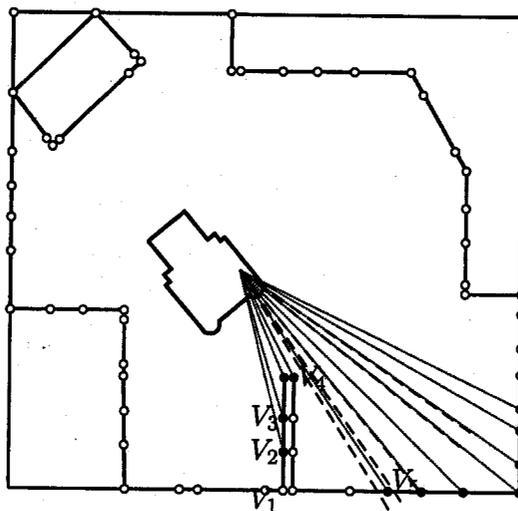


Figure 3.43 – Position 11 du véhicule pour une erreur en orientation de $+18^\circ$.

essai	Localisation Réelle			Position Estimée		$\varepsilon_{\theta_{min}}$ [deg]	$\varepsilon_{\theta_{max}}$ [deg]
	x_{re}	y_{re}	θ_{re}	x_{us}	y_{us}		
00	359	85	0	354	90	-14	11
01	347.5	135.5	30	343	139	-	-
02	318	170	51	325	165	-14	16
03	279	196	63	288	186	-9	23
04	242.5	212	71	250	217	-11	16
05	200	221	87	195	210	-	-
06	150	222.5	90	160	218	-14	11
07	148	235	126	142	240	-10	1
08	164	252.5	147	170	260	-12	14
09	168.5	267.5	180	160	260	-10	19
10	168.5	194	180	1 seul contour dans l'image			
11	171.5	173	202	180	180	-11	17
12	189	150	230	180	160	-12	11
13	173	148	261	180	140	-10	-10
14	245	140	268	240	148	-20	17
15	302.5	137.5	268	310	130	-10	18

Tableau 3.6 – Evaluation de la plage de variation de l'erreur admissible sur l'orientation estimée avant l'ajustement par vision monoculaire en conditions réelles.

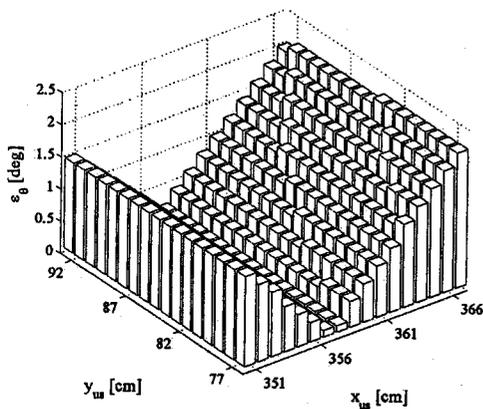
3.10.3 Influence d'une erreur en position

Dans ce paragraphe nous considérons l'influence d'une erreur en position sur la détermination de l'ajustement en orientation du véhicule. De façon similaire à l'étude en simulation, nous faisons varier la position estimée du robot dans un carré de 20 cm de côté centré autour de la position relevée au sol. Le pas de variation des coordonnées du véhicule est de 1 cm. L'axe vertical donne l'erreur absolue en orientation ε_{θ} après ajustement.

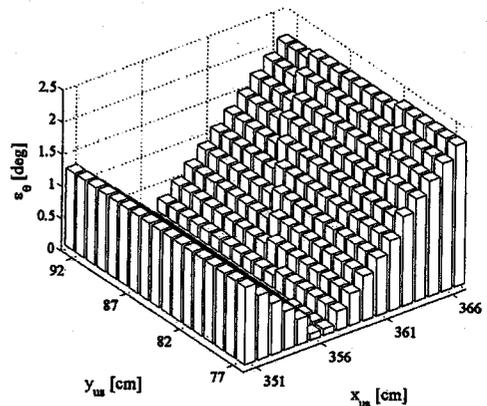
Lorsque les verticales observées sont situées suffisamment loin, de l'ordre de 1 m, la méthode donne des résultats similaires quelle que soit l'orientation initiale utilisée. Les figures 3.44(a) et 3.44(b) montrent respectivement les résultats obtenus pour le cas réel 00 avec $\theta_{est} = -10^\circ$ et $\theta_{est} = +10^\circ$ pour une plage de variation des erreurs de ± 8 cm autour de la position réelle car pour $\varepsilon_x < -8$ cm et $\varepsilon_y > +8$ cm un contour de l'image ne possède plus de correspondant dans le modèle et une erreur d'ajustement perturbe la comparaison visuelle des données. Lors de ces essais l'erreur en orientation après ajustement n'excède pas 3°. La répartition de l'erreur est bien en forme de "V". De plus, compte tenu de la configuration du véhicule dans l'environnement, les abscisses relatives des verticales par rapport à la caméra sont plus faibles que les ordonnées relatives, nous relevons alors dans ce cas une faible sensibilité aux erreurs sur l'estimation de l'ordonnée du véhicule. Lors des test pour les positions estimées des cas réels 02, 04, 13, 14 et 15 (voir annexe F) nous avons obtenus des résultats similaires à ceux du cas 00. Pour le cas 03 nous ne retrouvons pas la forme de "V" lorsque l'erreur initiale en orientation est de -10° , ce qui est justifié par le résultat de la plage de variation sur l'erreur initiale évaluée au paragraphe précédent qui était de $[-9, \dots, 23]$. En réitérant les essais avec une erreur initiale en orientation de -9° alors nous retrouvons la répartition en forme de "V". Pour les essais 05, 06, 09, 11, 12 la zone de fonctionnement n'est pas aussi étendue mais reste de l'ordre des ± 5 cm autour de la position réelle. Au delà de cette zone des contours orphelins apparaissent plus ou moins rapidement suivant la configuration. La position 10 ne possédant qu'un seul contour, elle ne peut pas fournir un ajustement en orientation. Les positions 01, 03 et 07 possèdent au moins un contour orphelin permanent (parasite). Dans ces cas, la répartition de l'erreur est imprévisible.

En comparant les résultats des essais réalisés autour des positions 14 et 15 qui sont des configurations similaires (la position 15 est plus proche du mur de droite que la position 14 (figure 2.25)), nous retrouvons la constatation faite lors de l'étude théorique de l'influence d'une erreur en position sur le calcul de l'ajustement qui montre que plus les distances entre les verticales observées et la caméra sont importantes

plus l'ajustement en orientation admet une erreur en position importante. Les verticales observées en 14 et 15 étaient pratiquement toutes les mêmes. La différence entre ces deux essais est la distance d'observation plus petite pour la position 15, ce qui se traduit par une évolution plus rapide de l'erreur d'ajustement en fonction de l'erreur en position.



(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$



(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure 3.44 – Essai 00 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.

3.10.4 Résumé

Les différents tests réalisés à partir d'images réelles, donnent des résultats satisfaisants. L'objectif des 3° d'erreur en orientation est respecté tant que les contours de l'image possèdent leur correspondant dans la liste des segments verticaux sélectionnés du modèle.

Afin d'améliorer les performances de la méthode, il est nécessaire de résoudre le problème des contours orphelins qui sont la principale cause des cas d'échecs. La partie suivante propose une solution à ce problème.

3.11 L'arbre d'interprétation à nœuds nuls

3.11.1 Problème lié à la structure de l'arbre

La présence d'un contour parasite dans l'image est un problème car notre arbre est indexé en profondeur par l'ordre d'apparition des contours dans l'image. Pour évoluer le long d'un chemin il faut obligatoirement affecter une verticale au contour courant. S'il ne correspond à rien il sera associé à une verticale quelconque ce qui perturbera toutes les associations suivantes du chemin.

Ce problème d'appariement peut survenir lorsqu'une ombre est projetée sur un mur, ou si une verticale n'a pas été sélectionnée mais est présente dans l'image (cas typique des contours situés à la limite du champ de vision de la caméra). Pour la seconde source de contours orphelins, nous pourrions augmenter l'angle ajouté à la demi-ouverture de la caméra afin qu'une zone de sélection des verticales du modèle plus grande soit utilisée. L'inconvénient est que plus on sélectionne de de verticales plus la taille de l'arbre augmente, et cela ne fait que repousser le problème à une erreur d'estimation de la localisation plus grande.

3.11.2 Solution proposée

Notre approche prévoit la création de nœuds particuliers, dit "nœuds nuls", qui permettent d'associer un contour donné de l'image à une "verticale virtuelle" de l'environnement. Cette verticale n'existe pas réellement, elle est uniquement ajoutée au modèle afin d'être associée avec des contours orphelins pour qu'ils n'empêchent pas la construction correcte de l'arbre. Puis l'arbre est construit de la même manière que précédemment avec seulement une différence en ce qui concerne le test de cohérence. Si nous créons un

nœud nul le test de cohérence n'a pas lieu et, si le nœud courant pour lequel nous créons des descendants est nul, nous remontons au nœud parent afin de tester la cohérence des associations le long du chemin. De cette manière, le nœud nul permet de neutraliser le contour orphelin d'un niveau donné de l'arbre. Et lors de l'évaluation d'un chemin, un contour parasite associé à une verticale virtuelle ne sera pas utilisé pour le calcul de l'ajustement.

La mise en œuvre permet également de prendre en compte le cas où le nombre de verticales sélectionnées est inférieur au nombre de contours détectés. Il suffit alors d'ajouter le complément de verticales virtuelles afin que les deux ensembles soient de même taille. Ainsi nous garantissons que le nombre de contours est toujours au moins inférieur ou égal au nombre de segments verticaux sélectionnés.

3.11.3 Evaluation avec des images réelles

3.11.3.1 Evaluation des ajustements pour des erreurs initiales nulles

Afin d'évaluer l'apport de l'utilisation des nœuds nuls nous reprenons les mêmes conditions d'essais que celles utilisées au paragraphe 3.10.1 lors de l'évaluation de la méthode d'ajustement. Sur l'ensemble des positions donnant le bon ajustement en orientation, l'utilisation d'un nœud nul n'altère pas la qualité de l'ajustement.

essai	Localisation Réelle			Localisation Estimée			Ajt. ss nœuds nuls	Ajt. avec nœuds nuls
	x [cm]	y [cm]	θ [°]	x [cm]	y [cm]	θ [°]	ε_θ [°]	ε_θ [°]
00	359	85	0	354	90	8	1.6	1.6
01	347.5	135.5	30	343	139	23	4.5	1.6
02	318	170	51	325	165	58	0	0.2
03	279	196	63	288	186	55	0.4	1.2
04	242.5	212	71	250	217	80	1.8	1.4
05	200	221	87	195	210	97	6.8	0.7
06	150	222.5	90	160	218	100	1.8	1.8
07	148	235	126	142	240	120	1	1
08	164	252.5	147	170	260	137	3	3
09	168.5	267.5	180	160	260	190	2	2
10	168.5	194	180	1 seul contour dans l'image				
11	171.5	173	202	180	180	212	0.7	0.5
12	189	150	230	180	160	220	0.8	0.5
13	173	148	261	180	140	270	3	3
14	245	140	268	240	148	258	3	3
15	302.5	137.5	268	310	130	258	3	3

Tableau 3.7 – Résultats des ajustements réalisés avec l'utilisation d'un nœud nul.

Pour les positions 01 et 03 où il y avait au moins un contour orphelin permanent dans l'image l'utilisation d'un nœud nul permet de retrouver une erreur après ajustement inférieure à 3°. Le cas 05, où l'algorithme n'arrivait pas à ajuster l'erreur en orientation car l'erreur de localisation était telle que la sélection de toutes les verticales utiles du modèle n'était pas possible, est aussi résolu par l'utilisation d'un nœud nul. Pour le cas 07, où un contour orphelin permanent ne gênait pas trop l'ajustement car le contour ne perturbait pas les autres associations du chemin, nous notons uniquement une amélioration de l'écart-type mesuré le long du chemin de 5.2° pour l'exécution sans nœud nul à 1.4° actuellement.

Le temps de traitement lors de l'utilisation d'un nœud nul est de l'ordre de 600 ms et passe à 750 ms pour l'utilisation de deux nœuds nuls.

3.11.3.2 Evaluation de la plage d'erreur initiale admissible

Nous avons vérifié au paragraphe précédent que le fait d'utiliser un arbre comportant un nœud nul permet de s'affranchir du problème des contours orphelins. Nous allons voir à présent si cette modification de l'algorithme autorise une plage de variation plus large de θ_{est} tout en assurant un ajustement correct. Nous avons repris les essais réalisés lors de la précédente évaluation de l'algorithme. Le tableau 3.8 présente les résultats obtenus avec un nœud nul dans l'arbre en comparaison de ceux de l'algorithme initial. Nous constatons un accroissement des capacités de la méthode d'ajustement. Les exemples donnés dans le tableau 3.8 ont quasiment tous une plage d'erreur admissible au minimum égale à $\pm 10^\circ$ autour de l'orientation réelle du véhicule à l'exception du cas 05 qui a une plage de variation de l'erreur inférieure à $\pm 10^\circ$ en raison de l'erreur en position importante. Nous constatons que pour le cas 07 la plage de variation sur l'erreur d'estimation de l'orientation atteint maintenant les $\pm 10^\circ$.

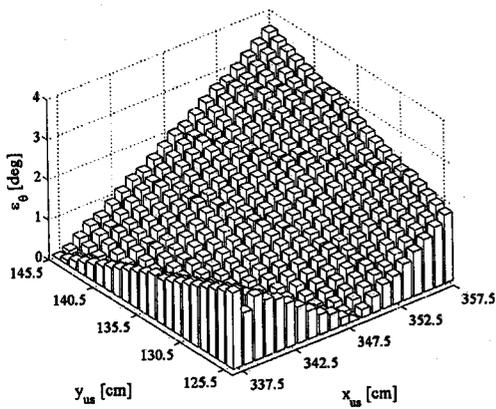
essai	Localisation Réelle			Position Estimée		Sans nœuds nuls		Avec nœuds nuls	
	x_{re}	y_{re}	θ_{re}	x_{us}	y_{us}	$\epsilon_{\theta_{min}}$	$\epsilon_{\theta_{max}}$	$\epsilon_{\theta_{min}}$	$\epsilon_{\theta_{max}}$
00	359	85	0	354	90	-14	11	-20	21
01	347.5	135.5	30	343	139	-	-	-14	9
02	318	170	51	325	165	-14	16	-17	19
03	279	196	63	288	186	-9	23	-15	23
04	242.5	212	71	250	217	-11	16	-17	16
05	200	221	87	195	210	-	-	-2	10
06	150	222.5	90	160	218	-14	11	-14	11
07	148	235	126	142	240	-10	1	-10	10
08	164	252.5	147	170	260	-12	14	-12	14
09	168.5	267.5	180	160	260	-10	19	-10	19
10	168.5	194	180	1 seul contour dans l'image					
11	171.5	173	202	180	180	-11	17	-12	18
12	189	150	230	180	160	-12	11	-12	11
13	173	148	261	180	140	-10	10	-10	10
14	245	140	268	240	148	-20	17	-20	17
15	302.5	137.5	268	310	130	-10	18	-10	18

Tableau 3.8 – Evaluation de la plage de variation de l'erreur d'orientation avant correction - arbre élagué et utilisation de nœuds nuls.

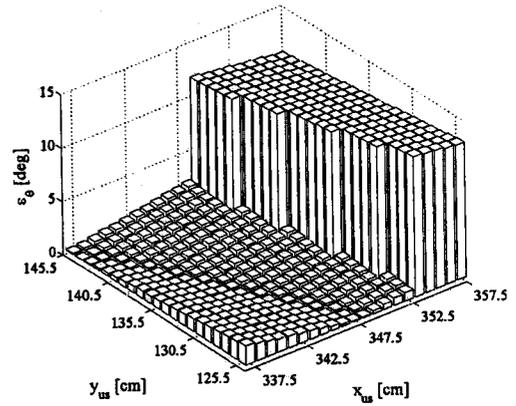
Les bornes de la plage limite n'ont pas évolué pour le cas 13 car dans cette configuration deux des trois contours détectés dans l'image sont très proches et deviennent orphelins à partir de la même valeur de l'orientation estimée. Dans ce cas il est inutile d'utiliser 2 nœuds nuls pour masquer ces deux contours, car il ne reste plus qu'un contour utile dans l'image ce qui n'est pas suffisant pour déterminer de façon sûre l'ajustement en orientation.

3.11.3.3 Influence d'une erreur en position sur l'ajustement

Dans ce paragraphe, nous reprenons les cas 01, 03 et 07 pour lesquels l'étude de l'influence de l'erreur de position était rendue impossible en raison de la présence d'un contour orphelin permanent dans l'image. Nous donnons l'évolution de l'erreur d'ajustement en fonction de la position estimée pour une erreur initiale de -10° et $+10^\circ$ (figures 3.45(a) à 3.47). Nous pouvons constater que l'ajout de nœuds nuls dans l'arbre d'appariement permet de retrouver les courbes caractéristiques de ce type d'évaluation. Tant que nous nous situons dans une région de l'ordre de 5 cm autour de la position réelle, l'erreur en orientation après ajustement n'excède pas les 3°. Mais au delà de cette limite l'ajustement n'est pas toujours assuré.

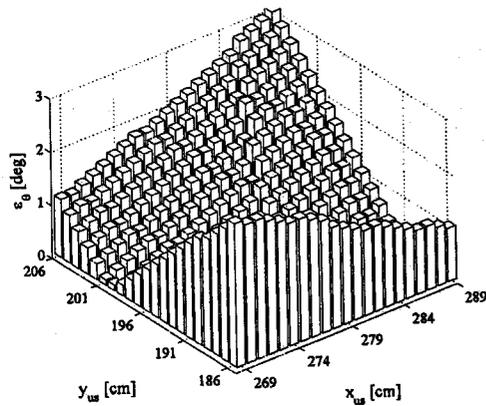


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

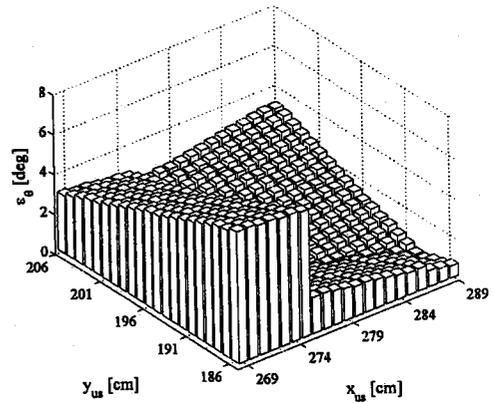


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure 3.45 – Essai 01 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée avec l'utilisation d'un nœud nul.

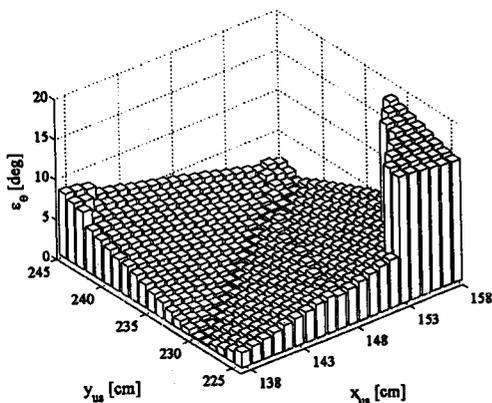


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

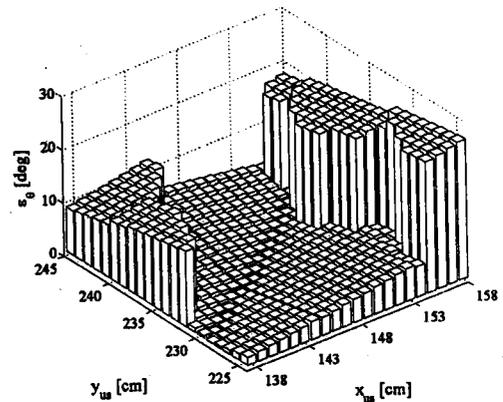


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure 3.46 – Essai 03 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée avec l'utilisation d'un nœud nul.



(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$



(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure 3.47 – Essai 07 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée avec l'utilisation d'un nœud nul.

3.11.4 Résumé

La solution retenue pour s'affranchir de la présence de contours orphelins dans l'image permet d'accroître les performances de la méthode d'ajustement. Les cas pour lesquels la présence d'un contour parasite rendait la méthode inopérante peuvent à présent être traités et nous obtenons une estimation correcte du cap du véhicule. De plus les configurations qui permettaient déjà un ajustement correct de l'orientation voient leur plage de variation de l'erreur admissible en orientation augmenter.

3.12 Conclusion sur l'ajustement en orientation

La méthode présentée dans ce chapitre avait pour objectif d'ajuster l'orientation estimée du véhicule connaissant la position de celui-ci dans son environnement. La position et l'orientation sont fournies par la méthode de correction de la localisation odométrique par capteurs à ultrasons.

Au cours de ce chapitre nous avons présenté comment étaient modélisés les repères naturels choisis pour l'ajustement de l'orientation ; nous avons choisi de modéliser les bords verticaux à l'aide de segments verticaux. Nous avons fixé également la contrainte sur le positionnement de la caméra afin que bords verticaux soient perçus dans le plan image comme des contours verticaux. Nous avons ensuite présenté comment étaient extraits les contours de l'image par l'adaptation à nos besoins d'outils de traitement des images. Nous avons ensuite montré comment à partir d'une observation et de la connaissance de la position de la caméra nous pouvions retrouver son orientation. Nous avons poursuivi par une présentation de la méthode utilisée pour apparier les contours de l'image avec les segments du modèle. Cette méthode est une méthode combinatoire utilisant un arbre d'appariement. Le problème principal de ce type de méthode est l'explosion combinatoire lors de l'utilisation d'un nombre croissant de données. Nous avons contourné efficacement ce problème en utilisant deux critères :

- d'une part nous limitons le nombre de segment du modèle à une région de perception située devant le véhicule, définie à partir de l'orientation estimée initiale. Cette technique permet de limiter le nombre de données issues du modèle.
- l'utilisation d'un critère d'évaluation de la cohérence de l'appariement des données extraites de l'image avec celles issues du modèle. Ce critère est utilisé lors de la création de l'arbre.

L'évaluation de la méthode a conduit à des résultats satisfaisants même en présence d'une erreur en position compatible avec les objectifs fixés, c'est à dire une erreur absolue sur x et y inférieure à 5 cm . La plage admissible sur l'erreur d'estimation initiale de l'orientation va bien au delà des 10° pour atteindre une erreur finale inférieure à 3° , ce qui satisfait à la condition posée pour le maintien d'une localisation estimée correcte. Lors des essais, il est apparu que la présence de contours orphelins dans l'image conduisait à une limitation des performances de l'algorithme. Ce problème a été résolu en intégrant les nœuds nuls qui permettent d'inhiber l'influence de tels contours. Nous avons supposé ici que ces contours orphelins étaient causés par des parasites dans l'image. Il est aussi possible de rencontrer dans l'environnement des objets non modélisés (chaises, meubles d'appoint, etc.) susceptibles d'introduire des contours supplémentaires. Les verticales virtuelles peuvent être un moyen de les prendre en compte.

Les temps de calcul de la méthode ont également été évalués. Ils sont de l'ordre de 600 ms avec l'utilisation de nœuds nuls. La méthode proposée vient donc bien compléter la méthode de correction par capteurs à ultrasons.

Conclusion et Perspectives

Conclusion

Le travail présenté dans cette thèse a trait au problème de la localisation d'un robot mobile évoluant dans un environnement d'intérieur structuré et connu. La méthode de localisation développée s'intègre au projet VAHM. Elle a pour but de rendre fiable l'estimation de la localisation du véhicule lors de ses déplacements autonomes.

Notre méthode de localisation utilise :

- Un odomètre qui permet d'estimer la position et l'orientation du véhicule lorsqu'il se déplace. Ce système est sujet à des erreurs, comme nous l'avons présenté aux chapitres 1 et 2. L'estimation fournie par ce système doit être corrigée régulièrement à l'aide de mesures prises par d'autres capteurs.
- Des capteurs à ultrasons disposés tout autour du robot déjà utilisés par d'autres modules d'assistance à la conduite, comme le suivi d'espace libre ou le suivi de mur. Ils sont utilisés pour construire une carte locale qui sera comparée à la carte connue de l'environnement afin de corriger en position et en orientation la localisation estimée fournie par l'odomètre.
- Un système de vision monoculaire destiné à ajuster l'orientation obtenue par la méthode de localisation par capteurs à ultrasons.

La première partie du travail effectué a consisté à améliorer la méthode initiale de localisation en cours de mouvement basée sur la correction de l'estimation odométrique à l'aide des données renvoyées par les capteurs à ultrasons. Pour cela, nous avons considéré les spécificités d'un capteur à ultrasons afin de valider la détection d'un obstacle en fonction de l'angle d'incidence de l'onde acoustique sur la surface de l'objet. Nous avons également pris en compte la détection de mesures improbables causées par le phénomène de réflexions multiples.

Les essais réalisés sur cette méthode ont permis d'obtenir de bons résultats concernant la position sous certaines conditions. Ces conditions de bon fonctionnement de la méthode sont celles pour lesquelles un nombre suffisant de capteurs à ultrasons perçoivent des surface réelles et que ces mesures soient réparties dans plusieurs directions. De plus si les axes acoustiques de ces capteurs sont orthogonaux aux segments qu'ils perçoivent, alors la précision du recalage est améliorée. Sur l'ensemble des résultats obtenus, lorsque le recalage peut s'exécuter nous atteignons dans 85% des cas une erreur de position inférieure à 5 cm dans chaque direction. Mais la faible directivité des capteurs à ultrasons ne permet pas un recalage en orientation satisfaisant au seuil fixé (3°).

Il est donc apparu nécessaire d'ajouter un capteur supplémentaire afin d'ajuster l'orientation fournie par la méthode utilisant les capteurs à ultrasons. Nous avons choisi la vision monoculaire car elle permet d'obtenir une mesure de l'angle sous lequel est vue chacune des balises naturelles de l'environnement plus précise que celle obtenue avec les capteurs à ultrasons.

Le premier problème que nous avons dû résoudre a été de choisir le type de caractéristiques à détecter dans l'environnement et comment les représenter dans le modèle. Nous avons choisi les bords verticaux des objets de l'environnement, tels que les bords d'une fenêtre ou les coins d'un meuble. Comme le plan image de la caméra est perpendiculaire au plan de déplacement du fauteuil, les projections des bords verticaux de l'environnement dans le plan image de la caméra sont des contours verticaux. Il est alors aisé de déterminer l'angle sous lequel sont vus ces bords en utilisant les positions de leurs contours dans le référentiel image. Nous avons mis en œuvre des outils de traitement d'image adaptés à notre situation.

Le second problème que nous avons à résoudre était de mettre en correspondance les verticales contenues dans le modèle avec les contours détectés dans l'image afin d'ajuster au mieux l'orientation du

véhicule. Nous avons décidé d'utiliser un arbre d'interprétation car il permet d'obtenir tous les appariements possibles des données. Afin de réduire la taille de l'arbre nous avons mis en œuvre deux critères. Le premier critère consiste à sélectionner une partie des verticales du modèle en fonction de l'estimation de la position et de l'orientation du véhicule. Le second critère se base sur la cohérence des appariements le long d'un chemin afin de ne pas "croiser" des correspondances. La méthode montre ses limites lorsque au moins un contour de l'image ne possède pas de correspondant dans la liste des segments verticaux sélectionnés dans le modèle. Pour cela nous avons proposé une solution basée sur l'utilisation de "nœuds nuls" qui permet d'ignorer le ou les contours n'ayant pas de bord correspondant.

Les essais réalisés à partir d'images réelles ont été concluants. Ils ont mis en évidence que la méthode permet de compenser des erreurs en orientation d'une amplitude supérieure à 10° , les ramenant à un ordre de grandeur de 3° et ce même lorsque la position estimée n'est pas totalement exacte. L'immunité de la méthode d'ajustement face à une erreur en position dépend de la proximité des observations. Plus ces observations seront loin de la caméra, plus la méthode pourra compenser une erreur en orientation importante et ce dans une région d'erreur en position plus importante.

Nous sommes conscients que dans certaines configurations l'ajustement ne pourra plus fonctionner. Mais compte tenu du contexte de l'application qui se base sur le partage de la tâche de navigation entre l'homme et le système, lorsque l'estimation de la localisation sera trop erronée, la navigation en mode autonome ne dirigera plus le fauteuil dans la bonne direction. Comme l'utilisateur reste le superviseur du système, il pourra prendre la décision d'arrêter le déplacement du véhicule ce qui enclenche automatiquement une procédure de localisation statique.

Perspectives

Lors des essais, nous avons pu noter un temps d'exécution moyen de 300ms pour la méthode de correction de la localisation odométrique par capteurs à ultrasons, et un temps d'exécution de l'ordre de 500ms pour la méthode d'ajustement de l'orientation par vision monoculaire. Au chapitre 2 nous avons estimé que pour une vitesse de 2km/h , le temps nécessaire au véhicule pour parcourir 50cm entre deux mises à jour de l'estimation odométrique est de 900ms . En laissant le traitement des données et le calcul des corrections à la charge d'un seul et même processeur, celui-ci ne disposera alors plus que de 100ms pour la réalisation d'autres tâches comme la commande des moteurs, la gestion de l'IHM, ou la détection d'obstacle. Dans la méthode d'ajustement de l'orientation par vision monoculaire, la charge majeure provient essentiellement du traitement de l'image qui prend approximativement 250ms à 300ms . Ce traitement pourrait être réalisé par un processeur dédié au traitement d'image, et ainsi libérer le processeur central pour la réalisation d'autres tâches. L'utilisation d'un système multiprocesseurs est également envisageable mais risque d'accroître le coût global du système ainsi que sa consommation d'énergie qui est un paramètre vital pour un système autonome.

Comme la plupart des premiers projets, visant à développer une assistance à la conduite de véhicule pour personnes handicapées initiés au début des années 90, le projet VAHM a cherché tout d'abord à adapter des outils développés initialement pour la robotique autonome au cas de l'assistance à la conduite d'un fauteuil roulant. Dans ce contexte la tâche de navigation était essentiellement confiée à la machine qui s'occupe de la planification et de l'exécution de chaque déplacement sans intervention de la personne. La nécessité d'une parfaite connaissance de la localisation est alors justifiée. L'objectif de cette thèse était donc de proposer un moyen d'estimation de la localisation fiable et rapide. La méthode de localisation développée est relativement fiable, son temps d'exécution devant être amélioré. Par ailleurs le travail effectué au cours de cette thèse nous a permis d'évaluer les performances des différents capteurs de perception ; les capteurs à ultrasons permettent d'obtenir une bonne précision en position par rapport aux obstacles et l'utilisation que nous faisons de la vision donne quant à elle une bonne précision en orientation.

La tendance actuelle en robotique mobile de réadaptation est de s'orienter vers une réelle coopération entre l'utilisateur et le fauteuil intelligent [Bou98]. Comme le stipule P.D. Nisbet dans [Nis02] : *"l'objectif le plus important est de développer des systèmes qui complètent, maximisent et augmentent les compétences de l'utilisateur et non qui les remplacent"*. On s'oriente donc vers le développement de "fauteuils intelligents" plus réactifs. Le projet VAHM n'est pas en reste puisqu'il développe une approche coopérative

entre l'utilisateur et la machine. L'utilisation du fauteuil intelligent en mode autonome dans ce contexte n'implique pas une fiabilité totale de la méthode de localisation : une intervention ponctuelle de la personne est admissible pour peu qu'elle soit ni trop fréquente ni trop complexe. Cet aspect du problème touchant à l'interaction humain-machine devra faire l'objet d'une étude spécifique.

Enfin l'hypothèse d'un univers totalement modélisé employée lors de cette thèse ne peut être considérée comme réaliste. Elle nous a toutefois permis d'évaluer les possibilités des capteurs utilisés pour la localisation et ainsi de déterminer leur domaine d'efficacité. Les capacités des capteurs ainsi définies pourront être employées dans d'autres contextes comme la navigation réactive ou l'assistance à la perception.

Cette page est laissée blanche intentionnellement

Annexe A

Algorithmes utilisés par la localisation dynamique

A.1 Version utilisant la distance point segment

```
ENTREES : Les points de la carte locale et les segments de la carte globale
SORTIES : Liste des appariements et le nombre d'appariements
POUR tous les points de la carte locale
| POUR tous les segments de la carte globale
| | Tester si le point courant est dans l'ellipse courante
| | SI OUI
| | | Vérifier si le point courant est déjà apparié
| | | SI NON
| | | | Apparier le point courant et le segment courant
| | | | Incrémenter le nombre d'appariement
| | | SINON
| | | | Calculer la distance dcc séparant le point courant du segment
| | | | courant
| | | | Calculer la distance dca séparant le point courant au segment auquel il
| | | | est déjà apparié
| | | | SI dcc < dca
| | | | | Supprimer l'appariement entre le point courant et le segment auquel
| | | | | il était apparié
| | | | | Apparier le point courant et le segment courant
| | | | FIN SI
| | | FIN SI
| | FIN SI
| FIN POUR
FIN POUR
```

Figure A.1 – Algorithme d'appariement avec test de la distance point/segment

A.2 Version utilisant la distance point segment et l'angle d'incidence

```
ENTREES : Les points de la carte locale et les segments de la carte globale
SORTIES : Liste des appariements et le nombre d'appariements
POUR tous les points de la carte locale
| POUR tous les segments de la carte globale
| | Tester si le point courant est dans l'ellipse courante
```

```

| | SI OUI
| | | Vérifier l'angle d'incidence
| | | SI -12 deg < Angle < +12 deg
| | | | Vérifier si le point courant est déjà apparié
| | | | SI NON
| | | | | Apparier le point courant et le segment courant
| | | | | Incrémenter le nombre d'appariement
| | | | SINON
| | | | | Calculer la distance dcc séparant le point courant du segment courant
| | | | | Calculer la distance dca séparant le point courant au segment auquel
| | | | | il est déjà apparié
| | | | | SI dcc < dca
| | | | | Supprimer l'appariement entre le point courant et le segment auquel
| | | | | il était apparié
| | | | | Apparier le point courant et le segment courant
| | | | | FIN SI
| | | | FIN SI
| | | FIN SI
| | FIN SI
| FIN POUR
FIN POUR

```

Figure A.2 – Algorithme d'appariement avec test de la distance point/segment et test de l'angle d'incidence

A.3 Version éliminant les appariements incohérents

Un appariement dit incohérent est un appariement associant un point du modèle local avec un segment du modèle global qui n'est physiquement pas perceptible compte tenu des positions du point et du segment dans l'environnement. C'est le cas lorsqu'un point du modèle local est susceptible d'être apparié avec un segment qui serait masqué par la présence d'un autre obstacle en ces deux derniers.

```

ENTREES : Les points de la carte locale et les segments de la carte globale
SORTIES : Liste des appariements et le nombre d'appariements
POUR tous les points de la carte locale
| POUR tous les segments de la carte globale
| | Tester si le point courant est dans l'ellipse courante
| | SI OUI
| | | Vérifier l'angle d'incidence
| | | IF -12 deg < Angle < +12 deg
| | | | Vérifier si le point courant est déjà apparié
| | | | SI NON
| | | | | Vérifier si le segment CI croise un segment du modèle
| | | | | SI NON
| | | | | Apparier le point courant et le segment courant
| | | | | Incrémenter le nombre d'appariement
| | | | | FIN SI
| | | | SINON
| | | | | Calculer la distance dcc séparant le point courant du segment courant
| | | | | Calculer la distance dca séparant le point courant au segment auquel
| | | | | il est déjà apparié
| | | | | SI dcc < dca
| | | | | Vérifier si le segment CI croise un segment du modèle

```

```
| | | | | SI NON
| | | | | Supprimer l'appariement entre le point courant et le segment
| | | | | auquel il était apparié
| | | | | Apparié le point courant et le segment courant
| | | | | FIN SI
| | | | | FIN SI
| | | | FIN SI
| | | FIN SI
| | FIN SI
| FIN POUR
FIN POUR
```

Figure A.3 – Algorithme d'appariement avec test de la distance point/segment et test de l'angle d'incidence et prise en compte des réflexions multiples

Dans l'algorithme A.3, CI est le segment qui part de la position du capteur à ultrasons définie dans le référentiel global et qui atteint le point du modèle local défini dans le même référentiel.

Cette page est laissée blanche intentionnellement

Annexe B

Intersection de deux segments

B.1 Position d'un point par rapport à une droite

Le problème est de déterminer la position d'un point A par rapport à une droite donnée repérée par deux de ses points B_1 et B_2 .

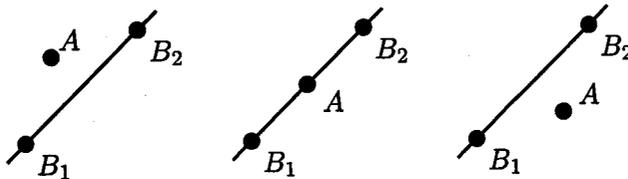


Figure B.1 – Position d'un point par rapport à une droite.

La méthode consiste à étudier le signe du déterminant de $B_1A \wedge B_1B_2$ comme critère de l'orientation de l'angle entre ces deux vecteurs.

La solution consiste donc à renvoyer le signe de :

$$\begin{vmatrix} x_A - x_{B_1} & x_{B_2} - x_{B_1} \\ y_A - y_{B_1} & y_{B_2} - y_{B_1} \end{vmatrix}$$

La fonction se résume à :

```
double signe( double a )
{
return ( ( a < 0.0 )? -1.0 : 1.0 );
}

double test_cote_du_segment_h( POINT a, POINT b1, POINT b2 )
{
double det;

det = ( ( a.x - b1.x ) * ( b2.y - b1.y ) ) -
      ( ( b2.x - b1.x ) * ( a.y - b1.y ) );

if( det != 0.0 )
    return signe( det );
else
return 0.0;
}
```

Remarque : Ce signe est fonction du fait que le système de coordonnées (x, y) est direct ou indirect.

B.2 Intersection de deux segments

Le problème est de déterminer si deux segments $[A_1, A_2]$ et $[B_1, B_2]$ ont une intersection vide ou non.

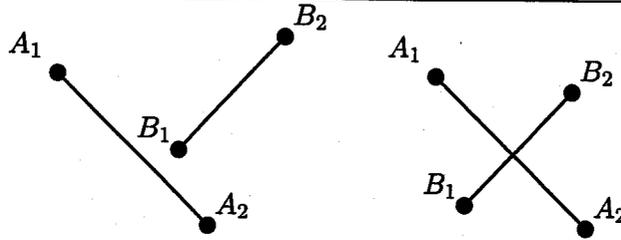


Figure B.2 - Intersection de deux segments.

La méthode consiste à vérifier que les points A_1 et A_2 sont de part et d'autre de la droite (B_1B_2) et que les points B_1 et B_2 sont de part et d'autre de la droite (A_1A_2) .

La solution est donnée par la fonction suivante :

```
double intersect( POINT a1, POINT a2, POINT b1, POINT b2 )
{
return MAX( test_cote_du_segment_h( a1, b1, b2 ) *
            test_cote_du_segment_h( a2, b1, b2 ),
            test_cote_du_segment_h( b1, a1, a2 ) *
            test_cote_du_segment_h( b2, a1, a2 ) );
}
```

Remarque : Cette méthode prend en compte le cas des intersections sur les extrémités des segments ($\text{det} == 0.0$ dans la fonction `test_cote_du_segment_h`)

Annexe C

Méthode de Gréville

C.1 Présentation

L'algorithme conçu par Gréville utilise une construction ligne par ligne de l'inverse d'une matrice à partir des colonnes de celle-ci. Son principal intérêt vient du fait qu'il peut fonctionner aussi bien pour des matrices non inversibles, conduisant ainsi à la notion de *pseudo-inverse*.

Les matrices pseudo-inverses, trouvent leurs applications à la résolution des systèmes d'équations linéaires quelles que soient leurs dimensions :

$$y = Ax$$

où $A \in \mathbb{R}^{m \times n}$, $y \in \mathbb{R}^m$ est le vecteur cherché, $x \in \mathbb{R}^n$ est le vecteurs des connaissances.

L'inverse généralisée d'un tel système est noté A^+ . L'inverse généralisée satisfait les conditions suivantes :

$$\begin{aligned} AA^+A &= A \\ A^+AA^+ &= A^+ \\ (AA^+)^T &= AA^+ \quad \text{Condition de symétrie} \\ (A^+A)^T &= A^+A \end{aligned}$$

La solution d'un système linéaire à partir de la pseudo-inverse A^+ s'écrit alors : $x = A^+y$.

C.2 Mise en œuvre

La détermination récursive de la matrice pseudo-inverse de A repose sur l'idée de partition de A en colonnes.

A un indice k donné, la pseudo-inverse est calculé à partir de A_{k-1}^+ et de la colonne courante d'indice k de la matrice A .

Soit $A_k = [A_{k-1} \quad a_k]$, où a_k désigne la k -ième colonne de A_k , alors [Koh84] :

$$A_k^+ = \begin{pmatrix} A_{k-1} (I - a_k p_k^T) \\ p_k^T \end{pmatrix}$$

$$\text{où } p_k = \begin{cases} (I - A_{k-1} A_{k-1}^+) a_k / \| (I - A_{k-1} A_{k-1}^+) a_k \|^2 & \text{si } (I - A_{k-1} A_{k-1}^+) a_k \neq 0 \\ (A_{k-1}^+)^T A a_k / (1 + \|A_{k-1}^+ a_k\|^2) & \text{sinon} \end{cases}$$

Les conditions initiales sont les suivantes :

$$\begin{aligned} A_1 &= a_1 \\ A_1^+ &= \begin{cases} a_1^T (a_1^T a_1)^{-1} & \text{si } a_1 \neq 0 \\ 0^T & \text{sinon} \end{cases} \end{aligned}$$

Cette page est laissée blanche intentionnellement

Annexe D

Détermination des coefficients du filtre de Deriche

Dans cette annexe nous détaillons les calculs de détermination des coefficients des filtres de lissage et de dérivation utilisés par le filtre de Deriche pour la détection de contours dans une image. Le calcul des paramètres des filtres est donné pour une réponse à une rampe.

D.1 Filtre de lissage

Soit le filtre défini par :

$$l(x) = b(\alpha|x| + 1)e^{-\alpha|x|}, \text{ où } x \in \mathbb{Z} \quad (\text{D.1})$$

que l'on peut séparer en deux séquences ; l'une causale $l_+(x)$ et l'autre non causale $l_-(x)$:

$$l(x) = l_+(x) + l_-(x) \quad (\text{D.2})$$

$$l_+(x) = \begin{cases} b(\alpha x + 1)e^{-\alpha x} & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

$$l_-(x) = \begin{cases} 0 & \text{si } x \geq 0 \\ b(-\alpha x + 1)e^{\alpha x} & \text{si } x < 0 \end{cases}$$

Étant donné que la transformée en Z est une opération linéaire, la transformée de $l(x)$: $l(z)$ est la somme des deux transformées $l_+(z)$ (de $l_+(x)$) et $l_-(z)$ (de $l_-(x)$).

Déterminons alors $l_+(z)$ et $l_-(z)$.

La transformée en Z de $l_+(x)$ est définie par :

$$\begin{aligned} l_+(z) &= \sum_{x=0}^{+\infty} b(\alpha x + 1)e^{-\alpha x} z^{-x} \\ &= b \sum_{x=0}^{+\infty} \alpha x e^{-\alpha x} z^{-x} + b \sum_{x=0}^{+\infty} e^{-\alpha x} z^{-x} \\ &= b\alpha z \sum_{x=0}^{+\infty} x e^{-\alpha x} z^{-x-1} + b \sum_{x=0}^{+\infty} e^{-\alpha x} z^{-x} \\ &= b\alpha z \frac{d}{dz} \sum_{x=0}^{+\infty} -e^{-\alpha x} z^{-x} + b \sum_{x=0}^{+\infty} e^{-\alpha x} z^{-x} \end{aligned}$$

$\sum_{x=0}^{+\infty} e^{-\alpha x} z^{-x}$ converge si et seulement si : $|e^{-\alpha} z^{-1}| < 1$.

$$\begin{aligned} l_+(z) &= -b\alpha z \frac{d}{dz} \left\{ \frac{1}{1 - e^{-\alpha} z^{-1}} \right\} + b \frac{1}{1 - e^{-\alpha} z^{-1}} \\ &= b\alpha z \frac{e^{-\alpha} z^{-2}}{(1 - e^{-\alpha} z^{-1})^2} + b \frac{1 - e^{-\alpha} z^{-1}}{(1 - e^{-\alpha} z^{-1})^2} \\ &= b \frac{\alpha e^{-\alpha} z^{-1} + 1 - e^{-\alpha} z^{-1}}{1 - 2e^{-\alpha} z^{-1} + e^{-2\alpha} z^{-2}} \end{aligned}$$

Si $I(z)$ est la transformée du signal d'entrée du filtre et $O(z)$ la transformée du signal de sortie, nous pouvons calculer la partie causale de la transformé en Z du signal de sortie :

$$O_+(z) = bI(z) + b(\alpha - 1)e^{-\alpha}I(z)z^{-1} + 2e^{-\alpha}O_+(z)z^{-1} - e^{-2\alpha}O_+(z)z^{-2} \quad (D.3)$$

ce qui donne l'équation récursive permettant de déterminer une partie du signal de sortie du filtre :

$$O_+(x) = bI(x) + b(\alpha - 1)e^{-\alpha}I(x - 1) + 2e^{-\alpha}O_+(x - 1) - e^{-2\alpha}O_+(x - 2) \quad (D.4)$$

De façon similaire, nous traitons la partie non causal du filtre.

$$\begin{aligned} l_-(z) &= b\alpha \sum_{x=-\infty}^{-1} x e^{\alpha x} z^{-x} + b \sum_{x=-\infty}^{-1} e^{\alpha x} z^{-x} \\ &= b\alpha z \sum_{x=0}^{+\infty} x e^{-\alpha x} z^x + b \sum_{x=0}^{+\infty} e^{-\alpha x} z^x - b \\ &= b\alpha z \frac{d}{dz} \left\{ \frac{1}{1 - e^{-\alpha} z} \right\} + b \frac{1}{1 - e^{-\alpha} z} - b \\ &= b\alpha \frac{e^{-\alpha} z}{(1 - e^{-\alpha} z)^2} + b \frac{1 - e^{-\alpha} z}{(1 - e^{-\alpha} z)^2} - b \frac{(1 - e^{-\alpha} z)^2}{(1 - e^{-\alpha} z)^2} \\ &= \frac{b(\alpha + 1)e^{-\alpha} z - be^{-2\alpha} z^2}{1 - 2e^{-\alpha} z + e^{-2\alpha} z^2} \end{aligned}$$

ce qui donne la transformée de la partie non causal du signal de sortie du filtre :

$$O_-(z) = b(\alpha + 1)e^{-\alpha}I(z)z + be^{-2\alpha}I(z)z^2 + 2e^{-\alpha}O_-(z)z - e^{-2\alpha}O_-(z)z^2 \quad (D.5)$$

ce qui donne l'équation récursive permettant de déterminer l'autre partie du signal de sortie du filtre :

$$O_-(x) = b(\alpha + 1)e^{-\alpha}I(x + 1) + be^{-2\alpha}I(x + 2) + 2e^{-\alpha}O_-(x + 1) - e^{-2\alpha}O_-(x + 2) \quad (D.6)$$

Il reste à déterminer le coefficient normalisateur b , qui est calculé de sorte que la réponse à un signal d'amplitude 1 soit un signal d'amplitude 1. Soit ;

$$\sum_{x=-\infty}^{+\infty} l(x) = 1$$

$$\begin{aligned}
\sum_{x=-\infty}^{-1} l_-(x) + \sum_{x=0}^{-\infty} l_+(x) &= 1 \\
\sum_{x=-\infty}^{-1} b(-\alpha x + 1)e^{\alpha x} + \sum_{x=0}^{-\infty} b(\alpha x + 1)e^{-\alpha x} &= 1 \\
-b \sum_{x=-\infty}^{-1} \alpha x e^{\alpha x} + b \sum_{x=-\infty}^{-1} e^{\alpha x} + b \sum_{x=0}^{+\infty} \alpha x e^{-\alpha x} + b \sum_{x=0}^{+\infty} e^{-\alpha x} &= 1 \\
-2 \sum_{x=0}^{+\infty} b \alpha x e^{-\alpha x} + 2 \sum_{x=0}^{+\infty} b e^{-\alpha x} - b &= 1 \\
-2b\alpha \frac{d}{d\alpha} \sum_{x=0}^{+\infty} e^{-\alpha x} + 2b \sum_{x=0}^{+\infty} e^{-\alpha x} - b &= 1 \\
-2b\alpha \frac{d}{d\alpha} \frac{1}{1 - e^{-\alpha}} + 2b \frac{1}{1 - e^{-\alpha}} - b &= 1 \\
\frac{2b\alpha e^{-\alpha} + 2b - 2be^{-\alpha} - b + 2e^{-\alpha}b - be^{-2\alpha}}{1 - 2e^{-\alpha} + e^{-2\alpha}} &= 1 \\
b \frac{2\alpha e^{-\alpha} + 1 - e^{-2\alpha}}{(1 - e^{-\alpha})^2} &= 1 \\
b &= \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}} \tag{D.7}
\end{aligned}$$

D.2 Filtre de dérivation

Soit le filtre défini par :

$$d(x) = kxe^{-\alpha|x|} \tag{D.8}$$

que l'on peut séparer en une partie causale $d_+(x)$ et une partie non causale $d_-(x)$:

$$d(x) = d_+(x) + d_-(x) \tag{D.9}$$

$$d_+(x) = \begin{cases} kxe^{-\alpha x} & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

$$d_-(x) = \begin{cases} 0 & \text{si } x \geq 0 \\ kx1e^{\alpha x} & \text{si } x < 0 \end{cases}$$

Ces deux parties du filtre ont pour transformée en Z :

$$\begin{aligned}
d_+(z) &= \sum_{x=0}^{+\infty} kxe^{-\alpha x} z^{-x} \\
&= k \sum_{x=0}^{+\infty} e^{-\alpha x} xz^{-x} = kz \sum_{x=0}^{+\infty} e^{-\alpha x} xz^{-x-1} \\
&= -kz \frac{d}{dz} \left\{ \sum_{x=0}^{+\infty} e^{-\alpha x} z^{-x} \right\} \\
&= -kz \frac{d}{dz} \left\{ \frac{1}{1 - e^{-\alpha} z^{-1}} \right\} \\
&= kz \frac{e^{-\alpha} z^{-1}}{1 - 2e^{-\alpha} z^{-1} + e^{-2\alpha} z^{-2}}
\end{aligned}$$

ce qui donne la transformée en Z du signal de sortie de la partie causal du filtre suivante :

$$O_+(z) = ke^{-\alpha} I(z)z^{-1} + 2e^{-\alpha} O_+(z)z^{-1} - e^{-2\alpha} O_+(z)z^{-2}$$

ce qui conduit à l'équation récurrente suivante :

$$O_+(x) = ke^{-\alpha}I(x-1) + 2e^{-\alpha}O_+(x-1) - e^{-2\alpha}O_+(x-2)$$

De façon analogue, nous allons déterminer maintenant la partie non causale du signal.

$$\begin{aligned} d_-(z) &= \sum_{x=-\infty}^{-1} kxe^{\alpha x}z^{-x} \\ &= kz \sum_{x=-\infty}^{-1} e^{\alpha x}xz^{-x-1} \\ &= -kz \frac{d}{dz} \left\{ \sum_{x=-\infty}^{-1} e^{\alpha x}z^{-x} \right\} \end{aligned}$$

en posant $t = -x$,

$$\begin{aligned} d_-(z) &= -kz \frac{d}{dz} \left\{ \sum_{t=\infty}^1 e^{-\alpha t}z^t \right\} \\ &= -kz \frac{d}{dz} \left\{ \sum_{t=0}^{\infty} e^{-\alpha t}z^t - 1 \right\} \\ &= -kz \frac{d}{dz} \left\{ \frac{1}{1 - e^{-\alpha}z} - 1 \right\} \\ &= -kz \frac{d}{dz} \left\{ \frac{e^{-\alpha}z}{1 - e^{-\alpha}z} \right\} \\ &= -kz \frac{e^{-\alpha}(1 - e^{-\alpha}z) + e^{-\alpha}ze^{-\alpha}}{1 - 2e^{-\alpha}z + e^{-2\alpha}z^2} \\ &= -kz \frac{e^{-\alpha} - e^{-2\alpha}z + e^{-2\alpha}z}{1 - 2e^{-\alpha}z + e^{-2\alpha}z^2} \end{aligned}$$

$$d_-(z) = -k \frac{e^{-\alpha}z}{1 - 2e^{-\alpha}z + e^{-2\alpha}z^2}$$

Nous obtenons alors :

$$O_-(z) = -ke^{-\alpha}I(z)z + 2e^{-\alpha}O_-(z)z - e^{-2\alpha}O_-(z)z^2$$

ce qui conduit à l'équation récurrente suivante :

$$O_-(x) = -ke^{-\alpha}I(x+1) + 2e^{-\alpha}O_+(x+1) - e^{-2\alpha}O_+(x+2)$$

Le coefficient k doit être déterminé en fonction du profil de contour que nous avons choisi, c'est-à-dire que la réponse à un signal de rampe unitaire soit un signal d'amplitude 1, soit :

$$\sum_{x=-\infty}^{+\infty} -xkxe^{-\alpha|x|} = 1$$

$$\sum_{x=-\infty}^{-1} -x^2 k e^{\alpha x} + \sum_{x=0}^{+\infty} -x^2 k e^{-\alpha x} = 1$$

$$\sum_{x=0}^{+\infty} -x^2 k e^{-\alpha x} + \sum_{x=0}^{+\infty} -x^2 k e^{-\alpha x} = 1$$

$$-2k \sum_{x=0}^{+\infty} x^2 e^{-\alpha x} = 1$$

$$-2k \frac{d}{d\alpha} \left\{ \frac{d}{d\alpha} \left\{ \sum_{x=0}^{+\infty} e^{-\alpha x} \right\} \right\} = 1$$

$$-2k \frac{d}{d\alpha} \left\{ \frac{d}{d\alpha} \left\{ \frac{1}{1 - e^{-\alpha}} \right\} \right\} = 1$$

$$-2k \frac{d}{d\alpha} \left\{ \frac{e^{-\alpha}}{(1 - e^{-\alpha})^2} \right\} = 1$$

$$-2k \frac{-e^{-\alpha}(1 - e^{-\alpha}) - 2e^{-\alpha}}{(1 - e^{-\alpha})^3} = 1$$

$$2k \frac{e^{-\alpha}(1 + e^{-\alpha})}{(1 - e^{-\alpha})^3} = 1$$

$$k = \frac{(1 - e^{-\alpha})^3}{2e^{-\alpha}(1 + e^{-\alpha})} \tag{D.10}$$

Cette page est laissée blanche intentionnellement

Annexe E

Résultats d'un traitement d'image

Cette annexe présente les différentes étapes du traitement d'image à partir des outils issus du domaine du traitement d'image que nous avons adapté et mise en œuvre pour la détection de contours verticaux dans une image lors de l'ajustement en orientation du VAHM.

Les figures E.1 à E.7 donnent les différentes phases de la détection de la position des contours verticaux dans l'image.

La figure E.9 représente le flux du traitement de l'image source. Les différentes étapes sont rappelées avec en sortie l'image correspondante.

L'image de la figure E.8 donne une représentation de la projection des contours détectés sur l'image originale. Nous pouvons noter la bonne localisation et l'unicité de la détection de chaque contours.

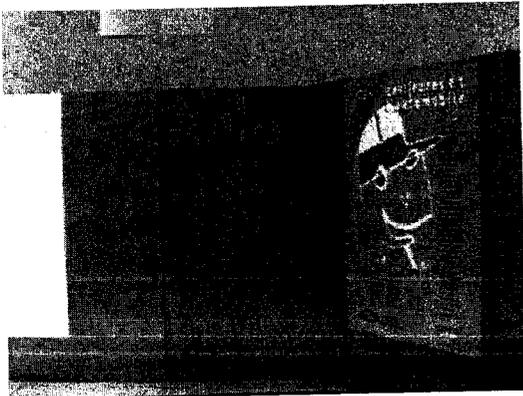


Figure E.1 – Image fournie par la caméra.

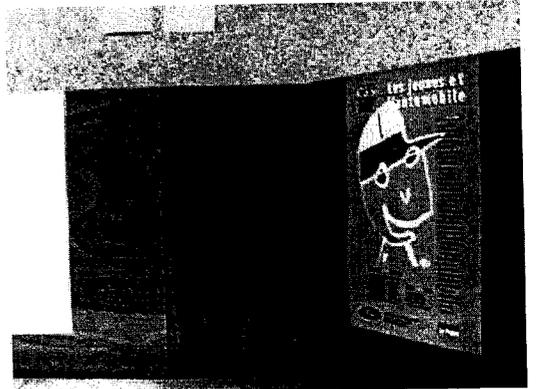


Figure E.2 – Image égalisée.

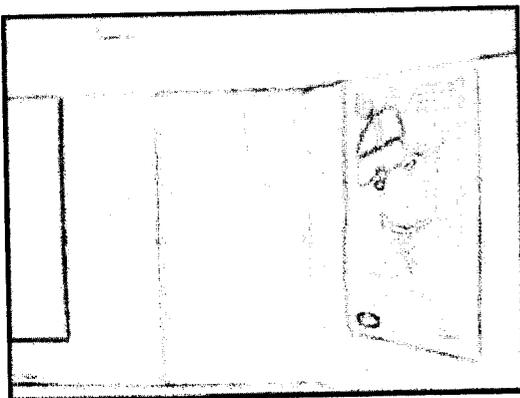


Figure E.3 – Image de la norme du gradient en lut inverse.

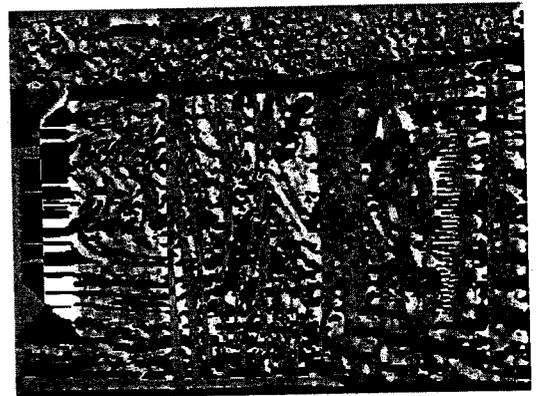


Figure E.4 – Image de la direction du gradient.

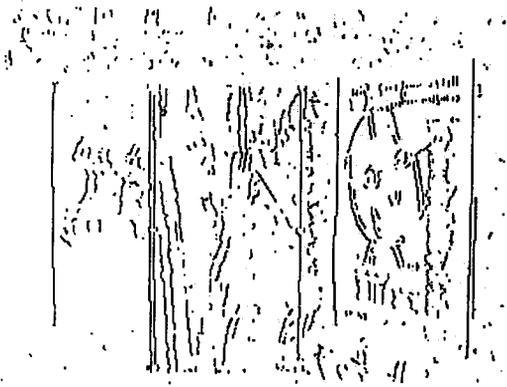


Figure E.5 – Image du gradient avec suppression des non maxima locaux en lut inverse.

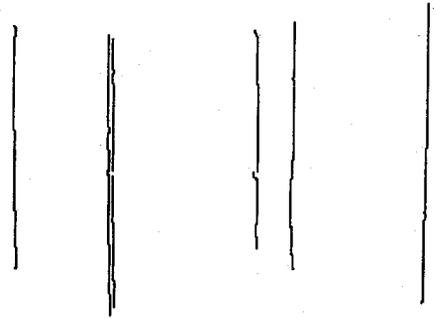


Figure E.6 – Image issue du chaînage de contour.

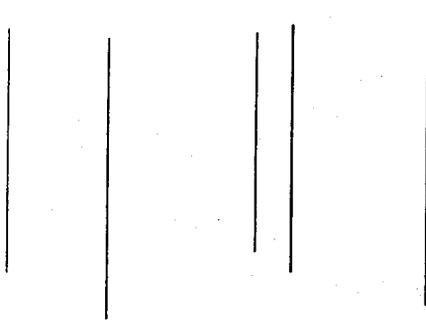


Figure E.7 – Représentation des contours détectés.

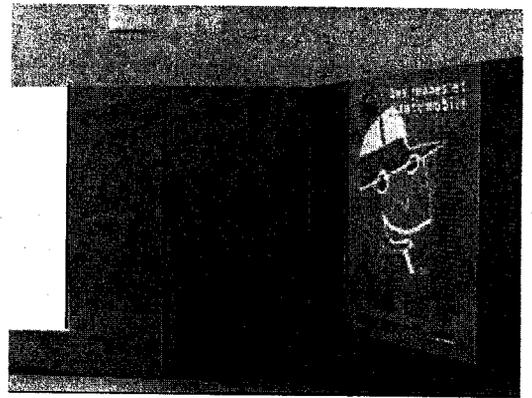


Figure E.8 – Contours détectés en surimpression de l'image originale.

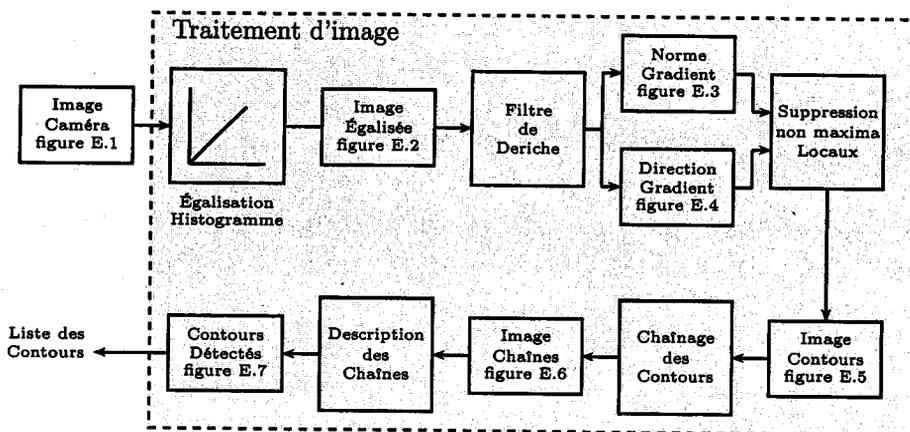
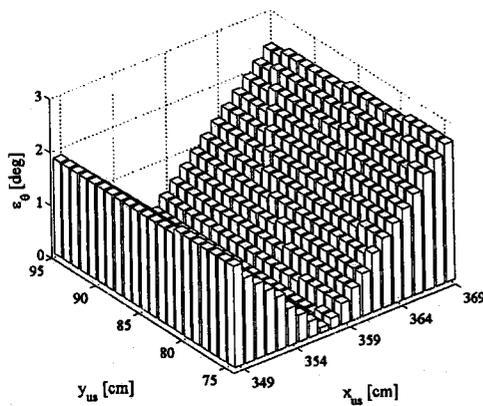


Figure E.9 – Schéma du traitement de l'image fournie par la caméra.

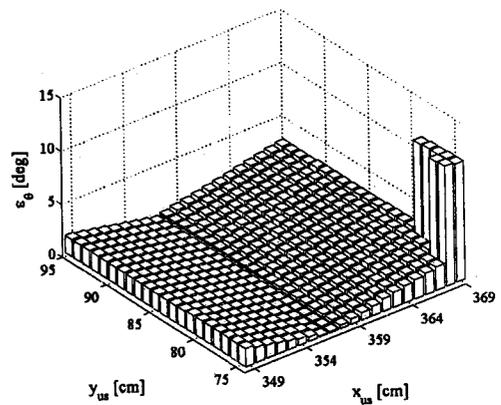
Annexe F

Résultats de l'influence d'une erreur en conditions réelles

Les résultats donnés dans cette annexe ont été obtenus pour une erreur initiale en orientation de $\varepsilon_\theta = 10^\circ$ et $\varepsilon_\theta = 10^\circ$ et pour une plage de variation de l'erreur en position de $\pm 10\text{ cm}$ autour de la position réelle.

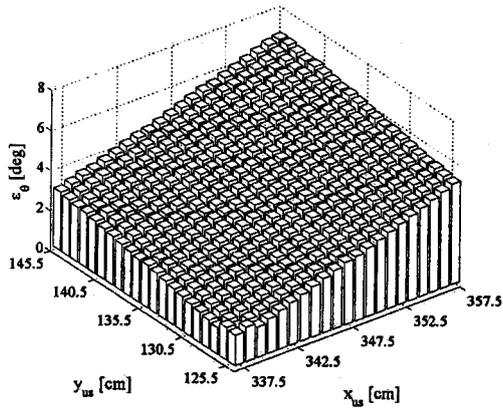


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

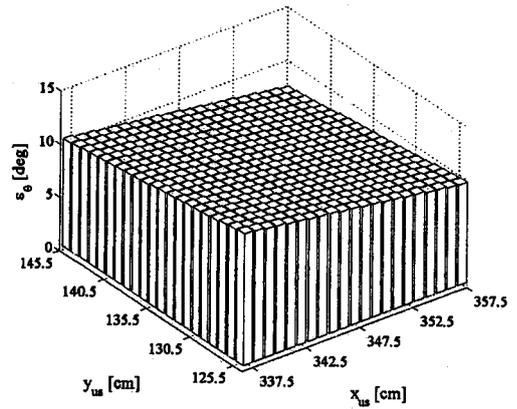


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.1 – Essai 00 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.

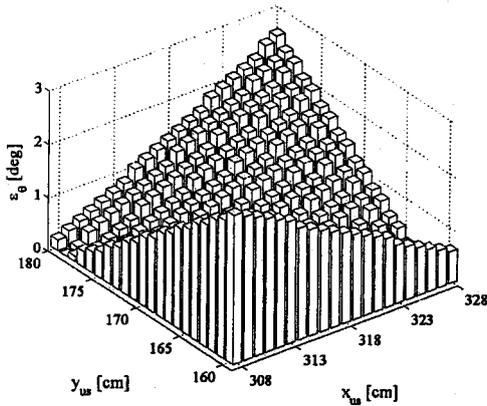


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

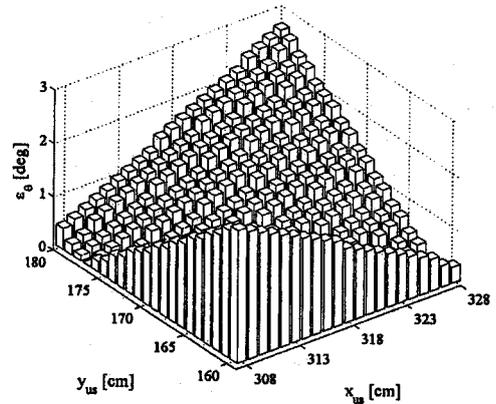


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.2 – Essai 01 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.

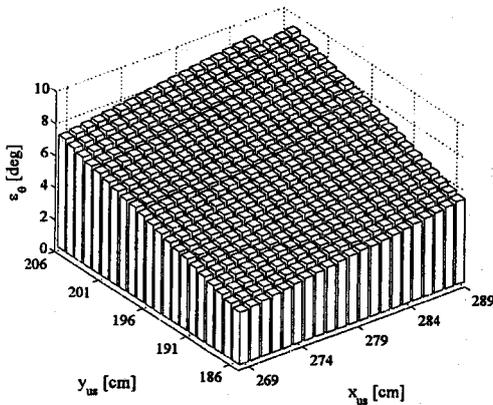


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

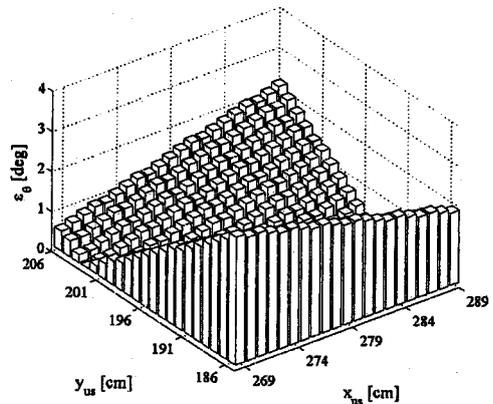


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.3 – Essai 02 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.



(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$



(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.4 – Essai 03 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.

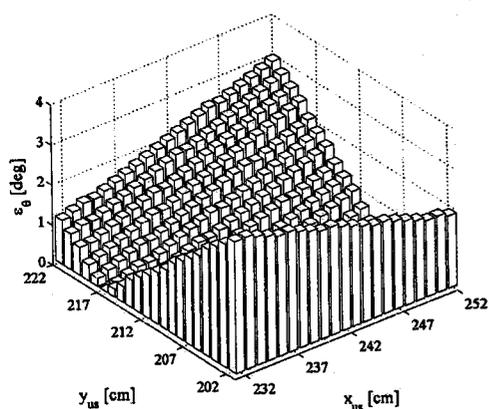
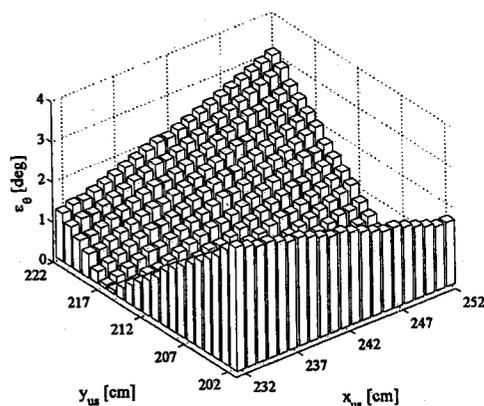
(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$ (b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.5 – Essai 04 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.

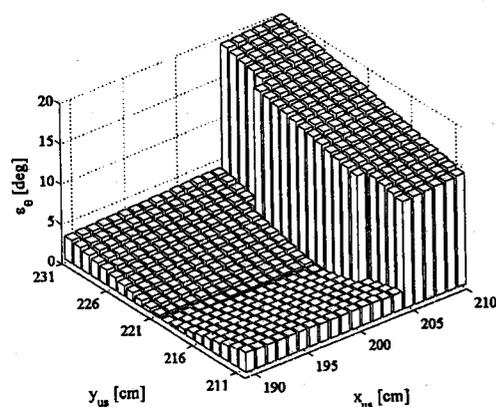
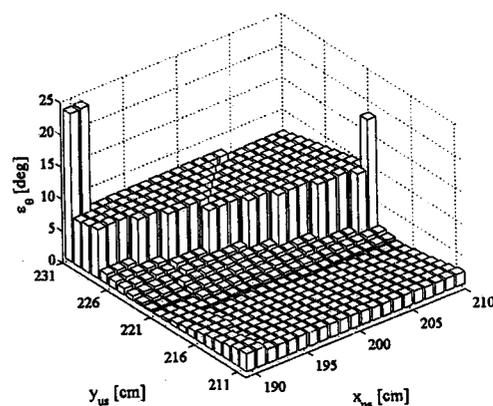
(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$ (b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.6 – Essai 05 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.

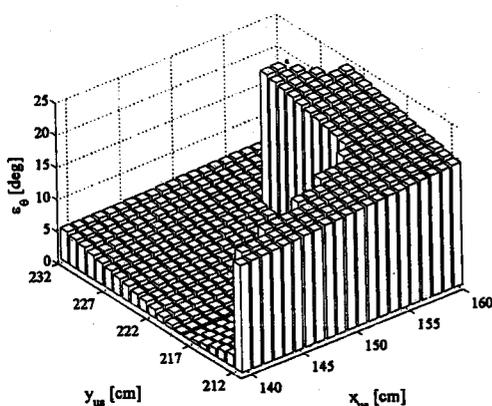
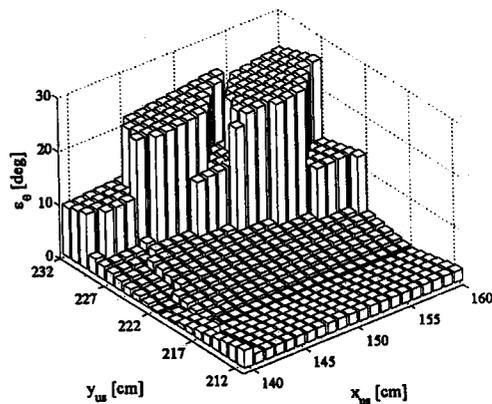
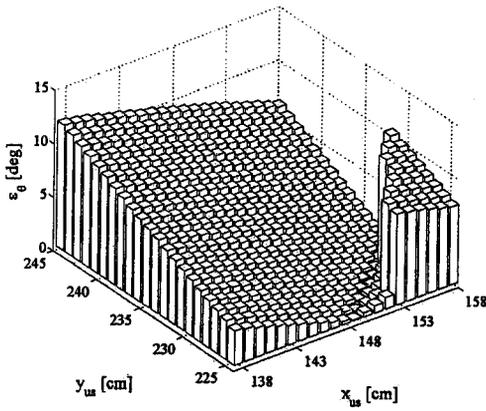
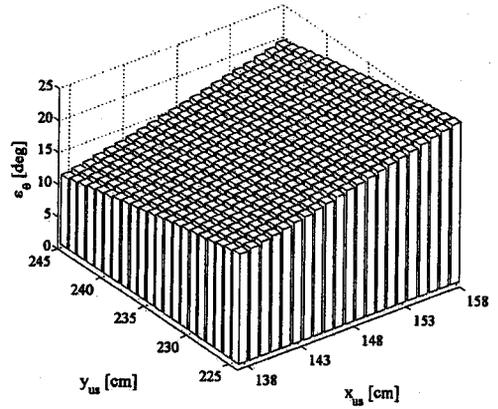
(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$ (b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.7 – Essai 06 : Répartition de l'erreur en orientation après ajustement en fonction de la position estimée.

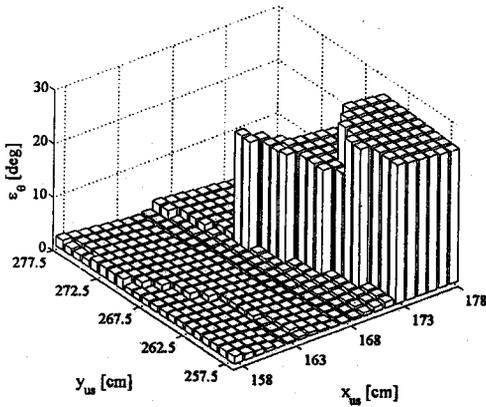


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

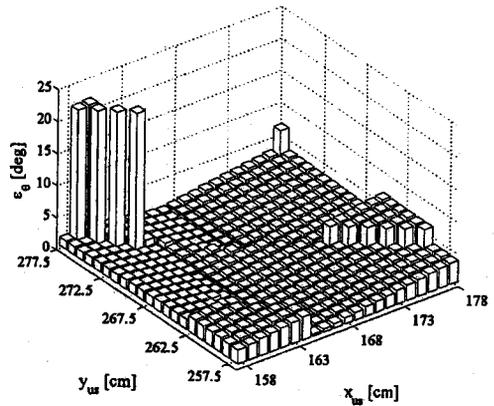


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.8 – Essai 07 : Répartition de l’erreur en orientation après ajustement en fonction de la position estimée.

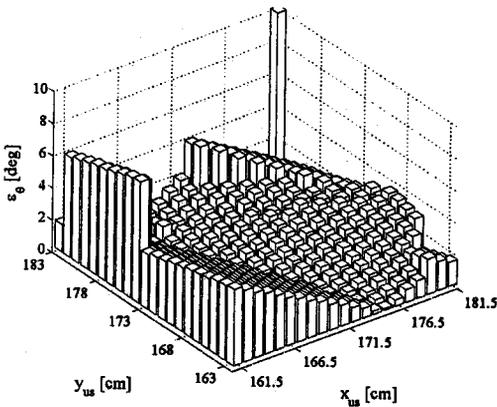


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

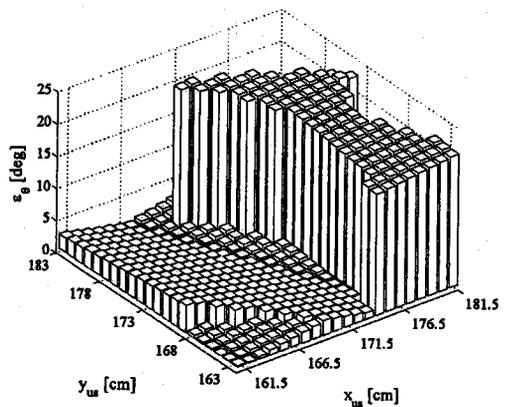


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.9 – Essai 09 : Répartition de l’erreur en orientation après ajustement en fonction de la position estimée.

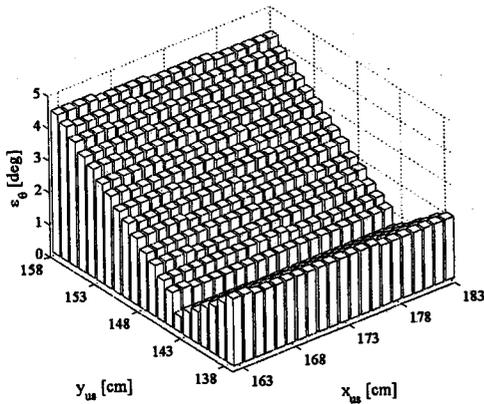


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

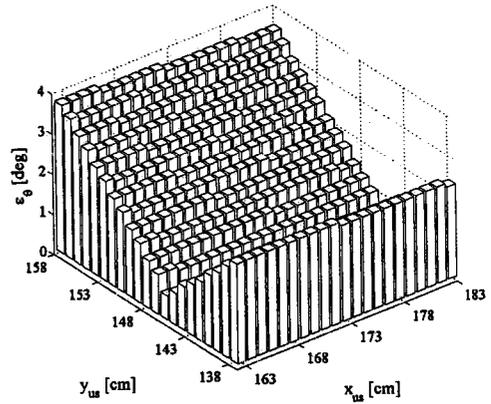


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.10 – Essai 11 : Répartition de l’erreur en orientation après ajustement en fonction de la position estimée.

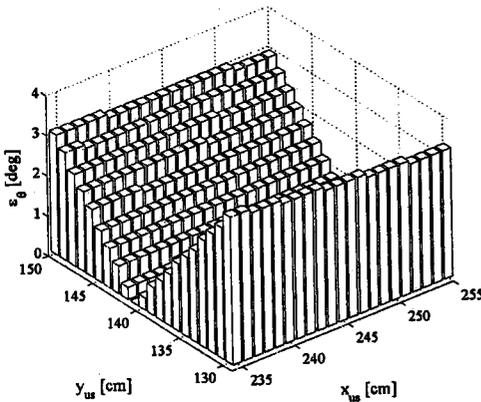


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

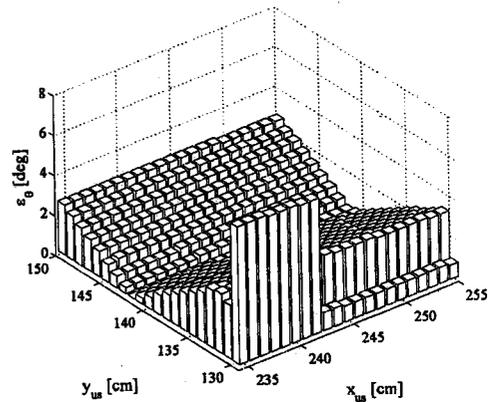


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.11 – Essai 13 : Répartition de l’erreur en orientation après ajustement en fonction de la position estimée.

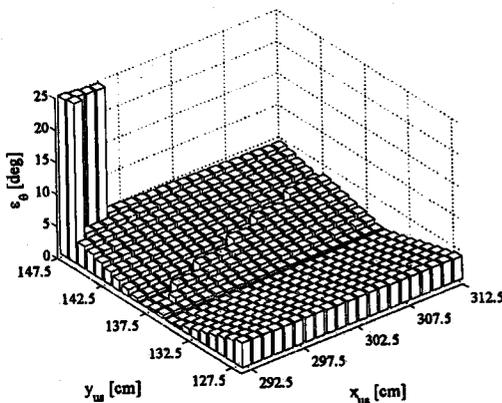


(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$

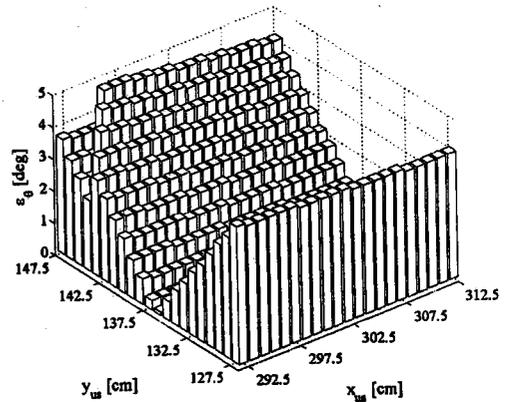


(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.12 – Essai 14 : Répartition de l’erreur en orientation après ajustement en fonction de la position estimée.



(a) Erreur en orientation initiale $\varepsilon_\theta = -10^\circ$



(b) Erreur en orientation initiale $\varepsilon_\theta = +10^\circ$

Figure F.13 – Essai 15 : Répartition de l’erreur en orientation après ajustement en fonction de la position estimée.

Cette page est laissée blanche intentionnellement

Bibliographie

- [Aid02] O. Ait AIDER, P. HOPPENOT et E. COLLE : Adaptation of lowe's camera pose recovery algorithm to mobile robot self-localisation. *Robotica*, 20:385-393, 2002.
- [Alt03] P. ALTHAUS et H. CHRISTENSEN : Automatic map acquisition for navigation in domestic environments. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, September 2003.
- [And92] C. ANDERSEN, C. MADSEN, J. SORENSEN, N. KIRKEBY, J. JONES et H. CHRISTENSEN : Navigation using range images on a mobile robot. *IEEE Transaction on Robotics and Automation*, 10(2-3):147-160, 1992.
- [Arr97] K.O. ARRAS et R.Y. SIEGWART : Feature extraction and scene interpretation for map-based navigation and map building. Dans *Proceedings of the SPIE, Mobile Robotics XII*, volume 3210, pages 42-53, 1997.
- [Arr99] K.O. ARRAS et N. TOMATIS : Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision. Dans *Proceedings of the 3rd European Workshop on Advanced Mobile Robots*, Zurich, Switzerland, september 1999.
- [Arr00] K.O. ARRAS, N. TOMATIS et R. SIEGWART : Multisensor on-the-fly localization using laser and vision. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 462-467, Takamatsu, Japan, october / november 2000.
- [Arr01] Kai O. ARRAS, Nicola TOMATIS, Björn T. JENSEN et Roland SIEGWART : Multisensor on-the-fly localization: Precision and reliability for applications. *Robotics and Autonomous Systems*, 34:131-143, 201.
- [Ayc97] O. AYCARD, F. CHARPILLET et J. HATON : A new approach to design fuzzy controllers for mobile robots navigation. Dans *Proceedings of the IEEE International Symposium on Computational Intelligence*, pages 68-73, 1997.
- [Bal95] T. BALCH et R. ARKIN : Motor schema-based formation control for multiagent robot teams. Dans *Proceedings of the International Conference on Multiagent Systems*, pages 10-16, 1995.
- [Bel94] D. BELL, J. BORENSTEIN, S. LEVINE, Y. KOREN et A. JAROS : The navchair: An assistive navigation system for wheelchairs, based on mobile robot obstacle avoidance. Dans *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2012-2017, San Diego, CA, may 1994.
- [Bor89] J. BORENSTEIN et Y. KOREN : Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19:1179-1187, 1989.
- [Bor90a] J. BORENSTEIN et Y. KOREN : Real-time obstacle avoidance for fast mobile robots in cluttered environment. Dans *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 572-577, 1990.
- [Bor90b] J. BORENSTEIN, S. LEVINE et Y. KOREN : The navchair - a new concept in intelligent wheelchair control for people with multiple handicaps. Dans *CSUN's Fifth Annual Conference on Technology and Persons with Disabilities*, pages 117-124, Los Angeles, California, March 1990.
- [Bor91a] J. BORENSTEIN et Y. KOREN : Histogram in-motion mapping for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 7(4):535-539, 1991.

- [Bor91b] J. BORENSTEIN et Y. KOREN : The vector field histogram – fast obstacle-avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288, June 1991.
- [Bor92] J. BORENSTEIN et Y. KOREN : Noise rejection for ultrasonic sensors in mobile robot applications. Dans *Proceedings of the International Conference on Robotics and Automation*, pages 1727–1732, 1992.
- [Bor93] S. BORTHWICK, M. STEVENS et H. DURRANT-WHYTE : Position estimation and tracking using optical range data. Dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2172–2177, 1993.
- [Bor95] U. BORGOLTE, R. Hoelper ans H. HOYER, H. HECK, W. HUMANN, J. NEZDA, I. CRAIG, R. VALLEGGI et A.M. SABATINI : Intelligent control of a semi-autonomous omnidirectional wheelchair. Dans *Proceedings of the 3rd International Symposium on Intelligent Robotic Systems*, pages 113–120, Pisa, Italy, July 1995.
- [Bor96] J. BORENSTEIN : *Navigating Mobile Robots: Systems and Techniques*. A.K. Peters, Ltd., 1996.
- [Bor02a] G. A. BORGES et M.-J. ALDON : A decoupled approach for simultaneous stochastic mapping and mobile robot localization. Dans *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, page Accepted for publication, Lausanne, Switzerland, september 2002.
- [Bor02b] G. A. BORGES et M.-J. ALDON : Optimal mobile robot pose estimation using geometrical maps. *IEEE Transactions on Robotics and Automation*, 18(1):87–94, february 2002.
- [Bou86] G. BOURHIS, J.M. ANDRE et E. YVROUD : Versatile control system of electric wheelchair for severely motor handicapped people. Dans *6^o congresso Brasileiro de Automatica*, Belo Horizonte, Brésil, November 1986.
- [Bou98] G. BOURHIS et Y. AGOSTINI : Man-machine cooperation for the control of an intelligent powered wheelchair. *Journal of Intelligent and Robotic Systems, Special Issue on 'Mobile Robots in Health Care Services'*, 22:269–287, 1998.
- [Bou00] Denis BOUVET : *Contribution à la localisation d'engins de chantiers routiers*. Thèse de doctorat, École Centrale de Nantes, novembre 2000.
- [Bou01] G. BOURHIS, O. HORN, O. HABERT et A. PRUSKI : The vahm project: autonomous vehicle for people with motor disabilities. *IEEE Robotics and Automation Magazine, Special Issue on 'Research on Autonomous Robotics Wheelchairs in Europe'*, 7(1), mars 2001.
- [Bro84] R.A. BROOKS : Aspects of mobile robot visual map making. Dans *Proceedings of the Second International Symposium of Robotics Research*, pages 287–293, Kyoto, Japan, August 1984. MIT Press.
- [Buc93] M. BUCHBERGER, K.W. JÖRG et E. von PUTTKAMER : Laserradar and sonar based world modeling and motion control for fast obstacle avoidance of the autonomous mobile robot mobot-iv. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 534–540, 1993.
- [Can83] J.F. CANNY : *Finding edges and lines in images*. Master's thesis, Massachusetts Institute of Technology, Cambridge, USA, 1983.
- [Cau02] C. CAUCHOIS, E. BRASSART, B. MARHIC et C. DROCOURT : An absolute localization method using a synthetic panoramic image base. Dans *IEEE Workshop on Omnidirectional Vision (OMNIVIS'02) at European Conference on Computer Vision (ECCV 2002)*, Copenhagen, Denmark, June 2002.
- [Cha85] R. CHATILA et J. LAUMOND : Position referencing and consistent world modeling for mobile robots. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 138–145, 1985.
- [Cho90] D.W. CHO : Certainty grid representation for robot navigation by a bayesian method. *Robotica*, 8:159–165, 1990.

- [Cho97] K.S. CHONG et L. KLEEMAN : Sonar based map building for a mobile robot. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1700–1705, avril 1997.
- [Chr98] Stéphane CHRISTY : *Localisation et modélisation tridimensionnelles par approximations successives du modèle perspectif de caméra*. Thèse de doctorat, informatique, GRAVIR-IMAG, Grenoble, Août 1998.
- [Con02] L. CONDE, G. PIRES et U.NUNES : A behaviour based fuzzy control architecture for path-tracking and obstacle avoidance. Dans *5th Portuguese Conference on Automatic Control*, pages 341–346, 2002.
- [Cou00] Alain COURCELLE : *Localisation d'un robot mobile : Application à l'aide à la mobilité des personnes handicapées moteur*. Thèse de doctorat, Université de Metz, janvier 2000.
- [Cox91] I.J. COX : Blanche: An experiment in guidance and navigation of an autonomous mobile robot. *IEEE Transactions on Robotics and Automation*, 7(3):193–204, 1991.
- [Cri03] Z. CRISMAN, E. CURRE, C. KWOK, N. RATLIFF, L. TSYBERT et D. FOX : Team description: Uwhuskies-02. Dans G. KAMINKA, P. LIMA et R. ROJAS, éditeurs : *RoboCup-2002: Robot Soccer World Cup VI*. Springer-Verlag, 2003.
- [Cro85] J. CROWLEY : Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, 1:31–41, march 1985.
- [Cro89] J.L. CROWLEY : World modeling and position estimation for a mobile robot using ultrasonic ranging. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1277–1282, May 1989.
- [Der87] R. DERICHE : Extraction de composantes connexes basée sur une détection optimal des contours. Dans *Proceedings of Cognitiva*, Paris La Villette, mai 1987.
- [Dru87] M. DRUMHELLER : Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2):325–332, march 1987.
- [Elf87] A. ELFES : Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3:249–265, june 1987.
- [Eve95] H. EVERETT : *Sensors for Mobile Robots: Theory and Application*. A.K. Peters, Ltd., 1995.
- [Fab00] E. FABRIZI et A. SAFFIOTTI : Extracting topology-based maps from gridmaps. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2972–2978, San Francisco, CA, USA, 2000.
- [Fly88] A. FLYNN : Combining sonar and infrared sensors for mobile robot navigation. *The International Journal of Robotics Research*, 7:5–14, 1988.
- [For93] J. FORSBERG, P. ÅHMAN et Å. WERNERSSON : The hough transformation inside the feedback loop of a mobile robot. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 791–798, 1993.
- [Fox99] D. FOX, W. BURGARD, F. DELLAERT et S. THRUN : Monte carlo localization: Efficient position estimation for mobile robots. Dans *Proceedings of National Conference on Artificial Intelligence*, 1999.
- [Gai03] J.P. GAILLARD, D. FREARD, E. COLLE et P. HOPPENOT : Operator's self confidence to detect mobile robot trajectory errors. *Le Travail Humain*, 66(1):1–21, 2003.
- [Goe99] Puneet GOEL, Stergios I. ROUMELIOTIS et Gaurav S. SUKHATME : Robot localization using relative and absolute position estimates. Dans *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1134–1140, october 1999.
- [Gri87] W.E.L. GRIMSON et T. LOZANO-PEREZ : Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):469–482, july 1987.
- [Gut02a] Jens-Steffen GUTMANN : Markov-kalman localization for mobile robots. Dans *Proceedings of the International Conference on Pattern Recognition*, august 2002.

- [Gut02b] Jens-Steffen GUTMANN et Dieter FOX : An experimental comparison of localization methods continued. Dans *Proceedings of the IEE/RSJ International Conference on Intelligent Robot and Systems*, october 2002.
- [Hag90] G. HAGER et M. MINTZ : Sensor modeling and robust sensor data fusion. Dans *Proceedings of the International Symposium on Robotics Research*, pages 69–74, 1990.
- [Hal01] Ireneusz HALLMANN et Barbara SIEMIATKOWSKA : Artificial landmark navigation system. Dans *Proceedings of the 9th International Symposium on Intelligent Robotic Systems*, july 2001.
- [Har01] R. HARTLEY et A. ZISSERMAN : *Multiple View Geometry in computer vision*. Cambridge University Press, 2001.
- [Hay01] J.-B. HAYET, F. LERASLE et M. DEVY : De l'utilisation d'amers plans pour la navigation d'un robot mobile en milieu intérieur. Dans *ORASIS 2001, Congrès francophone de vision, Cahors*, pages 193–202, IRIT, 118, route de Narbonne, 31062 Toulouse Cedex 4, France, 5–8 juin 2001.
- [Her03] S. HERNÁNDEZ, J.M. TORRES, C.A. MORALES et L. ACOSTA : A new low cost system for autonomous robot heading and position localization in a closed area. *Autonomous Robots*, 15:99–110, 2003.
- [Hon01] S. D'Attansio HONIGER, E. PERUCHON et P. COIFFET : Applications des robots mobiles à l'assistance des personnes handicapées. *Rencontres en rééducation : Robotique, domotique et handicap*, 16:27–35, 2001.
- [Hop90] P. HOPPEN, T. KNIERIEMEN et E. von PUTTKAMER : Laser-radar based mapping and navigation for an autonomous mobile robot. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 948–953, 1990.
- [Hop97] P. HOPPENOT et E. COLLE : Real-time mobile robot localisation with poor ultrasonic data. Dans *3rd IFAC Symposium on Intelligent Component and Instrument for Control Application (SICICA)*, pages 135–140, Annecy, France, Juin 1997.
- [Hop98] P. HOPPENOT et E. COLLE : Real-time localisation of a low-cost mobile robot with poor ultrasonic data. *IFAC journal, Control Engineering Practice*, 6:925–934, 1998.
- [Hop01] P. HOPPENOT et E. COLLE : Localisation and control of a rehabilitation robot by close human-machine co-operation. *IEEE Transaction on Neural System and Rehabilitation Engineering*, 9, June 2001.
- [Hor95] R. HORAUD et O. MONGA : *Vision par Ordinateur : outils fondamentaux*. Hermes, 1995.
- [Hor00] O. HORN, A. COURCELLE et M. KREUTNER : Le projet vahm (véhicule autonome pour handicapés moteur) : la localisation. *Revue Traitement du signal*, 17(3, spécial 2000):221–232, 2000.
- [Hor03a] O. HORN et A. COURCELLE : Interpretation of ultrasonic readings for autonomous robot localization. *Journal of Intelligent and Robotic System*, 2003.
- [Hor03b] O. HORN et M. KREUTNER : Perception par ultrasons et vision pour la localisation d'un fauteuil électrique autonome. Dans *Journée Nationale sur 'Image et Signal pour le Handicap'*, Paris, octobre 2003.
- [How03] Andrew HOWARD, Maja J. MATARIC et S. SUKHATME : Cooperative relative localization for mobile robot teams: An ego-centric approach. Dans *Proceedings of the Multi-Robot Systems Workshop*, Naval Research Laboratory Washington D.C., march 2003.
- [Hoy94] H. HOYER et R. HOELPER : Intelligent omnidirectional wheelchair with a flexible configurable functionality. Dans *Proceedings of the 17th Annual RESNA Conference*, pages 353–355, Memphis, TN. Washington, June 1994.
- [Hoy95] H. HOYER, R. HOELPER, U. BORGOLTE, C. BÜHLER, H. HECK, W. HUMANN, I. CRAIG, R. VALLEGGI et A.M. SABATINI : The omni wheelchair with high manoeuvrability and navigational intelligence. Dans *Rehabilitation Technology, Proc. of the 2nd TIDE Congress*, Brussels, 1995. IOS Press.

- [Hoy96] H. HOYER et U. BORGOLTE : *Office Wheelchair with High Manoeuvrability and Navigational Intelligence for People with Severe Handicap (OMNI)*, edited final report – omni consortium édition, 1996. <http://prt.fernuni-hagen.de/pro/omni/efr.ps.gz>.
- [Hoy99] H. HOYER, U. BORGOLTE et A. JOCHHEIM : The omni-wheelchair – state of the art – . http://www.dinf.ne.jp/doc/english/Us_Eu/conf/csun_99/session0274.html, March 5 1999. University at Hagen, Department of Electrical Engineering Control Systems engineering group, D-58084 Hagen, Germany – ulrich.borgolte@fernuni-hagen.de.
- [Jar88] R.A. JARVIS, J.C. BYRNE et K. AJAY : An intelligent autonomous guided vehicle: Localization, environment modeling and collision free path finding. Dans *Proceedings of the 19th ISIR The International Symposium and Exposition on Robots*, pages 767–792, 1988.
- [Koh84] T. KOHONEN : *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1984.
- [Kre01] M. KREUTNER et O. HORN : Contribution to rehabilitation mobile robotics : Localization of an autonomous wheelchair. Dans *Proceedings of the 7th International Conference on Rehabilitation Robotics*, pages 207–214, INT Evry, France, 2001.
- [Kre03] M. KREUTNER et O. HORN : Co-operation between ultrasound and monocular vision for the localization of a mobile robot. Dans *Proceedings of the Computing Engineering in Systems Applications*, Ecole Centrale Université de Lille, France, juillet 2003.
- [Kro89] Eric KROTKOV : Mobile robot localization using a single image. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 978–983, may 1989.
- [Kuc87] R. KUC et M. SIEGEL : Physically based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9:766–778, november 1987.
- [Laf01] I. LAFFONT, S. OLIVIER, M. MOKHTARI, J. HEIDMAN, C. DUMAS et O. DIZIEN : Robotique et handicap : perspectives. *Cahiers de Kinésithérapie*, 208(2):31–34, juin 2001.
- [Lan02] A. LANKENAU et T. RÖFER : Mobile robot self-localization in large-scale environments. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1359–1364, 2002.
- [Lan03] A. LANKENAU, T. RÖFER et B. KRIEG-BRÜCKNER : Self-localization in large-scale environments for the bremen autonomous wheelchair. *Spatial Cognition III : Lecture Notes in Artificial Intelligence 2685*, pages 34–61, 2003.
- [Lee00] Wang-Heun LEE, Kyoung-Sig ROH et In-So KWEON : Self-localization of a mobile robot without camera calibration using projective invariants. *Pattern Recognition Letters*, 21:45–60, 2000.
- [Len00] S. LENSER et M. VELOSO : Sensor resetting localization for poorly modeled mobile robots. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.
- [Leo91] J.J. LEONARD et H. DURRANT-WHYTE : Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.
- [Leo92] J.J. LEONARD et H. DURRANT-WHYTE : *Direct Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publisher, 1992.
- [Lim92] J. LIM et D. CHO : Physically based sensor modeling for a sonar map in a specular environment. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1714–1719, 1992.
- [LoP02] E. LOPRESTI, R. SIMPSON, D. MILLER et I. NOURBAKHSH : Evaluation of sensors for a smart wheelchair. Dans *Proceedings of the 25th Annual Conference on Rehabilitation Engineering (RESNA)*, Minneapolis, June 2002.
- [Mar99] B. MARHIC, El. M. MOUADDIB, C. PEGARD et N. HUTIN : Localisation based on invariant-models recognition by syclop. Dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'99)*, 1999.
- [Mor85] H. MORAVEC et A. ELFES : High resolution maps from wide angle sonar. Dans *Proceedings of the International Conference on Robotics and Automation*, pages 116–121, 1985.

- [Mun92] J.L. MUNDY et A. ZISSERMAN : *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, MA, 1992.
- [Nas92] F. NASHASHIBI, M. DEVY et P. FILLATREAU : Indoor scene terrain modeling using multiple range images for autonomous mobile robots. Dans *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 40–46, 1992.
- [Nei99] José NEIRA, Juan D. TARDÓS, Joachim HORN et Günther SCHMIDT : Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 15(1):76–84, february 1999.
- [Nic85] P. NICKSON : Solide-state tachometry. *Sensors*, pages 23–26, april 1985.
- [Nis88] P.D. NISBET, I.R. LOUDON et J.P. ODOR : The call centre smart wheelchair. Dans *Proceedings of the 1st International Workshop on Robotic Applications to Medical and Health Care*, pages 9.1–9.10, Ottawa, 1988.
- [Nis02] P.D. NISBET : Who's intelligent? wheelchair, driver or both? Dans *Proceedings of the 2002 IEEE International Conference on Control Applications*, Glasgow, 2002.
- [Nom03] NOMADIC : Nomad 200 hardware manual, August 2003.
- [Nun00] U. NUNES, G. PIRES et P. COELHO : *Assistive Navigation Control Architecture*, pages 38–43. World Scientific Publisher, 2000.
- [Ori95] G. ORIOLO, M. VENDITTELLI et G. ULIVI : On-line map building and navigation for autonomous mobile robots. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2900–2906, 1995.
- [Par96] J.R. PARKER : *Algorithms for Image Processing and Computer Vision*. Wiley Computer Publishing, 1996.
- [Pfi02] Sam T. PFISTER, Kristo L. KRIECHBAUM, Stergios I. ROUMELIOTIS et Joel W. BURDICK : Weighted range sensor matching algorithms for mobile robot displacement estimation. Dans *Proceedings of the International Conference on Robotics and Automation*, Washington D.C., may 2002.
- [Pir02] G. PIRES et U. NUNES : A wheelchair steered through voice commands and assisted by a fuzzy-logic controller. *International Journal of Intelligent and Robotic Systems*, 34(3):301–314, July 2002.
- [Pol91] POLAROID : Polaroid ultrasonic ranging system user's manual, 1991.
- [Pru03] Alain PRUSKI : *Assistance technique au handicap (Traité IC2, série Systèmes automatisés)*. HERMES Sciences Publication, Paris, 1 édition, 2003.
- [Ras90] U. RASCHKE et J. BORENSTEIN : A comparison of grid-type map-building techniques by index of performance. Dans *Proceedings of the International Conference on Robotics and Automation*, volume 3, pages 1828–1832, 1990.
- [Röf98] T. RÖFER et A. LANKENAU : Architecture and applications of the bremen autonomous wheelchair. Dans P.P. WANG, éditeur : *Proceedings of the Fourth Joint Conference on Information Systems 1. Association for Intelligent Machinery*, pages 365–368, 1998.
- [Roh97] Kyoung Sig ROH, Wang Heon LEE et In So KWEON : Obstacle detection and self-localization without camera calibration using projective invariants. Dans *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 1030–1035, 1997.
- [Sch97] K. SCHILLING, R. LIEB et H. ROTH : Indoor navigation of mobile robots based on natural landmarks. Dans *Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications*, pages 527–530, Annecy, France, 1997.
- [Sch98] K. SCHILLING, H. ROTH, R. LIEB et H. STÜTZLE : Sensors to improve the safety for wheelchair users. Dans *Proceedings of the 3rd Technology for Inclusive Design and Equality Congress*, Helsinki, Finland, June 1998.

- [Sim95] R. SIMPSON, S.P. LEVINE, D.A. BELL, Y. KOREN, J. BORENSTEIN et L.A. JAROS : The navchair assistive navigation system. Dans *1995 International Joint Conference on Artificial Intelligence*, Summer 1995.
- [Sim02] R. SIMPSON, D. POIROT et F. BAXTER : The hephaestus smart wheelchair system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(2):122–125, June 2002.
- [Sou96] D. SOUTHWELL, A. BASU et J. REYD : Panoramic stereo. Dans *Proceedings of the IEEE International Conference on Pattern Recognition*, Vienne, Austria, August 1996.
- [Thr00a] S. THRUN, M. BEETZ, M. BENNEWITZ, W. BURGARD, A. CREMERS, F. DELLAERT, D. FOX, D. AHNEL, C. ROSENBERG, N. ROY, J. SCHULTE et D. SCHULZ : Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19, November 2000.
- [Thr00b] S. THRUN, D. FOX, W. BURGARD et F. DELLAERT : Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128:99–141, 2000.
- [Thr02] S. THRUN : Learning occupancy grids with forward sensor models. *IEEE Transaction on Autonomous Robots*, 2002.
- [Tra97] P.E. TRAHANIAS, M.I.A. LOURAKIS, A.A. ARGYROS et S.C. ORPHANOUDAKIS : Navigational support for robotic wheelchair platforms: an approach that combines vision and range sensors. Dans *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 20–25, Albuquerque, New Mexico, 1265-70, April 1997.
- [Ulr00] I. ULRICH et I. NOURBAKHSH : Appearance-based place recognition for topological localization. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1023–1029, San Francisco, CA, April 2000.
- [Wal87] S. WALTER : The sonar ring: Obstacle detection for a mobile robot. Dans *Proceedings of the International Conference on Robotics and Automation*, volume 3, pages 1574–1579, 1987.
- [Wei95] G. WEISS et E. von PUTTKAMER : A map based on laserscans without geometric interpretation. *Intelligent Autonomous Systems*, pages 403–407, 1995.
- [Wer00] S. WERNER, B. KREIG-BRÜCKNER et Th. HERRMANN : Modelling navigational knowledge by route graphs. *Lecture Notes in Artificial Intelligence*, 1849:295–316, 2000.
- [Wij98] O. WIJK, P. JENSFELT et H. CHRISTENSEN : Triangulation based fusion of ultrasonic sensor data. Dans *Proceedings of the International Conference on Robotics and Automation*, volume 4, pages 3419–3424, may 1998.
- [Wij01] O. WIJK : *Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization*. Phd thesis, Royal Institute of Technologie, SE-100 44 Stockholm, Sweden, 2001.
- [Yag94] Y. YAGI, S. KAWATO et S. TSUJI : Real time omnidirectional image sensor (copis) for vision-guided navigation. *IEEE Transaction on Robotics and Automation*, 10(1):11–22, February 1994.
- [Yan95] H.A. YANCO, A. HAZEL, A. PEACOCK, S. SMITH et H. WINTERMUTE : Initial report on wheel-lesley: a robotic wheelchair system. Dans *Proceedings of the Workshop on Developing AI Applications for the Disabled, held at the International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.
- [Zho03] Chao ZHOU et Tieniu Tan YUCHENG WEI : Mobile robot self-localization based on global visual appearance features. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003.

Cette page est laissée blanche intentionnellement

Résumé

Au cours de ces dix dernières années, la robotique mobile d'assistance aux personnes handicapées n'a cessé de se développer. Elle a pour objectif d'assister les personnes handicapées dans leurs déplacements et dans l'exécution de tâches de la vie courante. Dans ce contexte le projet VAHM vise à concevoir un véhicule destiné à évoluer dans un environnement d'intérieur et susceptible de réaliser certains déplacements de manière autonome. Dans cette optique le véhicule doit pouvoir se situer dans son espace de travail. Les travaux présentés dans ce mémoire concernent la méthode définie pour la localisation du fauteuil dans un environnement connu, fondée sur l'utilisation d'un odomètre et de capteurs extéroceptifs. Notre démarche consiste dans un premier temps à corriger la localisation estimée par l'odomètre en utilisant les mesures de distance fournies par les capteurs à ultrasons puis à ajuster l'orientation corrigée à l'aide des informations extraites d'une image. La méthode de localisation se décompose en deux modes.

Le premier mode de correction utilise une carte de l'environnement décrivant les frontières entre l'espace libre et l'espace occupé à l'aide de segments horizontaux. Les distances fournies par les capteurs à ultrasons permettent de construire un modèle local constitué de points qui seront mis en correspondance avec le modèle global afin d'ajuster la localisation du véhicule en cherchant à minimiser les distances entre les points et les segments selon le critère des moindres carrés.

Le second mode permet l'ajustement de l'orientation en utilisant une image fournie par une caméra. L'utilisation d'une image permet de percevoir des bords verticaux sous la forme de contours verticaux faciles à extraire à l'aide d'outils de traitements d'image. La procédure utilise un arbre d'appariement pour rechercher la correspondance entre les contours détectés et les verticales du modèle et en déduire l'orientation du véhicule. La taille de l'arbre est limitée par deux critères : l'ordre de grandeur de l'orientation (fournie par le recalage) et la cohérence des appariements.

Les méthodes proposées ont d'abord été testées sur des données simulées avant d'être évaluées à partir de données réelles. Les résultats de localisation font apparaître des erreurs en position de l'ordre de cinq centimètres et trois degrés correspondant aux objectifs fixés initialement.

Abstract

During these ten last years, the mobile robotics of assistance to the disabled have quickly developed. It with for objective to assist the people handicapped in their displacements and the execution of tasks of the everyday life. In this context the VAHM project aims to design a vehicle which could move in an autonomous way in an indoor environment. Accordingly the vehicle must be able to locate itself in its workspace. This memory deals with of the localization method of the wheelchair, into a known environment, using an odometer and exteroceptive sensors. Our initially step consists in correcting the localization estimated by the odometer by using the measurements of distance provided by the ultrasonics sensors then to adjust the orientation corrected using the information extracted from an image. The method of localization breaks up into two modes.

The first mode of correction uses a map of the environment describing the borders between free space and the occupied space using horizontal segments. The distances provided by the ultrasonics sensors make it possible to build a local model made up of point which will be put in correspondance with the global model in order to adjust the localization of the vehicle to minimize the distances between the points and the segments according to the criterion of least squares.

The second mode allows the adjustment of the orientation by using an image provided by a camera. The use of an image makes it possible to perceive vertical as vertical edges easily extracted using image processings tools. The precedure use an appariement tree to find the correspondance between the verticals of the model end the edges of the image to compute the orientation of the vehicle who is pruned by using two criterions: the approximation of the orientation (given by the first methode) and the coherence of the appariements.

The methods suggested have been initially tested on data simulated before validating their effectiveness on real data. The results of localization reveal errors in position of about five centimetres and three degrees corresponding to the objectives laid down initially.