HAL
open science

Cartographie de l'environnement et suivi simultané de cibles dynamiques par un robot mobile

Abdellatif Baba

To cite this version:

Abdellatif Baba. Cartographie de l'environnement et suivi simultané de cibles dynamiques par un robot mobile. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2007. Français. NNT : . tel-00260984

HAL Id: tel-00260984

https://theses.hal.science/tel-00260984

Submitted on 6 Mar 2008

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

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

UNIVERSITE TOULOUSE III – PAUL SABATIER
LAAS-CNRS

THESE

en vue de l'obtention du

DOCTORAT DE L'UNIVERSTIE DE TOULOUSE
délivré par l' Université Toulouse III – Paul Sabatier

***Discipline* : Robotique et Intelligence Artificielle**

présentée et soutenue

par

Abdellatif BABA

Le 18 Décembre 2007

Titre :

Cartographie de l'Environnement et Suivi Simultané de Cibles Dynamiques par Un Robot Mobile

Directeur de Thèse :

Raja CHATILA

JURY :

M. Rachid ALAMI	Président
M. Philippe BONNIFAIT	Rapporteur
M. Christian LAUGIER	Rapporteur
M. Roland CHAPUIS	Examinateur
M. Michel COURDESSES	Examinateur
M. El Mustapha MOUADDIB	Examinateur
M. Raja CHATILA	Directeur de Thèse

Remerciement

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS à Toulouse. Je tiens à remercier Mr. Malik Ghallab (l'ancien directeur du LAAS) de m'avoir accueilli dans le laboratoire en septembre 2003. Mes remerciements vont également aux directeurs successifs du pôle RIA, mon directeur de thèse Mr. Raja CHATILA, puis Mr. Rachid ALAMI de m'avoir permis d'effectuer cette thèse dans leur groupe de Recherche.

Je remercie chaleureusement mes rapporteurs Mr. Philippe BONNIFAIT et Mr. Christian LAUGIER pour le soin avec lequel ils ont lu ce manuscrit, ainsi que pour la qualité de leurs critiques. Mes remerciements vont également à Mr. Rachid ALAMI, Mr. Roland CHAPUIS, Mr. Michel COURDESSES et Mr. El Mustafa MOUADDIB qui ont eu la gentillesse de faire partie de mon jury de soutenance.

J'aimerais adresser ma profonde gratitude à Raja CHATLA pour avoir dirigé ma thèse et pour m'avoir fait profiter de ses compétences scientifiques et ses hauts caractères personnels.

Je tiens à remercier aussi Mme. Aurélie CLODIC qui m'a beaucoup aidé pendant plusieurs phases de ma thèse. Je remercie également Mme. Jackie SOM la secrétaire du Pôle RIA pour les efforts qu'elle a fait afin de faciliter et réussir mes participations aux conférences internationales, et le soutiens humain qu'elle m'a adressé à chaque fois j'ai eu un problème.

Je remercie les amis : (Ali, Anis, Ayman, Chadi, Fadi, Hassan, Maxime, Sébastien, Thierry G., Thomas, Wael, Wassime et William). Avec eux, j'ai passé de très bons moments pendant les quatre dernières années.

Je veux remercier très sincèrement ma mère, la grande femme de ma vie et je souhaite qu'elle soit fière de son fils. Finalement, je veux présenter cette thèse à l'âme de mon père. L'homme dont j'ai souhaité la présence à ma soutenance afin d'assister à la réalisation de son rêve, mais le destin en a décidé autrement. Injuste destin souvent plus rapide que nos espoirs.

Résumé :

L'avancement technologique de notre époque s'impose petit à petit dans notre vie pour résoudre tous nos problèmes professionnels ou même quotidiens. Aujourd'hui, il nous propose d'accepter de nouvelles entités artificielles intelligentes, pour qu'elles soient intégrées dans notre société humaine. Ces nouvelles machines qui essaient de copier nos capacités perceptuelles et parfois physiques sont des robots. L'existence des robots dans notre univers nous invite à savoir de plus en plus comment il faut interagir avec eux. En revanche, notre existence dans leur univers exige qu'ils soient capables de nous percevoir, nous comprendre et interagir avec nous. Pour qu'un robot assume toutes ces fonctions, il doit exploiter tous ses capteurs qui représentent pour lui une approximation de certains sens de l'être humain. Mais, à cause du bruit qui peut caractériser les mesures acquises par ses capteurs, il est nécessaire d'utiliser plusieurs méthodes de filtrage pour améliorer la qualité de ces mesures et ainsi du raisonnement qui en dépend. D'autre part la fusion des données des capteurs est utile et souvent utilisée pour compenser les points faibles de chaque capteur.

Une capacité essentielle pour qu'un robot mobile soit capable de comprendre son environnement est de pouvoir en construire la carte. Une carte précise et détaillée représente une source d'information indispensable pour réaliser une bonne localisation du robot. Mais pour élaborer une carte correcte, une localisation du robot dans son environnement est aussi importante. Les deux fonctions sont liées, ont la même priorité et doivent être achevées simultanément. Ce problème est déjà connu et référencé par le terme (SLAM : Simultaneous Localization And Mapping). Des entités mobiles dans la zone de perception du robot pendant qu'il construit sa propre carte peuvent provoquer de graves erreurs au niveau de sa localisation, et des inexactitudes dans la carte. Les traces des objets mobiles doivent être détectées et éliminées du modèle. Nous proposons d'adopter la représentation de l'environnement par une grille d'occupation qui décompose l'environnement en cellules indépendantes les unes des autres. A chaque cellule est assignée une valeur décrivant sa probabilité d'être occupée. Cette représentation fournit un modèle de l'environnement pouvant être mis à jour à une fréquence élevée ce qui présente l'intérêt de permettre la détection et le suivi d'éléments mobiles autour du robot.

Nous employons la grille elle-même pour effectuer la détection en cherchant les groupes de cellules qui viennent de changer leurs états d'une façon brutale entre deux balayages successifs du laser dans le champ de mesure. Le suivi des cibles dynamiques est ensuite effectué, où la technique de (Multiple Model Particle Filter) est utilisée avec le modèle choisi de la dynamique des objets. Une procédure qui attribue les mesures acquises aux cibles correspondantes est aussi présentée. Dans un environnement très peuplé par des personnes mobiles dans la zone de mesure du laser, la procédure d'attribution échoue parfois et spécialement pour des cibles ayant des trajectoires croisées. De plus, une erreur de localisation du robot pendant son mouvement, peut déclencher une étape de détection et de suivi pour des éléments qui sont en fait statiques dans le monde réel. Pour essayer d'améliorer la qualité de l'information sensorielle du robot et d'augmenter la fiabilité du suivi, nous utilisons une caméra catadioptrique dont les données seront fusionnées avec le laser.

Pour détecter des cibles mobiles en utilisant des images panoramiques, le fond de scène est tout d'abord extrait à partir de trois images successives (prises de la même position avec un délai de temps constant entre elles), en employant un (Temporal Median Operator). Une image binaire est calculée à partir d'une approche modifiée de flot optique où un nouveau terme qui compense les changements d'illumination est calculé et pris en compte. Deux stratégies qui déterminent les directions des cibles détectées à partir d'images déployées (sur une forme cylindrique) et d'images panoramiques brutes sont respectivement présentées et comparées. La synchronisation des deux capteurs (laser et caméra) est cruciale pour permettre la fusion de leurs données. Nous proposons une technique probabiliste pour coupler des mesures reçues par l'un et par l'autre capteur et pour les associer à un même événement.

Nous présentons enfin comment les différents algorithmes proposés sont intégrés dans l'architecture du robot.

Table des matières

1. Introduction Générale (Etat de l'art et Positionnement du Problème) 7

1.1. Introduction…… 7
1.2. La Cartographie……... 9
1.2.1. Modèle Géométrique…….. 10
1.2.2. Modèle Topologique…… 12
1.2.3. Carte d'Occupation……. 13
1.3. Cartographie et Suivi Simultanés…… 16
1.4. Détection Visuelle et Fusion Avec le Laser…… 18
1.5. Organisation Du Manuscrit……. 20

2. Cartographie Dans Un Environnement Statique (Grille d'Occupation) 23

2.1. Introduction……. 25
2.2. Robotique Probabiliste…… 25
2.2.1. Probabilité Conditionnelle et Loi de Bayes…… 25
2.2.2. L'hypothèse de Markov…….. 27
2.2.3. Le Filtre Bayesien……... 27
2.3. Le Télémètre Laser (SICK)……. 28
Le Modèle du Capteur……. 28
2.4. La Grille d'Occupation…… 29
2.5. Conclusion……... 32

3. Cartographie et Suivi Simultané 35

3.1. Introduction……. 37
3.2. Suivi d'objets mobiles ……...……. 37
Le modèle proposé pour la dynamique du système (MSDM)……. 39
3.3. Estimation non linéaire des processus markoviens……. 41
3.3.1. Positionnement du Problème…….. 41
3.3.1.1. Prédiction Markovienne……. 42
3.3.1.2. Correction Bayesienne……... 42
3.3.2. Le Filtrage……... 42
3.3.2.1. Le Filtre de Kalman……... 43
3.3.2.2. Le Filtre de Kalman Etendu……... 44
3.3.2.3. Le Filtrage Particulaire……... 45
3.3.2.4. MMPF (Multiple Model Particle Filter)…… 48
3.4. Simulations…….. 52
3.5. Expérimentations……. 54
3.5.1. Groupage de Cellules Voisines Occupées…….. 54
3.5.2. Détection de Cibles Mobiles……... 54
3.5.3. Suivi de Cible Mobile……. 56
Suivi Multi Cibles (Data Association)……... 57
3.6. Conclusion et Discussion……. 59

Table des matières

4. Détection Visuelle d'Objets Mobiles (En Utilisant une Caméra Catadioptrique) 61

4.1. Introduction……. 63
4.2. Caméra Catadioptrique ……... 63
4.2.1. Combinaison miroir parabolique / objectif télécentrique……... 64
4.2.2. Modèle de Projection…….. 65
4.2.3. Redressement Cylindrique d'une Image Panoramique …….. 66
4.3. Histogramme……... 68
4.4. Traitement d'image (Extraction du Fond d'image) …… 70
4.4.1. Matrice de Convolution……. 70
4.4.2. Gaussian Average Operator…… 70
4.4.3. Le Médian……... 72
4.4.3.1. Filtrer une Image en Utilisant (Median Operator)……. 73
4.4.3.2. Extraction du Fond d'image en Utilisant (Temporal Median Operator)……... 74
4.5. Détecter une Cible Dynamique en Utilisant le Flot Optique ……. 75
Une Approche Différentielle……... 75
4.6. Déterminer la Direction de Cible Détectée…….. 80
4.7. Déterminer les Directions de Plusieurs Cibles…… 82
4.8. Détection des Cibles Mobiles à Partir d'Images Panoramiques non Déployées (Images Brutes)…….. 83
4.9. Comparaison Entre les Deux Techniques Suivies Pour Déterminer les Directions des Cibles Détectées…….. 86
4.10. Conclusion…… 86

5. Fusion de Capteurs (Et Alignement des Données) 89

5.1. Introduction……. 91
5.2. Fusion de Capteurs…….. 91
5.3. Considérations Techniques…….. 93
5.4. Résultats Expérimentaux……. 94
5.5. Architecture Algorithmique……. 96
5.6. Conclusion……... 97

Conclusion Générale 99

Références 103

Table des figures

1.1. SLAM (Simultaneous Localization And Mapping)……..... **9**

1.2. Carte géométrique représente une salle de la Cité de l'Espace à Toulouse……..... **11**

1.3. Carte topologique……..... **13**

1.4. Grille d'occupation…… **13**

2.1. Description du télémètre laser…… **28**

2.2. Le modèle du capteur se présente comme une distribution normale dans une cellule…..... **29**

2.3. Dans chaque exemple ; les trois probabilités partielles se partagent entre eux une somme probabiliste égale à un. …….…… **31**

2.4. Un système du seuillage va traduire l'état d'une cellule selon sa probabilité d'occupation calculée comme une seule valeur entre zéro et un…..…… ……..... **32**

2.5. Exemple d'une grille d'occupation à 2D ……... **32**

3.1. Géométrie de (Cible – Observateur) illustrée en deux dimensions…….……... **38**

3.2. (a) l'approche de suivi comme elle est proposée dans [52]; (b) l'approche modifiée qui convient mieux à la caractéristique aléatoire dans le mouvement de l'être humain……... **39**

3.3. Le filtrage se compose comme d'habitude de deux phases (prédiction et correction)……... **41**

3.4. (a) cent particules dans chacun des trois nuages; (b) après le choix de la particule la plus probable à l'instant (t) , une nouvelle itération commence…… **48**

3.5. (A). L'observateur (le robot dans notre cas) se déplace suivant la ligne droite (en vert), pendant que la cible suit sa trajectoire; (B). Représente un zoom pour une zone de trajectoire …..... **52**

3.6. (A). Une trajectoire plus compliquée, (B). Suivi de deux cibles à la fois…….. **53**

3.7. Quelques exemples des groupes considérés……... **54**

3.8. Exemple d'une grille d'occupation avec un objet se déplaçant vers la gauche……..... **56**

3.9. Deux couches virtuelles : la première représente la grille d'occupation construite, alors que la seconde représente le niveau où le suivi des cibles mobiles s'effectue…… **56**

3.10. Suivi d'une personne se déplaçant à proximité du robot ……... **57**

3.11. Deux cibles suivies à proximité du robot, ……. **57**

3.12. Représentation de la figure (3.11) en 3D…… **58**

3.13. Suivi Multi Cible. …….…… **59**
3.14 Cartographie et suivi simultanés…… **60**

4.1. trois types de miroirs convexes. …… **64**

4.2. Le capteur omnidirectionnel utilisé……. **65**

4.3. Camera catadioptrique constituée d'un miroir parabolique et d'une caméra classique……. **65**

4.4. produire un point sur un cylindre à partir d'un point d'une image panoramique……... **67**

4.5. Le principe suivi pour faire l'interpolation…… **67**

4.6. A. histogramme pour une image ne couvre pas toute la plage de 256 niveaux d'intensité lumineuse, B. une image normalisée son histogramme couvre toute la plage……... **69**

4.7. La matrice de convolution…….. **70**

4.8. Une image filtrée en utilisant l'opérateur de Gaussien ……. **72**

4.9. Le médian……... **72**

4.10. Une image rectifiée en utilisant l'opérateur de médian…….. **73**

4.11. choisir le médian entre les pixels qui ont les mêmes coordonnées matricielles dans les images successives……... **74**

4.12. Trois images successives acquises à un délai de temps de 5 secondes, avec une image qui représente le fond de scène extrait en utilisant (temporal median operator)……... **74**

4.13. Une personne détectée dans une image binaire……... **78**

4.14. Une personne détectée (un changement de lumière entre le fond extrait et l'image courante)….. **79**

4.15. Une zone des colonnes intéressantes (sont côte à côte et elles ont des probabilités supérieurs)… **81**

4.16. Deux cibles détectées, avec un histogramme qui montre deux zones des colonnes intéressantes.. **82**

4.17. A partir de trois images panoramiques brutes prises aux moments différents, le fond de scène est extrait……. **83**

4.18. Une image binaire résultante de l'application de flux optique…… **84**

4.19. A. une image qui résulte après l'application d'une étape de normalisation, B. Histogramme 3D, C. image finale illustre deux cibles détectées, D. Histogramme 2D……... **86**

5.1. N registre pour N cibles…….. **92**

5.2. distribution Gaussienne, (modèle de pondération)……. **92**

5.3. A. La zone d'intersection de deux capteurs est illustrée en gris. B. le robot Dala équipé par un télémètre laser et une caméra catadioptrique…….. **94**

5.4. Une cible se déplaçant à proximité du robot…….. **95**

5.5. Deux cibles se déplaçant à proximité du robot……... **95**

5.6. Les composants de notre architecture algorithmique……. **97**

CHAPITRE I

Introduction Générale
(Etat de l'art et Positionnement du problème)

1.1. Introduction

Se localiser dans son environnement, c'est répondre à la question « où suis-je ? » posée en permanence par un robot mobile autonome. La version la plus difficile de cette question, communément référencée par le terme « robot kidnappé », est le problème de fournir une estimation de la position du robot sans estimation précédente, par rapport à une référence globale connue. Mais le problème général de la localisation consiste à estimer la position à partir d'une position précédente et de nouvelles observations de l'environnement toujours par rapport à une référence globale connue. Deux catégories de localisation peuvent être considérées, selon que les mesures fournies par les capteurs portent sur la position du repère lié au robot mobile par rapport à un autre référentiel connu (mesures extéroceptives) ou sur le déplacement relatif du robot entre deux instants successifs (mesures proprioceptives, par exemple odométriques).

Effectuer une meilleure localisation exige de détecter des amers connus qui permettent au robot de corriger l'estimation odométrique de sa position. L'odométrie est en effet fragile comme méthode de localisation du robot pour plusieurs raisons : patinage des roues non mesurable, inégalités de terrain, erreurs dues aux incertitudes de mesure et de calibration, etc. qui peuvent entraîner des erreurs cumulatives avec le temps. Les amers peuvent être connus (constituer une carte) ou être construits au fur et à mesure du déplacement. Il s'agit donc de maximiser la correspondance entre des cartes locales successivement construites avec l'évolution du mouvement du robot sur sa trajectoire s'il n'y a pas une base de données qui contient des cartes fournies et indexées a priori, ou pour un système qui dispose de cartes indexées, ce sera une maximisation de

corrélation entre les cartes construites et celles qui sont déjà indexées. Les techniques d'estimation (filtrage de Kalman, Kalman Etendu, filtrage particulaire, …etc.) sont à la base de cette combinaison de mesures extéroceptives et proprioceptives. Le télémètre laser, le sonar et la vision avec toute sa diversité (mono, stéréo, perspective, panoramique,...etc.), sont utilisés pour acquérir les données extéroceptives. Le GPS peut constituer aussi une source d'information externe pour l'estimation de la position et peut aussi être combiné avec des données proprioceptives.

La capacité d'un robot mobile à construire une carte pour décrire son environnement et s'y localiser représente donc une caractéristique essentielle pour garantir sa navigation. Afin de construire un modèle correct de l'environnement le robot doit être capable de distinguer entre deux catégories d'objets :

1. Objets Statiques : représentant les parties fixes de l'environnement qui doivent être incluses dans la carte élaborée.

2. Objets Mobiles : des personnes ou d'autres robots, ainsi que des objets à état variable (portes, fenêtres,…etc.).

D'autre part, la localisation du robot est importante pour effectuer une construction incrémentale de sa carte afin de mettre en correspondance les cartes locales (sinon, le problème du robot kidnappé, plus complexe, serait posé en permanence). D'où la nécessité du fonctionnement simultané de la localisation et de la cartographie qui est appelé le « SLAM » (Simultaneous Localization And Mapping) et qui nécessite la réalisation itérative des étapes suivantes:

1. Une phase d'acquisition de données à un instant (t_0), effectuée par un robot localisé à (x_0, y_0, θ_0), pour construire une partie de sa carte et choisir des amers qui vont être utiles pour les étapes suivantes de sa localisation, Figure (1.1).

2. Avec le déplacement du robot dans son environnement, les erreurs odométriques s'accumulent. A l'instant (t_1) le robot déclenche une nouvelle phase d'acquisition des données pour évaluer sa nouvelle position par rapport aux amers choisis dans l'étape précédente (s'ils sont toujours dans son champ de vision), c'est-à-dire effectuer une étape de correction au niveau de sa localisation, ce qui est indispensable pour construire une nouvelle partie de sa carte et mettre à jour la partie qui est déjà construite. La localisation et la mise à jour de la carte s'accomplissent simultanément.

3. Cette carte élaborée à partir de la fusion de plusieurs perceptions est plus précise et utile pour la localisation du robot dans les perceptions suivantes.

Les données acquises par les capteurs sont souvent bruitées, incomplètes, ou ne couvrent pas la totalité du champ de vision du robot à cause de masquages par exemple. L'utilisation de plusieurs sources sensorielles permet d'acquérir des données plus riches, parfois complémentaires, qu'il faudra fusionner.

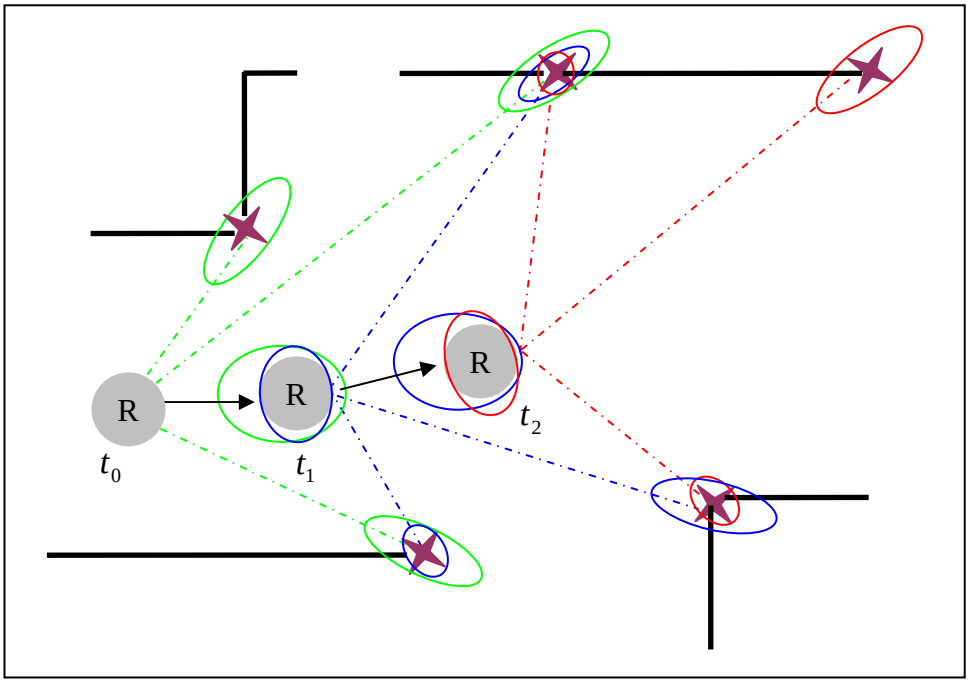


Figure (1.1), SLAM (Simultaneous Localization And Mapping). Trois instants de trajectoire du robot sont illustrés. Les étoiles sont les amers détectés par le robot. La vraisemblance de position d'un amer augmente à chaque fois qu'il est vu et distingué à partir d'une position différente. L'incertitude du positionnement du robot diminue à chaque fois qu'une étape de correction basée sur la perception des amers est effectuée.

1.2. La Cartographie

Un robot qui ne possède *a priori* aucune information sur l'environnement, dans lequel il doit se déplacer et agir, doit être capable de modéliser son environnement grâce à l'ensemble de ses capteurs. Ce modèle est indispensable pour sa localisation ainsi que pour planifier ses mouvements afin d'accomplir ses missions. Dans certaines applications robotiques, une carte de l'environnement peut être fournie par des sources extérieures comme les images aériennes ou satellites, mais dans le plupart des cas, ces données sont insuffisantes pour les applications qui ont besoin de perception précise dans une zone d'activité locale du robot. D'autre part, un robot autonome doit être capable de réagir à des modifications inattendues dans son environnement. La construction d'une telle carte devra être incrémentale, en fusionnant les perceptions successives acquises par les capteurs du robot au cours de son déplacement.

La difficulté d'un problème comme la cartographie vient généralement de plusieurs raisons [103] parmi lesquelles :

1. La taille de la carte construite : plus l'environnement décrit par une carte est grand, plus les traitements seront lourds et l'espace mémoire pour la stocker important.

2. Le bruit : si les mesures des capteurs et les mouvements effectués par les actionneurs du robot ne sont pas bruités, la cartographie devient un problème simple. Mais le bruit est toujours présent et son traitement est essentiel.

3. Ambiguïté perceptuelle : plus les différents endroits ont les mêmes aspects, plus il est difficile d'établir des bonnes correspondances entre ces endroits traversés avec le temps aux différents points. De manière plus générale, le problème de l'association de données doit être traité correctement.

4. « Boucler la boucle » : construire une carte dans une boucle fermée exige de corriger les erreurs odométriques qui s'accumulent, sous peine d'obtenir une carte incohérente. La carte construite incrémentalement peut se déformer si aucun amer n'est revu pendant le déplacement pour localiser le robot. La reconnaissance des amers vus à la position de départ et la construction de la carte par l'utilisation d'une technique du (SLAM) permet d'obtenir une carte cohérente.

La construction d'une carte exige d'autre part de choisir la nature des représentations qui vont la composer. Trois types de modèles de l'environnement sont utilisés en général [33] :

1.2.1 Modèle Géométrique

C'est un modèle qui contient des informations métriques sur l'environnement. C'est souvent une représentation de base utilisant l'ensemble des primitives 2D acquises par un balayage horizontal d'un télémètre laser, ou des primitives 3D comme les segments regroupés en facettes [94], ce qui permet de décrire des amers dédiés à la localisation et de représenter aussi l'espace navigable. Selon Rives et Devy [73], la sélection des amers devrait répondre à divers critères, en particulier : pouvoir discriminant, domaine de visibilité important, stabilité, invariance et bonne adaptation à la métrologie. Plusieurs types d'amers géométriques peuvent être distingués, comme les points, les coins, les segments, les polygones,….etc. [43]:

1. Points ou objets considérés comme ponctuels [75] [87] [76] [41] [24] : dans les cartes 2D, ils représentent notamment les objets verticaux tels que des poteaux ou des troncs d'arbres en environnement extérieur et des coins de murs en environnement intérieur.

2. Coins : par rapport aux points, ils incluent également une information d'orientation, et éventuellement un angle relatif entre les segments adjacents à ce coin.

3. Segments, qui correspondent généralement aux frontières d'obstacles [84] [75] [27] [49] [1] [30] [106] [24]: en général, ces segments sont représentés par leurs droites supports (de préférence en coordonnées polaires [98]) et portent éventuellement des informations sur la position géométrique de leurs extrémités.

4. Polygones de forme et taille quelconques correspondant aux frontières d'obstacles dans le plan horizontal [84] et plus généralement des polyosides (des polygones comportant des trous) [60].

5. Portions de courbes et notamment arcs de cercle [75] [112].

Les représentations géométriques sont souvent préférées pour modéliser des environnements structurés [83]. Mais elles sont aussi exploitées pour décrire des environnements naturels avec des terrains plutôt plats [12]. Des approches récursives [29, 37] sont développées pour segmenter l'environnement en utilisant le sonar. Des cartes géométriques qui incluent des coins et des surfaces partielles sont aussi construites [80, 35, 2]. Castellanos et al. [30,31] extraient des points, des coins, des segments et des surfaces à partir des informations sensorielles (laser et vision). Ayache et al. [6] ont aussi extrait des primitives 3D en utilisant la stéréovision. Janet et al. [63] ont utilisé les réseaux de neurones pour regrouper des données acquises par le sonar alors que Kwon et al. [71] ont utilisé une technique stochastique pour le faire. Delahoche et al. [38] ont utilisé la vision omnidirectionnelle pour extraire des segments verticaux (coins, portes,...etc.) dans un environnement structuré (d'intérieur).

Une carte géométrique peut être une représentation incluant tout type d'objets de n'importe quelle taille ou n'importe quelle forme. Les primitives sont construites par fusions successives de données élaborées à partir d'autres données métriques brutes (non traitées), comme les points directement issus des mesures, ce qui permet la réduction de l'espace mémoire nécessaire pour stocker la carte. Cependant, une carte géométrique est plus difficile à construire en comparaison avec une grille d'occupation, en particulier avec des données bruitées, à cause des approximations nécessaires. En outre, afin d'identifier des formes géométriques utiles pour la localisation du robot ou effectuer une reconnaissance d'objets par exemple, il faut mettre en œuvre des traitements de mise en forme sur l'ensemble des données acquises chaque fois que le robot effectue un raisonnement sur sa propre carte.

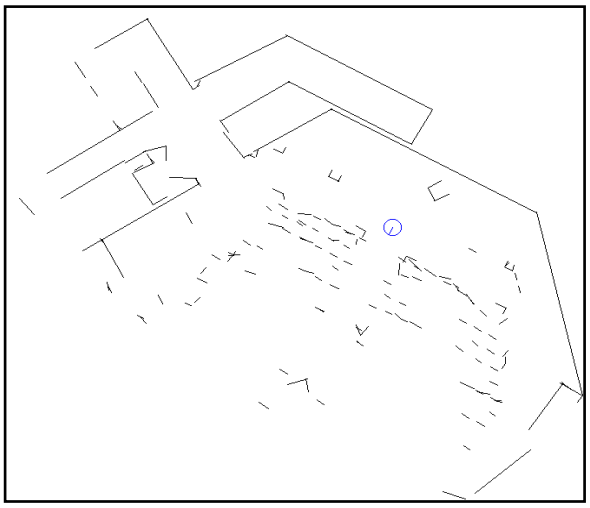


Figure (1.2), Carte géométrique représentant une salle de la Cité de l'Espace à Toulouse, [84].

1.2.2 Modèle Topologique

Il représente l'espace navigable du robot sous la forme d'un graphe qui contient toutes les informations sur la connexité et l'ordre des entités incluses. Autrement dit, c'est une représentation qui décrit les relations entre des éléments potentiels de l'environnement sans utiliser une référence globale absolue, ni une métrique en général. Nous pouvons distinguer deux méthodes pour la construire :

1. Elaborer un modèle géométrique puis chercher ses propriétés topologiques, comme celui proposé par Chatila et al. [33]. Où la définition du nœud est liée à une unité fonctionnelle ou topologique (corridor, bureau,…), alors que les connecteurs entre les nœuds représentent des portes ou des escaliers.

2. Construire une carte topologique qui s'appuie sur des informations qualitatives ou physiques comme des couleurs ou des textures qui permettent de distinguer la nature physique d'une partie du terrain traversé pour la considérer comme une entité discrète dans le graphe bâti, et de rajouter éventuellement des informations métriques locales (distances et directions) sur des entités déjà perçues [76]. Une telle carte topologique est construite par Brooks [25] en utilisant des informations visuelles, où les nœuds sont appelés (meadows) et les connecteurs sont des (freeways).

Les environnements structurés sont les plus facilement modélisés d'une façon topologique, car ils induisent une structure topologique avec moins d'ambiguïtés (par exemple le passage d'une pièce à un couloir). L'avantage majeur des représentations topologiques est d'éviter, le cas échéant, les approximations géométriques par des primitives, tout en rendant compte de la structure de l'environnement. Ceci peut permettre une localisation qualitative dans un lieu, effectuée à partir d'informations odométriques et une connaissance de la structure et de la nature des lieux la constituant [59].

A la structure topologique on peut aussi associer une signature identifiable ou un amer assez distinctif utile pour la localisation du robot. Pour cela, l'exploitation de données multi-sensorielles devient importante. Une boussole et 16 sonars sont utilisés chez Kuipers et al. [70] pour distinguer les différents nœuds. Chez Kortenkamp et al. [68] les sonars sont insuffisants pour effectuer l'identification de nœud, pour cela la vision (une caméra perspective) est proposée avec une limitation imposée sur les directions qui peuvent être prises par le robot (chaque situation angulaire du robot, est perpendiculaire à la précédente) ; un petit nombre d'images pour chaque situation sera ainsi suffisant pour garantir une bonne identification. Koenig et al. [67] ont proposé une carte topologique enrichie par des informations augmentées, où la longueur des corridors avec la largeur des connecteurs sont déjà fournies dans une base de données.

La carte topologique présente quelques avantages. Sa structure peut être exploitée par les algorithmes de planification de mouvement. C'est de plus une représentation très économe de l'espace mémoire (structure de graphe). Par contre, l'absence d'informations géométriques dans une

carte topologique pure peut être inadaptée dans certains cas où une localisation précise est nécessaire. Par ailleurs une ambiguïté de positionnement du robot peut apparaître si plusieurs parties de l'environnement (les nœuds) ont des apparences presque semblables.

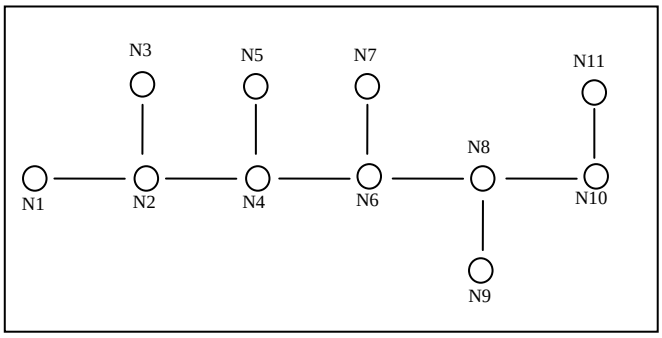


Figure (1.3), Carte topologique

1.2.3 Carte d'Occupation

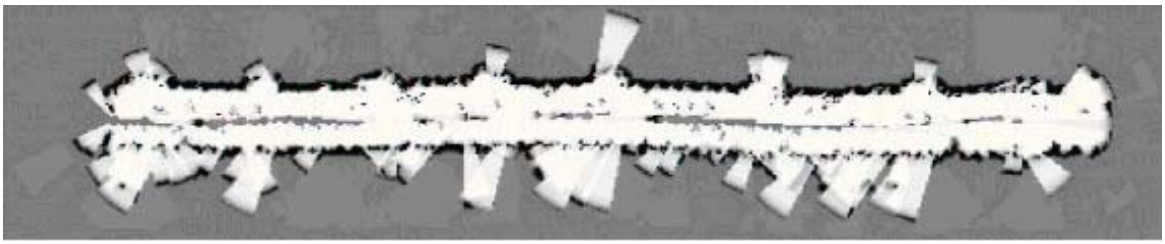


Figure (1.4), Grille d'occupation [104]

La grille d'occupation est une représentation de l'environnement qui décompose l'espace dans lequel un robot mobile évolue en un ensemble de cellules. La probabilité d'occupation de chaque cellule est une valeur estimée à partir des mesures fournies par les capteurs. Alors que beaucoup d'approches qui traitent du SLAM supposent que les environnements d'un robot mobile sont statiques (hypothèse peu compatible avec des robots de services interagissant avec des humains), la grille d'occupation représente un modèle capable de faire la mise à jour de l'environnement à une fréquence élevée, et permettant de réviser facilement les probabilités d'occupation, donc de suivre l'évolution de l'environnement autour du robot, ce qui est indispensable pour une meilleure réactivité. De plus, une grille d'occupation est capable de modéliser des environnements de forme quelconque, et ne cherche pas à approximer les données par des primitives qui peuvent être inadéquates. Elle est en général préférée pour les applications qui reposent sur la détection de l'espace navigable (espace libre) [72] ou pour l'évitement d'obstacles [23]. En revanche, elle souffre de la discrétisation par une grille qui induit des déformations et exige un espace de stockage important pour de grandes résolutions.

Cette méthode de cartographie était présentée au départ par Elfes [45] qui a proposé une représentation discrète 2D de l'espace navigable pour un robot mobile à l'aide des capteurs à

ultrasons. En 1992, Elfes [44] lui-même, a proposé une extension appelé (la grille d'inférence), où il a attribué à chaque cellule un vecteur qui contient plusieurs informations (sa probabilité d'occupation, sa classe d'appartenance,…etc.). D'autres cartes d'occupation ont aussi été construites en employant le sonar [110]. Toutes ces cartes utilisent un modèle de capteur permettant de calculer la probabilité d'occupation d'une cellule à partir des mesures : $p(m_{x,y} | z)$, ce qui est appelé le modèle inverse du capteur car il trace les mesures des capteurs pour revenir à leur cause (la présence de l'objet). Ce modèle du capteur est bien analysé par Thrun [104] qui a montré comment ce modèle peut provoquer des problèmes conflictuels dans une zone de visibilité commune entre deux sonars voisins sur le robot. Pour cela, il a proposé une autre approche basée sur le modèle direct : $p(z|m)$, qui décrit la probabilité de recevoir une mesure d'un objet présent sachant l'état de la carte (dans laquelle cet objet existe) a priori. L'algorithme EM (Expectation Maximisation) est utilisé pour calculer ces probabilités, puis il a appliqué une étape d'approximation de Laplacien pour le calcul de la carte finale soit trouvée. L'utilisation de l'algorithme (EM) peut rendre l'approche coûteuse en temps d'exécution.

D'autres travaux ont tenté d'employer la logique floue pour évaluer l'état d'occupation des cellules de la grille. Oriolo et al. [89] ont modélisé la cellule par un ensemble flou. Ainsi, chaque cellule peut exprimer à la fois deux états partiels (E = Vide et O = Occupée) et le degré d'appartenance entre eux se détermine en utilisant la théorie des possibilités. Dans cette approche chaque cellule peut avoir des données conflictuelles ($E \cap O$) fournies par les sonars, et sera considérée comme une cellule ni libre ni occupée. Eliminer l'ambiguïté de l'état de ces cellules exige de nouvelles données sensorielles, ce qui nécessite plus de navigation du robot (c'est-à-dire plus de temps). Pagac et al. [90] proposent d'utiliser la théorie de l'évidence, où à chaque cellule sera attribué trois valeurs de croyance $m(E)$, $m(O)$ et $m(E \cup O)$ telles que leur somme soit égale à un. Ensuite la règle de combinaison de Dempster-Shafer est appliquée pour mettre à jour la carte à partir des données des capteurs. Une théorie de l'évidence se caractérise normalement par deux seuils sur les probabilités, un inférieur (seuil de plausibilité) et un autre supérieur (seuil de croyance). Selon la largeur de l'intervalle entre les deux seuils, cette approche peut modéliser un état où il y a un manque de données (cellule à l'état inconnu). Une comparaison entre les deux dernières méthodes et celle qui utilise uniquement la loi de Bayes a été effectuée par Ribo et Pinz [93]. Ils ont appliqué les trois méthodes sur un robot dans un environnement structuré. Ils ont trouvé que l'approche qui utilise la logique floue a donné une carte plus robuste aux fausses alarmes et aux réflexions multiples. Par contre, elle fournit la carte la moins précise et la plus grande au niveau de l'espace occupé de mémoire, alors que l'approche probabiliste apparaît la plus rapide.

Tous les travaux discutés ci-dessus souffrent d'un problème commun, qui est le type du capteur utilisé (le sonar) à cause de ses défaillances nombreuses (fausses alarmes, détections manquées, difficulté de détection d'espaces étroits, données reflétées tardivement par une forme circulaire ou angulaire, ou même données diffusées face à une surface de verre,…etc.). Pour cela, d'autres approches proposent de fusionner les données acquises par le sonar avec celles d'autres capteurs. Stepan et al. [101] ont expérimenté la fusion (sonar, laser) et (sonar, camera perspective) dans un environnement statique d'intérieur pour construire une grille d'occupation robuste pour une meilleur planification. Pour eux, partager la même grille entre deux capteurs qui ont deux

fréquences différentes d'acquisition de données, ainsi que des qualités différentes des mesures, pose un problème de compatibilité entre les deux capteurs. Ils ont donc préféré produire deux grilles, une pour chaque capteur, puis les fusionner en utilisant la règle de Bayes. La fusion des deux grilles (une du sonar et une du laser) a donné une meilleure carte, même si sa résolution n'est pas meilleure qu'une carte construite seulement avec le laser. En fait, l'utilisation d'un seul capteur (le laser) plus précis peut être plus favorable car dans ce cas, il y n'y a pas besoin de créer deux grilles, donc le temps de calcul et l'espace mémoire sont plus faibles. D'autre part, le laser est plus adapté pour faire la détection et le suivi d'objets mobiles autour du robot dans un environnement dynamique que le sonar à cause de la mauvaise résolution des sonars. Le laser est aussi préféré pour les applications de type « SMAT » (Simultaneous Mapping And Tracking), que nous allons discuter dans un chapitre suivant, car il permet de produire une carte plus précise.

Une carte présentée par une grille d'occupation était la base chez Szabo [102] pour créer une procédure de navigation topologique. Birk et al. [22] ont proposé de construire une seule grille globale à partir de plusieurs grilles locales déjà élaborées par plusieurs robots. La conception d'une grille d'occupation temporelle capable de modéliser et de classifier différentes régions selon les propriétés temporelles de leur état d'occupation est proposée par Arbuckle [3]. Il exploite le principe d'Elfes [45] mais en calculant les cycles du temps d'occupation des cellules. Cependant, il faut un temps assez long pour que le robot apprenne un modèle assez fiable d'un environnement contenant des entités dont l'état change (portes, fenêtres, chaises …etc.). La conception de grille d'occupation peut être aussi utile dans une autre application [48] qui extrait une silhouette 3D en fusionnant plusieurs grilles élaborées sur la base de plusieurs images prises par différentes caméras (dans l'expérience citée, 8 caméras sont utilisées et une extraction d'une seule silhouette a duré 13 secondes). Zelinsky [111] a utilisé un modèle d'une représentation pyramidale dans laquelle l'espace est subdivisé récursivement en quatre régions carrées jusqu'à ce que chaque carré présente un statut d'occupation homogène ou atteigne la taille minimale autorisée. Cette méthode semble utile pour réduire la place mémoire réservée pour stocker une grille d'occupation. Mais elle exige plusieurs passages dans l'environnement décrit. Un modèle d'un environnement multi-résolution a été proposé par Arleo et al. [4]. Dans ce modèle, chaque obstacle aperçu et introduit dans la carte est partitionné en cellules, la résolution du découpage dépendant de la résolution des capteurs et de la forme de l'obstacle. A cause des modifications des paramètres de résolution de la carte (surtout dans le cas d'objets ayant des formes irrégulières), la mise à jour de ce modèle ne peut suivre une fréquence élevée.

Les exemples cités ci-dessus illustrent la grande gamme d'applications où une grille d'occupation peut être exploitable et utile pour représenter l'environnement en robotique mobile. Grâce à la simplicité de sa construction, la possibilité de la construire passivement (« en passant », sans contrainte particulière sur la trajectoire du robot), la facilité d'intégrer divers types de capteurs et son adaptabilité pour modéliser un environnement dynamique, nous avons choisi cette représentation comme méthode de cartographie pour effectuer le « SMAT ».

Dans le chapitre suivant, nous illustrons comment nous construisons la carte, dans une première phase dans un environnement statique. Le capteur utilisé est le télémètre laser (Grâce à sa résolution, il n'existe pas de mesures conflictuelles comme celles qui peuvent être fournies par les

sonars). La taille d'une cellule sera de (5cm x 5cm) compatible avec la taille du robot et celle des objets dans l'environnement (nous n'avons pas besoin de suivre une approche de subdivision récursive, car la taille d'une cellule est déjà petite, et nous n'avons pas de problème particulier d'espace mémoire). Une cellule peut exprimer trois états probabilistes : Libre, Inconnue ou Occupée, dont la somme est égale à un. Contrairement à la théorie de l'évidence, nous n'utilisons pas des seuils (un inférieur et autre supérieur). En revanche nous avons une étape de pondération à effectuer pour qu'à chaque cellule soit attribué un état parmi les trois possibles ; en suivant cette stratégie nous évitons l'ambiguïté classique qui est liée à la signification de la probabilité (0.5). Afin de garantir une mise à jour rapide, nous avons utilisé la règle de Bayes pour calculer les trois probabilités partielles de chaque cellule. Toutes les cellules dans notre grille sont initialisées à l'état inconnu.

1.3. Cartographie et Suivi Simultanés

Une perception locale peut être suffisante pour qu'un robot mobile se déplace librement dans un environnement peuplé en évitant les obstacles. Mais pour qu'un tel robot soit bien localisé dans son environnement (pas simplement avec des moyens odométrique), et pour qu'il planifie son mouvement en utilisant des points d'intérêt fixes, il doit avoir une carte correcte, nettoyée des traces des trajectoires d'objets mobiles, et construite pendant l'évolution du robot (une carte incrémentale). C'est par conséquent un problème du SLAM qui suppose que le robot est capable de représenter l'environnement de manière compatible avec le rythme d'évolution de celui-ci. Pour cette raison, le robot doit être capable de détecter et de faire le suivi d'objets mobiles. De plus, le suivi des trajectoires d'entités mobiles (être humains ou autres robots) représente un aspect indispensable pour un robot dans un environnement humain qui doit être interactif.

Un certain nombre de travaux récents sur la cartographie et la localisation dans un environnement peuplé ont abordé ce sujet. La plupart des approches qui font le suivi des cibles mobiles supposent qu'une carte décrivant l'environnement est connue a priori. Par exemple, Burgard et al. [16] effectuent une étape de mise à jour pour une carte statique au rythme de la réception des données sensorielles les plus récentes pour représenter les personnes pénétrant dans son champ de perception. Montemerlo et al. [81] présentent une approche qui réalise simultanément la localisation du robot avec le suivi des personnes à l'aide d'un filtre particulaires conditionnel. Cette approche est basée sur un modèle brownien du mouvement humain, ce qui ne semble pas si robuste. Arras et al. [5] ont réalisé une localisation basée sur des segments extraits, alors que les traces des personnes mobiles sont éliminées de la carte résultante. Fox et al. [47] utilisent une technique probabiliste pour filtrer les mesures qui ne correspondent pas au modèle d'environnement. Krishna et al. [69] ont proposé de faire la détection et le suivi en utilisant le sonar pour que le robot évite la collision avec les obstacles dynamiques qui peuvent apparaître dans sa trajectoire pendant sa navigation. Dans notre approche basée sur une grille d'occupation mise à jour en fonction des observations, il n'est pas nécessaire qu'un modèle de l'environnement soit connu à priori.

Wolf et al. [108] ont construit deux grilles d'occupation séparées; L'une décrit les parties statiques de l'environnement (ce qui est utile pour localiser le robot) et l'autre les cibles détectées.

Le laser est utilisé comme capteur, mais aucune prédiction du mouvement des cibles n'est proposée. D'autre part aucun détail sur les dimensions des cellules de grille n'est fourni, alors que le maintien de deux grilles de même taille simultanément double la mémoire nécessaire et nécessite de les maintenir en cohérence. D'autres techniques font du suivi de personnes, de la prédiction des mouvements, ou adaptent le comportement du robot selon l'information obtenue sur les personnes dans son voisinage [17, 66, 99], mais ne s'intéressent pas à améliorer le modèle de l'environnement en le nettoyant des traces des cibles suivies.

D'autres approches qui font le suivi de cible ne s'intéressent pas à construire une carte simultanément [74, 85]. Par exemple Le filtre de Kalman avec un balayage laser 2D est utilisé chez Bennewitz et al. [16], pour faire le suivi des personnes en utilisant des modèles du mouvement a priori connus. L'objectif est de créer une certaine connaissance chez le robot sur les modèles de mouvement pour permettre une prédiction robuste du comportement humain. Un télémètre laser est également utilisé [81] pour faire la détection de personnes se déplaçant par un robot en mouvement mais seulement une personne à la fois peut être suivie. Un filtre particulaire et le laser sont aussi utilisés [97] pour effectuer le suivi de cibles multiples à partir d'un robot en mouvement en utilisant une association de données conjointe (Joint Probabilistic Data Association - JPDA) mais avec l'adoption d'un modèle linéaire du mouvement humain. Notre approche propose de construire une carte de l'environnement simultanément avec le suivi des personnes mobiles multiples pour éviter de les inclure comme des parties fixes dans cette carte. Notre approche, ne s'appuie pas sur quelques modèles limités et prédéfinis de l'environnement comme elle ne suppose pas un modèle spécifique du mouvement des cibles.

Wang et al. [107] ont abordé notre objectif. Leur approche segmente les données acquises par le laser pour les distinguer et les exclure de la carte élaborée ; ils ont appliqué cette approche pour des environnements urbains (assistance à la conduite), avec une utilisation du GPS pour la localisation de la voiture porteuse du système. Coué et al. [36] ont présenté un système pour faire le suivi d'objets mobiles en utilisant le laser, l'objectif des travaux étant aussi d'aider le conducteur à éviter la collision avec des entités dynamiques, mais aucune carte de l'environnement n'était élaborée.

Récemment, Hashimoto et al. [57] ont présenté une méthode pour effectuer le suivi d'objets mobiles multiples détectés en utilisant le laser par l'intermédiaire d'une règle heuristique et d'une grille d'occupation; le suivi d'objets mobiles est basé sur un filtre de Kalman et un algorithme d'attribution. Dans leurs expériences la taille des cellules de la grille est de 30x30 cm^2, ce qui diminue la résolution de la carte construite. Pour distinguer un objet statique d'un objet dynamique ils utilisent 10 tirs successifs du laser. D'autre part, ils ont aussi besoin de plus de 13 balayages successifs pour prendre la décision de faire le suivi d'un objet mobile. Ces seuils empiriques rendent leur approche lourde en ce qui concerne le temps du traitement, et peu robuste. Hahnel, et al. proposent une approche probabiliste [54], dans laquelle ils ont filtré les données sensorielles fournies par la détection des personnes dynamiques, comme ils ont intégré les résultats du filtrage au processus de cartographie. Ils ont utilisé un algorithme EM (Expectation Maximization) [55] pour identifier les mesures, afin de les attribuer aux cibles correspondantes (data association). Cette dernière approche semble robuste, mais elle exige qu'une technique d'alignement soit appliquée sur

chaque couple de cartes successives pour détecter les objets mobiles. D'où, un temps de calcul induit par l'EM et l'alignement peut être important.

L'approche que nous utilisons n'a pas besoin des formes géométriques spécifiques. En plus, nous n'utilisons pas de seuillage, comme nous ne suivons pas une technique d'alignement pour des cartes entières successives. Notre représentation choisie (grille d'occupation) est assez flexible pour décrire les changements environnementaux. Le mouvement humain est détecté simultanément avec la mise à jour successive qui doit être effectuée seulement dans la zone de perception du robot pour économiser du temps du traitement de cette étape. Le suivi est basé sur une approche où la dynamique du système est modélisable par le (Système de Markov à saut) en combinaison avec une technique de filtrage particulaire.

1.4. Détection Visuelle et Fusion Avec le Laser

La détection puis le suivi des cibles mobiles peut être effectué à partir d'une séquence d'images prises par une ou plusieurs caméras dans de nombreuses applications. Dans cette étude nous ne citerons pas les travaux qui s'intéressent à effectuer le suivi des parties corporelles de l'homme (yeux, faces, mains,…etc.) ou ceux qui emploient différentes techniques de filtrage pour effectuer un suivi visuel à haut niveau sur la base d'informations traitées sur la cible (couleurs, motifs,…etc.). Notre objectif principal est de réaliser un détecteur visuel rapide qui compte sur les changements des caractéristiques lumineuses de pixels entre une image courante et autre représentant le fond de scène, afin de fusionner les données fournies par ce détecteur avec celles qui sont déjà fournies par un détecteur laser, pour améliorer les performances d'un système qui fait du SMAT.

Liu et al. [78] proposent un système omnidirectionnel pour détecter et suivre des cibles multiples, leur approche de détection est appliquée sur des images couleur RGB, les images étant déployées (images panoramiques projetées sur un cylindre). Le problème du changement de position d'objets appartenant au fond de la scène n'est pas considéré dans leur modèle, ni dans d'autres approches [62, 100, 109, 56]. Toutes les approches mentionnées sont aussi sensibles aux changements d'illumination. D'autres travaux [61, 64], emploient la fusion de couleurs et des segments détectés dans une image pour déterminer le fond de scène. Mais nos contraintes temporelles imposent l'utilisation des images en niveaux de gris. Veit et al. [105] proposent un algorithme qui exploite deux techniques (SIFT et KLT) pour isoler et segmenter les parties locales qui sont en mouvement dans une image. Malgré le bon niveau de détection présenté dans leurs expériences, cette approche est loin de fonctionner en temps réel car elle consomme une seconde pour accumuler les informations dont elle a besoin à partir de 25 images successives, et 3 secondes supplémentaires pour la segmentation à partir de 10 images. Le nombre d'images pour effectuer le processus peut varier entre 3 et 30 images en fonction de plusieurs contraintes expérimentales.

Une image décrivant une scène était présentée comme une carte d'occupation chez Beymer et Konolige [21], chaque image acquise est utilisée pour prédire la prochaine scène, les entités dynamiques sont distinguées en appliquant une étape de soustraction entre la scène courante et la scène prédite. Il ne semble pas y avoir de prise en compte des informations odométriques du robot

pour prédire la scène suivante, et une seule personne était détectée et suivie ; les expérimentations ont illustré un manque de robustesse. Une autre approche qui fait le suivi d'une seule cible a été proposée par Bai et al. [11], pour lesquels le modèle de cible doit être choisi a priori et manuellement. Une approche qui fait la segmentation et le suivi d'un objet mobile est proposée par Ding et al. [39], où la cible doit être aussi choisie a priori et entourée par une fenêtre pour la segmenter et prédire sa prochaine position dans une image suivante à partir de sa position dans deux images précédentes. Les deux dernières approches ont besoin d'une étape d'initialisation manuelle à chaque fois pour garantir leur fonctionnement, ce qui les rend moins pratiques ou pas vraiment applicables en ligne.

Quelques autres expériences ont employé plusieurs caméras, parfois [34] fixées dans différentes positions choisies dans l'environnement (pas sur un robot) pour bien identifier les cibles et suivre leurs trajectoires même dans le cas où elles se croisent. Dans [91], deux caméras pan & tilt sont embarquées dans la tête du robot pour détecter et suivre des formes spécifiques comme des boules rouges. Ces travaux ne donnent pas une solution pratique qui peut traiter le problème de détection ou du suivi sans exiger des contraintes parfois très spécifiques. Des simulations sont présentées dans [79] pour effectuer le suivi d'un objet mobile dans une grille d'occupation connue et fournie a priori. Une version modifiée de distribution de Gaussiennes basée sur une transformation de distance est introduite pour modéliser la position a posteriori de la cible détectée et suivie. Cette dernière approche s'intéresse plutôt à planifier le mouvement du robot pour garder toujours la cible dans son champ de vision. Puisque le capteur visuel est supposé virtuel, aucune méthode du traitement d'image n'est proposée.

Le détecteur visuel doit être capable de fonctionner dans le minimum de temps possible. Pour cela nous avons traité nos images en niveaux de gris avec une résolution basse. Dans notre système, nous avons profité du champ de vision panoramique (360°) d'une caméra catadioptrique pour effectuer la détection de plusieurs cibles simultanément dans la scène, sans contraintes particulières sur les cibles détectées (ni leurs comportements, ni leurs formes ni leurs apparences). Finalement, le changement d'illumination est pris en compte dans l'approche que nous proposons, comme nous le verrons dans le quatrième chapitre.

L'importance de fusionner les données de plusieurs sources sensorielles vient, comme nous l'avons déjà dit, du fait qu'une mesure ou une transformation de données est souvent bruitée, ce qui peut influencer négativement la qualité de l'information finale. De plus, dans certains cas, si un seul capteur devient incapable de fournir au robot une information riche et utile à cause de ses propres limitations ou de quelques difficultés environnementales, la fusion de plusieurs capteurs complémentaires est une solution efficace. Chez Chakravarty et al. [32], les données télémétriques acquises par le laser sont fusionnées avec celles qui sont fournies par un système de vision panoramique pour faire le suivi multi-cibles à partir d'un robot stationnaire. Le fonctionnement en temps réel exige une faible résolution de l'image couleur traitée. Mais malgré cela, une personne détectée en appliquant l'algorithme de soustraction Gaussien du fond doit être très proche de la caméra embarquée sur le robot pour qu'elle soit détectée, et qu'elle soit suivie. Dans le même objectif, Bellotto et al. [15] proposent un système qui fait l'intégration d'informations fournies par le laser et une camera PTZ, les jambes de la cible sont détectées par le laser en utilisant des modèles a

priori ce qui peut représenter une limitation importante et une raison de fragilité du système, alors que le visage de la cible est détecté par le capteur visuel. La fusion des données des deux sources sensorielles est effectuée en utilisant le « unscented Kalman filter ». Chez Reulke et al. [92], une caméra perspective équipée avec trois capteurs CCD et une tête qui tourne de 360° autour de son axe vertical pour fournir une image panoramique couleur à une résolution élevée, est fusionnée avec un télémètre laser 3D pour garantir un bon positionnement et orientation d'un véhicule se déplaçant dans un environnement urbain et sans aucune donnée GPS.

La technique de fusion que nous proposons dans ce mémoire fonctionne avec un robot stationnaire, car l'extraction du fond de scène et la détection des cibles à partir de caméra catadioptrique exigent que les images successives soient prises de la même position. Mais, quand le robot commence à avoir un mouvement continu, il peut compter seulement sur son détecteur laser qui n'impose pas des modèles prédéfinis pour les cibles. Les jambes des personnes détectées correspondent à des mesures sur des cellules de la grille, cellules qui sont groupées de manière à les associer à une seule cible. Le système fait une mise à jour continue dans la zone de perception du robot pour chaque balayage laser 2D, avec une détection du changement d'état des groupes de cellules.

1.5. Organisation Du Manuscrit

La construction incrémentale d'une grille d'occupation décrivant un environnement dynamique d'un robot mobile qui s'y déplace est le sujet principal de ce manuscrit. Les techniques de détection ou de suivi d'objets mobiles effectués en utilisant différents types de capteurs et en appliquant différentes méthodes, sont basées sur cette représentation et permettent d'améliorer la perception et l'opération d'un robot autonome interactif.

Après ce chapitre introductif, les chapitres suivants s'organisent comme suit :

Chapitre 2 : Ce chapitre introduit quelques notions probabilistes avec un rappel du filtrage Bayesien. Puis, il se focalise sur l'élaboration d'une grille d'occupation en utilisant le télémètre laser pour décrire un environnement statique d'un robot mobile.

Chapitre 3 : Une approche de cartographie et de suivi simultanés dans un environnement dynamique est proposée dans ce chapitre. Le modèle dynamique du système (robot-cible) est détaillé. Après le développement de la technique de filtrage utilisée (filtrage particulaire), ce chapitre se termine en illustrant un résultat avec les techniques et les conditions expérimentales que nous avons utilisées.

Chapitre 4 : Ce chapitre présente une étude de détection d'objets mobiles en utilisant un capteur visuel (caméra catadioptrique). Il commence par une brève présentation de la caméra catadioptrique, du modèle de projection et du redressement cylindrique d'une image panoramique. Après une présentation de quelques concepts essentiels pour les travaux effectués dans ce chapitre, une méthode d'extraction du fond de scène en utilisant le médian temporel est proposée. Ensuite,

une approche modifiée de flot optique, qui prend en compte les changements d'illumination des pixels dans deux images successives prises à deux instants différents, est discutée. Enfin, nous comparons deux techniques différentes qui peuvent isoler et grouper les pixels qui appartiennent aux cibles mobiles détectées.

Chapitre 5 : L'alignement des données des capteurs utilisés (laser et caméra) est effectué en utilisant une technique probabiliste présentée dans ce chapitre. L'architecture algorithmique qui montre les liens entre toutes les fonctions qui réalisent les différentes fonctions discutées dans tous les chapitres précédents est aussi décrite et discutée.

Enfin, une conclusion générale avec une discussion des perspectives clôt le mémoire.

CHAPITRE II

Cartographie Dans Un Environnement Statique (Grille d'Occupation)

2.1. Introduction

2.2. Robotique Probabiliste

2.2.1. Probabilité Conditionnelle et Loi de Bayes
2.2.2. L'hypothèse de Markov
2.2.3. Le Filtre Bayesien

2.3. Le Télémètre Laser (SICK)
Le Modèle du Capteur

2.4. La Grille d'Occupation

2.5. Conclusion

2.1. Introduction

Dans le chapitre précédent, nous avons discuté l'importance de la cartographie pour réussir la localisation du robot et pour lui garantir une bonne navigation. Trois types de cartographie ont été présentés, et parmi les trois types de cartes discutés, nous avons sélectionné la grille d'occupation, car elle représente le modèle le plus adéquat pour réaliser notre objectif de faire la cartographie et le suivi simultané, qui sera détaillé dans le chapitre suivant.

Ce chapitre introduit les concepts de base et les outils que nous avons utilisés pour construire une grille d'occupation. Nous commençons par un bref rappel des règles probabilistes que l'élaboration d'une telle carte nécessite, puis nous exposons les détails techniques de l'élaboration de la carte (le modèle du capteur, le filtre utilisé, les composants du vecteur d'état d'une cellule …etc.). Enfin, nous expliquons les considérations pratiques pour produire une grille décrivant une partie statique de l'environnement. A la fin du chapitre nous présentons une conclusion générale.

2.2. Robotique Probabiliste

S. Thrun et al ont intitulé leur livre sur la robotique « Probabilistic Robotics » [103], marquant en cela le rôle central des incertitudes et du raisonnement probabiliste pour les traiter en robotique. Grâce à leur capacité à traiter les différentes incertitudes qui caractérisent normalement les mesures acquises ou les actions effectuées par un robot mobile et interactif dans un monde réel, les approches probabilistes sont en effet devenues une solution efficace, essentielle et très fréquente dans beaucoup d'études robotiques. Notre approche s'inscrit dans ce cadre riche des travaux dans ce domaine. Nous présentons donc ici les règles probabilistes utiles pour traiter les incertitudes qui se manifestent généralement dans ce type d'applications à cause de plusieurs raisons :

1. L'incomplétude des données et des connaissances sur l'environnement, ainsi que les approximations algorithmiques qui dénaturent ces connaissances.
2. Le bruit et l'imprécision des mesures qui peut affecter l'ensemble du système (capteurs, effecteurs).
3. Les défaillances, de toute nature, les erreurs algorithmiques qui peuvent générer des anomalies dans le fonctionnement du robot.

2.2.1. Probabilité Conditionnelle et Loi de Bayes

Soient, x et y deux événements :

1. La probabilité d'une disjonction :

$$P(x \cup y) = P(x) + P(y) - P(x \cap y) \qquad \textbf{2. 1}$$

Dans le cas où x et y sont indépendantes, cette dernière règle s'écrit comme suit :

$$P(x \cup y) = P(x) + P(y) \qquad \textbf{2. 2}$$

2. La probabilité conditionnelle :

$$P(x \mid y) = \frac{P(x \cap y)}{P(y)} \qquad \textbf{2. 3}$$

D'où : $P(x \cap y) = P(x \mid y)P(y) = P(y \mid x)P(x)$ **2. 4**

Le terme $P(x)$ est la probabilité de x a priori. Le terme $P(x \mid y)$ est appelé la probabilité à posteriori de x sous condition de y

Pour des événements indépendants :

$$P(x \mid y) = P(x) \Rightarrow P(x \cap y) = P(x)P(y) \qquad \textbf{2. 5}$$

En plus : $P(\bigcap_{i=0}^{n} x_i) = P(x_0)P(x_1 \mid x_0)P(x_2 \mid x_1 \cap x_0).....P(x_n \mid \bigcap_{i=0}^{n-1} x_i)$ **2. 6**

3. On appelle distribution marginale de y par rapport à x, la distribution de probabilité suivante:

$$P(y) = \sum_x P(y \mid x)P(x) \qquad \textbf{2. 7}$$

4. Le théorème de Bayes est utilisé pour déterminer ou mettre à jour l'estimation d'une probabilité d'un évènement à partir d'une ou de plusieurs observations. Ce théorème énonce en conséquence une probabilité conditionnelle pour évaluer la valeur de x en sachant la valeur de y si elle est connue :

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)} \qquad \textbf{2. 8}$$

Le terme $P(y \mid x)$ pour un y connu est appelé la fonction de vraisemblance de x. De même, le terme $P(y) \neq 0$ est appelé probabilité a priori de y, qui peut être remplacé par la probabilité marginale (2.7), pour nous donner :

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{\sum_x P(y \mid x)P(x)} \qquad \textbf{2. 9}$$

Le dénominateur exprime toutes les réalisations possibles de y et peut être considéré comme une constante de normalisation η :

$$P(x \mid y) = \eta\, P(y \mid x)P(x) \qquad \textbf{2. 10}$$

2.2.2. L'hypothèse de Markov

L'hypothèse de Markov propose que l'état présent ne dépende que d'un nombre fini d'états passés ; ce qui s'exprime comme suit :

$$P(x_t \mid x_{0:t-1}) = P(x_t \mid x_{t-1}, x_{t-2}, ..., x_{t-n}) \qquad \textbf{2. 11}$$

Au premier ordre, cette hypothèse peut se traduire par la formulation suivante où l'état présent ne dépend que de l'état précédent :

$$P(x_t \mid x_{0:t-1}) = P(x_t \mid x_{t-1}) \qquad \textbf{2. 12}$$

Cette hypothèse permet en particulier de réduire la complexité et le temps du calcul.

2.2.3. Le Filtre Bayesien

Le filtre bayesien est un estimateur récurrent d'état. Dans le cas gaussien, la probabilité de l'état a posteriori est représentée comme une distribution normale autour de l'état vrai décrite par l'équation suivante :

$$P(x) = \det(2\pi\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \qquad \textbf{2. 13}$$

x : Est un vecteur d'état, μ est un vecteur de moyenne (qui a une dimension égale à celle de l'état), Σ est une matrice de covariance symétrique et semi positive.

Le filtre de Bayes représente l'algorithme le plus général pour calculer une connaissance du robot de l'état de son environnement à partir de différentes mesures et commandes, par exemple, déterminer en ligne la localisation du robot dans son environnement en utilisant ses propres moyens (ses capteurs). Cette connaissance se représente par l'ensemble des distributions conditionnelles probabilistes sur les données disponibles, chaque densité probabiliste décrivant une hypothèse possible dans l'espace d'état.

$$p(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) \qquad \textbf{2. 14}$$

x_t représente l'état à un instant donné (t), ($z_{1:t}, u_{1:t}$) sont respectivement l'ensemble des mesures et des commandes sur tous les instants passés jusqu'à l'instant courant.

L'algorithme du filtre de Bayes respecte l'hypothèse de Markov et se compose de deux étapes fondamentales qui se répètent:

1. Une étape de prédiction :

$$\hat{p}(x_t) = \Sigma_{x_{t-1}} p(x_t \mid u_t, x_{t-1})\ p(x_{t-1}) \qquad \textbf{2. 15}$$

$p(x_{t-1})$ décrit une probabilité de l'état x_{t-1} a priori.

$p(x_t \mid u_t, x_{t-1})$ représente la probabilité pour qu'une cause (ici une commande : u_t) qui peut arriver à l'instant (t), soit capable de provoquer une transition d'état.

2. Une étape de correction :

$$p(x_t \mid z_t) = \eta \; p(z_t \mid x_t) \; \hat{p}(x_t) \qquad \textbf{2. 16}$$

$p(z_t \mid x_t)$ décrit la probabilité de recevoir une mesure z_t en sachant que l'état courant x_t, et $\hat{p}(x_t)$ résulte de la première étape.

2.3. Le Télémètre Laser (SICK)

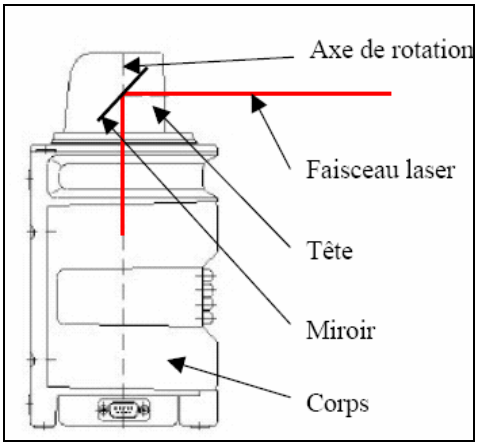


Figure (2.1). Description du télémètre laser

Le modèle d'un capteur observant l'environnement du robot doit permettre de répondre aux questions suivantes :

1. Prédiction : quelles sont les informations qui peuvent être fournies quand le capteur observe à partir d'une position donnée une scène connue?

2. Explication : à partir des informations fournies par le capteur, quelle connaissance peut-on déduire sur l'environnement ?

Le principe du télémètre laser est de mesurer le temps de vol du rayon laser "(temps d'aller + temps de retour)/2". Il est constitué d'un corps et d'une tête. Pour effectuer les mesures, la tête se met en rotation autour de son axe à des fréquences allant de 5 à 20 Hz. Le faisceau laser est dévié par un miroir qui en tournant avec la tête permet au laser de faire un balayage sur 360°. Le capteur communique avec l'extérieur à l'aide d'une liaison série RS-232. Grâce à sa haute précision par rapport au sonar, et à sa haute vitesse, ce capteur est très fréquent pour les applications robotiques.

Le Modèle du Capteur

Pendant son balayage, le laser fournit la distance et la direction de chaque point mesuré. Le mouvement mécanique du miroir qui est indispensable pour la fonctionnalité du capteur peut produire quelques erreurs de mesure. Pour cela le modèle du capteur souhaité doit prendre en compte l'incertitude pour les deux valeurs mesurées (ρ et θ), ce qui est exprimé par la distribution gaussienne suivante :

$$\eta \exp\left[-\frac{1}{2}\left(\frac{(\rho-d)^2}{\sigma_d^2}+\frac{(\theta-\Theta)^2}{\sigma_\Theta^2}\right)\right] \qquad \textbf{2. 17}$$

η : est une constante de normalisation, ρ et θ : sont respectivement la distance et la direction mesurées dans le référentiel local du robot. d et Θ : sont respectivement la distance et la direction calculables entre le centre de cellule touchée par le laser et la référence où se localise le robot. σ_d et σ_Θ sont respectivement l'écart type radial et angulaire du capteur.

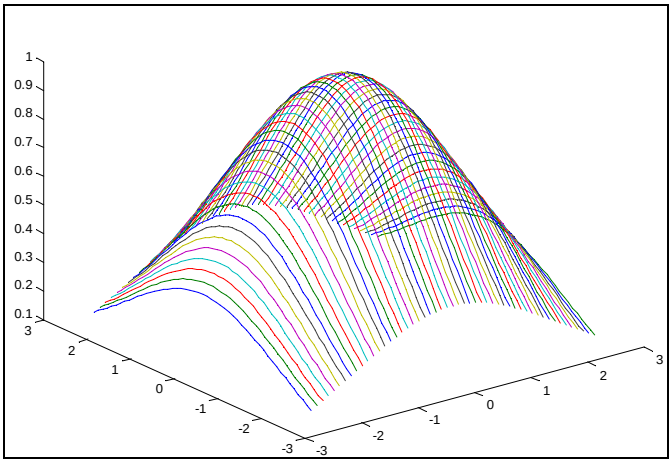


Figure (2.2), Le modèle du capteur se présente comme une distribution normale dans une cellule.

2.4. La Grille d'Occupation

Parmi les trois modèles représentatifs que nous avons discutés dans le dernier chapitre, nous avons adopté la grille d'occupation pour construire notre carte. L'utilisation de cette représentation décompose l'environnement modélisé en cellules, et à chaque cellule est assigné un vecteur d'état X_t qui se compose des variables suivants :

1. Trois états partiels ($x^{lib}_t, x^{inc}_t, x^{occ}_t$) représentent trois croyances (Libre, Inconnu et Occupé). La croyance de l'état de cellule sera caractérisée par celle qui est dominante parmi eux. Leurs valeurs sont calculées en se basant sur des règles probabilistes, et leur somme est égale à 1.
2. Les coordonnées de la cellule (cartésiennes et angulaires).
3. Une variable booléenne qui exprime l'état de visibilité d'une cellule par rapport au robot (masquée ou visible).
4. Deux variables décrivant l'état du groupage d'une cellule dans son voisinage et le groupe auquel elle appartient.
5. Des variables booléennes utiles pour donner des indications sur la cellule (déjà perçue par le laser, occupée à un instant donné par le robot lui-même, appartenance à un groupe mobile).

L'utilité de l'ensemble des variables citées ci-dessus sera clarifiée dans la suite dans ce chapitre. Dans notre approche, la détermination des composantes du vecteur d'état pour une cellule donnée ne prend pas en compte les valeurs des variables qui appartiennent aux vecteurs d'état des cellules voisines, ce qui est bien adapté à l'utilisation du laser dont le faisceau est suffisamment fin pour que l'hypothèse que la mesure appartienne à une seule cellule (de 5cmx5cm) soit valide. Par conséquent, notre technique du groupage (qui sera expliquée plus loin dans ce chapitre) n'induit pas de corrélation entre l'état d'occupation d'une cellule à partir de l'état de ses voisines.

Pour calculer et mettre à jour les valeurs des trois états partiels dans chaque cellule de la grille élaborée, nous avons utilisé le filtre bayésien discret (Discrete Bayes Filter DBF) qui est normalement applicable aux problèmes avec des espaces d'état finis. Contrairement à l'application classique de ce filtre qui exige de prendre en compte la mesure seulement dans l'étape de correction, nous avons employé les mesures reçues du capteur pour produire une certaine prédiction du changement de l'état de la cellule. Dans ce cas, la mesure est le seul porteur de l'information de l'environnement. Les deux étapes sont alors appliquées au même moment selon la loi de Bayes, en remplaçant l'équation (2.18) dans l'équation (2.19). Donc, soit z_t une mesure reçue à un instant donné (t), les deux étapes de ce filtre peuvent être écrites comme suit :

1. Etape de prédiction :

$$\begin{aligned}\hat{p}(x_t^{occ}) = {} & p(x_t^{occ} \mid z_t, x_{t-1}^{occ}) * p(x_{t-1}^{occ}) + \\ & p(x_t^{occ} \mid z_t, x_{t-1}^{lib}) * p(x_{t-1}^{lib}) + \\ & p(x_t^{occ} \mid z_t, x_{t-1}^{inc}) * p(x_{t-1}^{inc})\end{aligned} \qquad \textbf{2. 18}$$

Cette étape prédit la probabilité partielle d'occupation d'une cellule à l'instant (t) comme une somme de produits de deux probabilités. La première $p(x_{t-1}^{j})$ est la probabilité à l'instant ($t-1$) de chacun des trois états partiels d'une cellule, où : ($j = occ,\ lib\ \ et\ \ inc$). La deuxième $p(x_t^{occ} \mid z_t, x_{t-1}^{j})$ est la probabilité qu'une mesure z_t arrivant à l'instant (t) puisse produire une transition d'état de cellule de l'instant ($t-1$) à l'instant suivant ; les valeurs pouvant être prises par cette dernière probabilité sont pré-choisies selon les considérations pratiques de l'expérience.

2. Etape de correction :

$$p(x_t^{occ}) = p(x_t^{occ} \mid z_t) = p(z_t \mid x_t^{occ}) * \hat{p}(x_t^{occ}) * p_{dist} \qquad \textbf{2. 19}$$

La deuxième étape est liée au fait que les mesures sont bruitées, et ce bruit est exprimé par le terme : $p(z_t \mid x_t^{occ})$ qui décrit la probabilité d'avoir une mesure à l'instant (t) si la cellule mesurée était occupée au même moment. Cette probabilité se calcule en utilisant le modèle du capteur laser représenté par l'équation (2.17) et illustré dans la figure (2.2). La probabilité d'occupation d'une cellule est aussi basée sur les caractéristiques du capteur télémétrique liées à la distance de mesure qui expriment que plus une cellule occupée est proche du robot, plus sa probabilité d'occupation est grande. C'est pourquoi, nous considérons un facteur d'affaiblissement fondé sur la distance:

$$P_{dist} = \frac{(distMax - distMeasur)}{distMax} \qquad \textbf{2. 20}$$

$distMax$ est la distance maximale qui peut être mesurée, et $distMeasur$ est la distance mesurée entre une cellule et le robot.

De même, pour calculer la probabilité de l'état partiel (libre) d'une cellule, les deux étapes précédentes se répètent:

$$\begin{aligned}\hat{p}(x_t^{lib}) = {} & p(x_t^{lib} \mid z_t, x_{t-1}^{occ}) * p(x_{t-1}^{occ}) + \\ & p(x_t^{lib} \mid z_t, x_{t-1}^{lib}) * p(x_{t-1}^{lib}) + \\ & p(x_t^{lib} \mid z_t, x_{t-1}^{inc}) * p(x_{t-1}^{inc})\end{aligned} \qquad \textbf{2. 21}$$

$$p(x_t^{lib}) = p(x_t^{lib} \mid z_t) = p(z_t \mid x_t^{lib}) * \hat{p}(x_t^{lib}) \qquad \textbf{2. 22}$$

L'état inconnu représente l'état d'une cellule cachée, et puisque la somme des trois états partiels d'une cellule est égale à un, nous avons:

$$p(x_t^{inc}) = 1 - (p(x_t^{lib}) + p(x_t^{occ})) \qquad \textbf{2. 23}$$

Toutes les cellules dans la grille sont initialisées à l'état inconnu a priori.

$$p(x_{t=0}^{lib}) = 0\,,\ p(x_{t=0}^{occ}) = 0 \text{ et } p(x_{t=0}^{inc}) = 1 \qquad \textbf{2. 24}$$

Après le calcul des trois probabilités partielles, une étape de comparaison entre elles doit être effectuée pour décider de l'identité de l'état partiel dominant qui va caractériser une cellule donnée à un instant donné. La figure (2.3) présente trois cas différents, afin d'expliquer comment la probabilité d'occupation d'une cellule peut être partagée. Par exemple, dans la situation illustrée dans la figure (2.3.A) nous avons une cellule jugée avoir l'état inconnu car c'est l'état partiel dominant, et non parce que la probabilité $p(x_t^{inc}) = 0.5$. Pareillement pour la deuxième situation (2.3.B) où la cellule sera jugée occupée car c'est l'état dominant. Au fur et à mesure que le robot se déplace dans son environnement, les cellules qui tombent dans son champ de perception changent leur état en fonction de l'état dominant issu du calcul des équations ci-dessus.

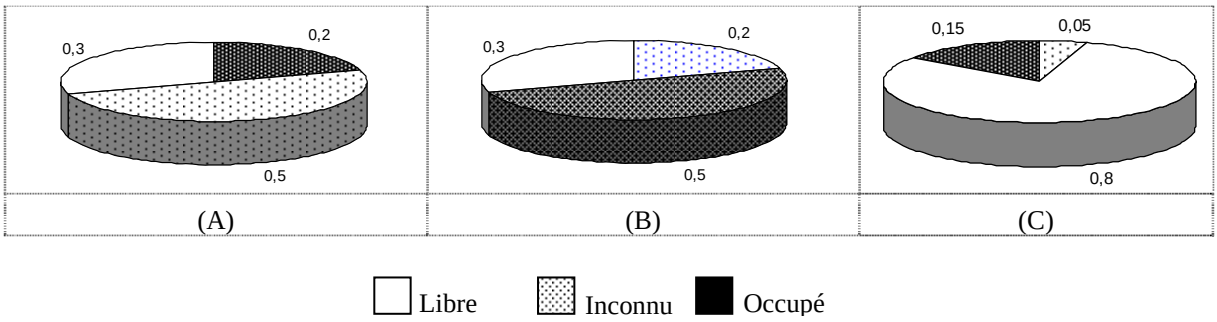


Figure (2.3). Dans chaque exemple, les trois états partiels ont une somme égale à un. Mais, il y a toujours un état partiel dominant. La première cellule sera jugée inconnue, la deuxième sera occupée, alors que la troisième sera libre.

En fait, éviter le calcul d'une seule valeur probabiliste (entre zéro et un) pour juger l'occupation de cellule représente une bonne solution pour éviter l'ambiguïté classique qui est liée à l'interprétation d'une telle valeur pour qu'elle possède une certaine signification entre trois possibles (occupée, libre ou inconnue). Utiliser une seule valeur est souvent associé avec un système du seuillage (inférieur et supérieur) comme celui montré dans la figure (2.4) qui exprime une certaine fragilité du jugement autour des seuils choisis.

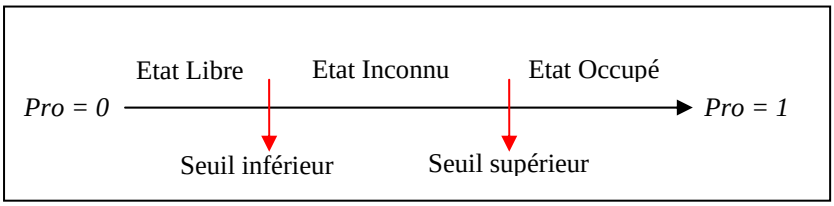


Figure (2.4). Contrairement à notre technique, un système de seuillage va traduire l'état d'une cellule selon sa probabilité d'occupation calculée comme une seule valeur entre zéro et un. Le problème est la fragilité du jugement autour les deux seuils.

La figure (2.5), illustre une grille d'occupation construite pour une partie de la salle robotique (Gérard BAUZIL) dans notre laboratoire. La taille de cellule (5x5 cm^2) fournit une bonne résolution pour la carte élaborée. Pendant l'expérience, le robot est localisé en utilisant un module préexistant (POM : POsition Manager) qui fusionne toutes les informations estimées et fournies par les autres modules (odométrie, localisation sur segments d'un modèle connu), pour garantir une meilleure localisation du robot.

Figure (2.5). Exemple d'une grille d'occupation 2D construite dans notre laboratoire. Les valeurs d'occupation sont codées comme suit : (vert : inconnu ; blanc : libre ; bleu : occupé). La superficie est (10x8m²) ; une cellule correspond à (5x5cm²).

2.5. Conclusion

Ce chapitre nous a introduit brièvement quelques règles probabilistes de base, l'hypothèse de Markov et le filtre de Bayes. Le capteur utilisé est un télémètre laser, son principe et le modèle du capteur sont illustrés. Puis, pour calculer et mettre à jour la probabilité d'occupation de la grille élaborée le (Discrete Bayes Filter DBF) est employé en considérant que l'état d'une cellule puisse exprimer trois états partiels dont la somme des probabilités est égale à un. L'état de la cellule sera attribué à l'état partiel dominant après l'application d'une étape de comparaison. Une carte construite dans un environnement statique dans notre laboratoire est illustrée. L'approche suivie ne prend pas en compte l'occupation des cellules voisines, ce qui est adapté à l'utilisation du capteur choisi (laser). La construction d'une carte dans un environnement dynamique, où il y a des cibles mobiles à distinguer pour que les traces de leurs trajectoires soient retirées de la carte résultante sera discutée dans le chapitre suivant qui présente une technique de SMAT (Simultaneous Mapping And Tracking).

CHAPITRE III

Cartographie et Suivi Simultanés

3.1. Introduction

3.2. Suivi d'objets mobiles

Le modèle proposé pour la dynamique du système
(MSDM, Multiple Switching Dynamic Model)

3.3. Estimation non linéaire des processus markoviens

3.3.1. Positionnement du Problème

3.3.1.1. Prédiction Markovienne

3.3.1.2. Correction Bayesienne

3.3.2. Le Filtrage

3.3.2.1. Le Filtre de Kalman

3.3.2.2. Le Filtre de Kalman Etendu

3.3.2.3. Le Filtrage Particulaire

3.3.2.4. MMPF (Multiple Model Particle Filter)

3.4. Simulations

3.5. Expérimentations

3.5.1. Groupage de Cellules Voisines Occupées

3.5.2. Détection de Cibles Mobiles

3.5.3. Suivi de Cible Mobile

Suivi Multi Cibles (Data Association)

3.6. Conclusion et Discussion

3.1. Introduction

La capacité de faire la détection et le suivi d'objets mobiles est cruciale pour un système robotique. La navigation du robot dans un environnement dynamique nécessite une connaissance précise et à jour de son entourage afin d'assurer l'évitement d'éventuels obstacles ou d'engager des interactions. D'autre part, une telle capacité est indispensable pour la construction d'une carte de l'environnement exprimant les parties statiques pour assurer une bonne localisation du robot. De plus, certaines applications faisant appel à une interaction plus fine avec l'environnement nécessitent outre la simple localisation, la reconnaissance sémantique d'objets [42]. Tel est le cas, par exemple, dans la manipulation ou le guidage du robot par les mouvements de la main humaine, où la reconnaissance et le suivi de la main de l'opérateur est indispensable.

De façon générale, la résolution de ce problème est basée sur l'estimation d'un processus markovien à partir d'observations bruitées en vue de l'extraction, aussi fidèle que possible, d'une information utile (trajectoires d'objets mobiles présents : positions, vitesses et accélérations) à partir des mesures entachées de bruit délivrées par un ensemble de capteurs installés sur le robot. Dans les contextes robotiques cités, l'estimation non linéaire se doit, en plus des performances habituelles d'accrochage et de précision, d'assurer simultanément la détection et le suivi d'objets multiples, ce qui fait apparaître une difficulté supplémentaire qui est la résolution du problème de multi cibles. En l'absence de solution optimale au problème d'estimation dans le cas non linéaire, différentes solutions approchées ont été proposées dans la littérature. L'approche particulaire proposée dans [95, 53, 65], représente une solution numérique globale permettant d'adresser efficacement le problème de filtrage sans restriction particulière sur les équations d'état et/ou sur la nature de bruit. Ces performances largement démontrées dans divers domaines d'application, la prédisposent pour les problèmes de détection et de suivi d'objets mobiles.

La suite du chapitre s'articule comme suit : Dans le paragraphe 2, nous rappelons le problème de suivi d'objet, et nous présentons la modélisation adoptée pour le système (cible - observateur). La troisième section rappelle brièvement les bases de l'estimation non linéaire de processus markovien à temps discret, ainsi que sa solution classiquement adoptée (Filtre de Kalman Etendu), et le filtrage particulaire, d'abord dans sa version originale puis dans le MMPF (Multiple Models Particle Filter) adaptée au modèle choisi pour la dynamique de cible. Nos stratégies pratiques qui profitent de la carte construite simultanément pour faire la détection de cibles mobiles et ensuite le suivi, sont présentés dans le quatrième paragraphe qui illustre aussi quelques résultats expérimentaux. Dans le cinquième et dernier paragraphe, nous discutons les résultats obtenus par l'approche proposée, en portant une attention particulière aux limitations de l'approche utilisée et nous proposons une solution afin d'y remédier.

3.2. Suivi d'objets mobiles

Le problème du suivi d'objet mobile est défini mathématiquement dans les coordonnées cartésiennes illustrées dans la figure (3.1), en considérant l'ensemble du système se composant d'un observateur (ici le robot) et d'une cible mobile [52]. Dans ce cas nous supposons que l'état de la cible est déterminé par sa position et sa vitesse :

$$X^C = \left[x^C \;\; y^C \;\; \dot{x}^C \;\; \dot{y}^C\right]^T \qquad \textbf{3. 1}$$

Pareillement pour l'état du robot :

$$X^R = \left[x^R \;\; y^R \;\; \dot{x}^R \;\; \dot{y}^R\right]^T \qquad \textbf{3. 2}$$

La différence entre les deux états représente un état relatif à estimer :

$$X = X^C - X^R = \left[x \;\; y \;\; \dot{x} \;\; \dot{y}\right]^T \qquad \textbf{3. 3}$$

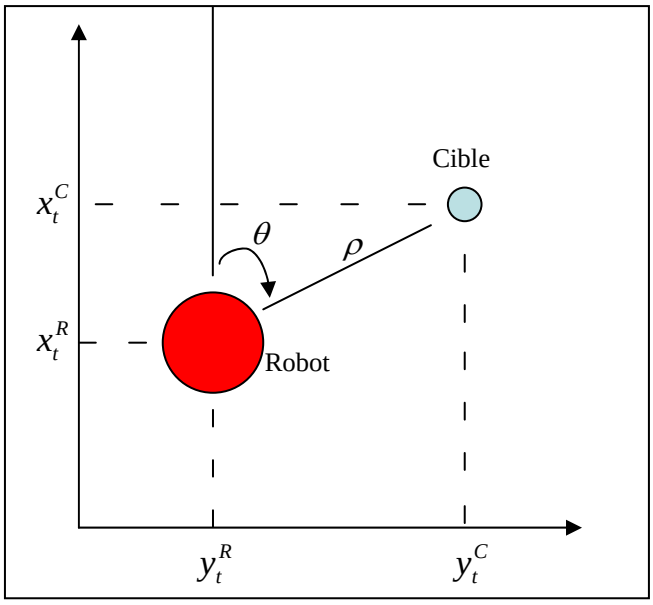


Figure (3.1). Géométrie de (Cible – Observateur) illustrée en deux dimensions.

Notre cible est supposée manœuvrante, ce qui est tout à fait le cas pour l'être humain. D'où, sa dynamique est modélisable par le (Système de Markov à saut). Son mouvement doit suivre à chaque instant un parmi trois comportements prévus : continuer sa trajectoire tout droit, tourner à droite (Clockwise) ou tourner à gauche (Anticlockwise). En conséquence : $S = \{1,2,3\}$, dénote l'ensemble de trois modèles illustrés dans la figure (3.2.a). Ces trois modèles sont adéquats pour estimer le mouvement d'un véhicule dont l'angle maximal de giration est limité et précalculable, mais pas pour estimer une trajectoire libre sans aucune limite, ce qui est notre cas, où il est possible à n'importe quel moment d'avoir un changement radical de la direction de la cible suivie. Pour prendre cette dernière vérité en compte, nous avons appliqué la modification illustrée dans la figure (3.2.b). Dans ce cas, le calcul va garder les mêmes valeurs de vitesses estimées pour $S = 2$ et $S = 3$ dans la figure (3.2.a), mais dans un sens inverse. Ainsi, la cible sera complètement entourée par trois zones de prédiction, et une seule mesure sera suffisante pour lancer l'autre phase du filtrage, celle de correction.

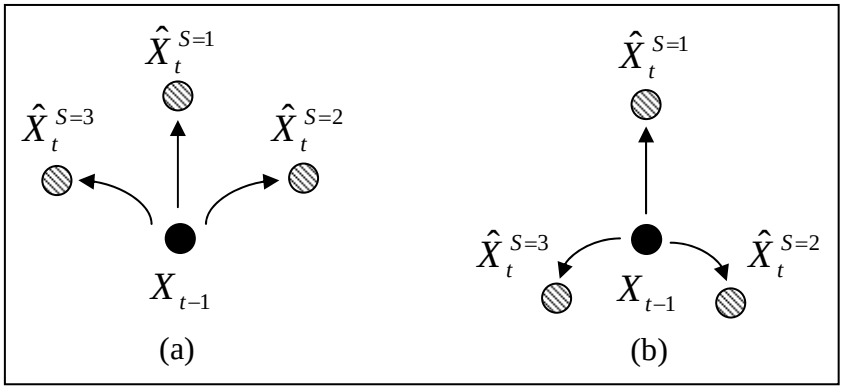


Figure (3.2). (a) L'approche de suivi proposée dans [52] où trois états sont prédits pour trois modèles de comportement possibles; (b) l'approche modifiée qui convient mieux à la caractéristique aléatoire du mouvement humain.

Le modèle proposé pour la dynamique du système (MSDM, Multiple Switching Dynamic Model)

Pour analyser le comportement d'un système dynamique il faut qu'il soit décrit au moins par deux modèles : un premier qui représente l'évolution d'état du système avec le temps et un second qui modélise une certaine relation entre les mesures et l'état du système.

Ainsi, la dynamique de notre système peut être décrite par l'équation suivante :

$$X_{t+1} = f(X_t, v_t) \qquad \textbf{3. 4}$$

(t) est l'indice du temps, X_t représente l'état relatif représenté dans l'équation (3.3), f est une fonction souvent non linéaire décrivant le mouvement de la cible. Le vecteur de mesures bruitées Z_t sera lié au vecteur d'état X_t par l'équation suivante :

$$Z_t = h(X_t) + w_t \qquad \textbf{3. 5}$$

v_{t-1} et w_t sont respectivement le bruit du processus et le bruit de mesure (les deux bruits sont supposés blancs et indépendants l'un de l'autre). Alors que :

$$h(X_t) = \left[h_\rho(X_t) \quad h_\theta(X_t)\right]^T \qquad \textbf{3. 6}$$

$$h_\rho(X_t) = \rho = \sqrt{x_t^2 + y_t^2}\ , \quad h_\theta(X_t) = \theta = \arctan(\frac{x}{y})$$

Faire le suivi d'un objet mobile peut être interprété comme une détermination à chaque instant de la probabilité a posteriori suivante :

$$p(X_t \mid Z_t) \qquad \textbf{3. 7}$$

Soit $r_{t+1} \in S$ le mode du régime qui prend une des trois valeurs selon le modèle du mouvement de cible observée. La dynamique du système peut être écrite selon Gordon [52] par :

$$X_{t+1} = f(X_t, X_t^R, X_{t+1}^R, r_{t+1}) + \Gamma_t v_t \qquad \textbf{3. 8}$$

Avec v_t le bruit du processus est un vecteur de dimension (2x1), et Γ_t qui représente une matrice de l'intervalle d'acquisition des données peut être écrite :

$$\Gamma_t = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \qquad \textbf{3. 9}$$

T : l'intervalle d'acquisition des mesures

$$f(X_t, X_t^R, X_{t+1}^R, r_{t+1}) = F^{(r_{t+1})}(X_t) \cdot (X_t + X_t^R) - X_{t+1}^R \qquad \textbf{3. 10}$$

$f(\cdot)$ est une fonction de transition d'état conditionnée par le comportement de la cible, $F^{(r_{t+1})}(\cdot)$ est une matrice de transition correspondant à la valeur prise par le mode du régime r_{t+1}.

Donc, pour $(r_{t+1} == 1)$ nous avons une matrice qui modélise un mouvement direct à une vitesse constante :

$$F^{(1)}(X_t) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textbf{3. 11}$$

Alors que si $(r_{t+1} == 2 \;\; Ou \;\; r_{t+1} == 3)$ nous avons deux modèles non linéaires qui modélisent les mouvements circulaires (rétrograde) et (direct) de la cible suivie:

$$F^{(j)}(X_t) = \begin{bmatrix} 1 & 0 & \frac{\sin(\Omega_t^j T)}{\Omega_t^j} & -\frac{(1-\cos(\Omega_t^j T))}{\Omega_t^j} \\ 0 & 1 & \frac{(1-\cos(\Omega_t^j T))}{\Omega_t^j} & \frac{\sin(\Omega_t^j T)}{\Omega_t^j} \\ 0 & 0 & -\cos(\Omega_t^j T) & \sin(\Omega_t^j T) \\ 0 & 0 & -\sin(\Omega_t^j T) & -\cos(\Omega_t^j T) \end{bmatrix} \qquad \textbf{3. 12}$$

Où : $j = 2 \; ou \; 3$

Ω_t^2, Ω_t^3 sont les taux de rotation conditionnés par le mode du régime et la vitesse de la cible suivie.

$a_m > 0$ est l'accélération typique d'une cible manœuvrante.

$$\Omega_t^2 = \frac{a_m}{\sqrt{(\dot{x}_t + \dot{x}_t^R)^2 + (\dot{y}_t + \dot{y}_t^R)^2}} \qquad \textbf{3. 13}$$

$$\Omega_t^3 = \frac{-a_m}{\sqrt{(\dot{x}_t + \dot{x}_t^R)^2 + (\dot{y}_t + \dot{y}_t^R)^2}} \qquad \textbf{3. 14}$$

Une matrice de probabilités de transition qui définit les probabilités de changement d'état parmi les trois comportements possibles, est la suivante :

$$\Pi_{ab} = \begin{pmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{pmatrix}, \text{Les valeurs choisies sont} \begin{pmatrix} .32 & .34 & .34 \\ .34 & .34 & .32 \\ .34 & .32 & .34 \end{pmatrix} \qquad \textbf{3. 15}$$

$\pi_{ab} \geq 0$, alors que la somme des probabilités sur la même ligne est égale à un.

Dans notre application, nous avons choisi la probabilité pour que la cible suivie aille tout droit π_{11} inférieure aux autres valeurs dans sa ligne, donc elle préfère changer ses directions aléatoirement pendant sa trajectoire. Nous avons également choisi les probabilités π_{23} et π_{32} inférieures aux autres valeurs dans leurs linges, parce qu'il y a moins de chance pour qu'une personne change soudainement sa direction de l'extrême droite à l'extrême gauche ou l'inverse.

3.3. Estimation non linéaire des processus markoviens

Différentes méthodes sont proposées dans la littérature pour traiter un problème d'estimation d'un système non linéaire. Dans ce paragraphe, le problème d'estimation est introduit de façon globale et simplifiée, puis les propriétés essentielles de trois filtres les plus connus (Filtre de Kalman, Kalman Etendu, Filtrage particulaire) sont brièvement expliquées. Compte tenu des avantages connus du filtre particulaire pour traiter ce type de systèmes, nous avons choisi l'une de ses variantes (le MMPF : Multiple Model Particle Filter) qui convient au modèle (MSDM) proposé ci-dessus, pour être le mécanisme réalisant le processus de suivi souhaité.

3.3.1. Positionnement du Problème

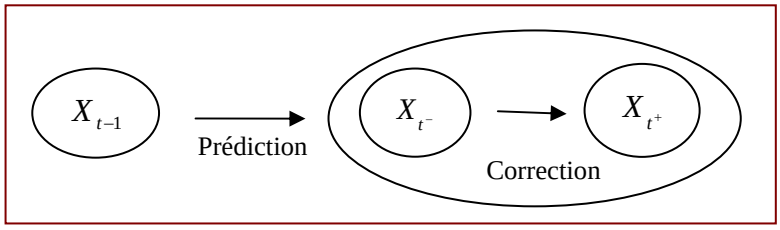


Figure (3.3), Le filtrage se compose comme d'habitude de deux phases (prédiction et correction)

De manière générale, le problème d'estimation dynamique consiste à reconstruire un processus stochastique à partir de son observation bruitée. Le filtrage permet d'avoir une information améliorée, plus précise et plus fiable que les mesures brutes grâce à un algorithme de calcul récursif sur des mesures bruitées pour déterminer l'état du système dynamique. Le filtrage est effectué en deux phases (figure 3.3), la première est la phase de prédiction qui déduit l'état à l'instant $(t)^-$ à partir d'observations incertaines qui sont prises jusqu'à l'instant $(t-1)$ et de l'évolution du système, la deuxième est la correction qui déduit l'état le plus probable a posteriori $(t)^+$ sachant l'état estimé a priori $(t)^-$.

3.3.1.1 Prédiction Markovienne

Il s'agit de calculer à l'instant (t) la densité $p(X_t \mid Z_{0:t-1})$ conditionnelle aux dernières observations $Z_{0:t-1}$, à partir de la connaissance de la densité probabiliste conditionnelle $p(X_{t-1} \mid Z_{0:t-1})$ calculée à l'instant précédent, et en se basant sur le caractère markovien du système.

$$p(X_t \mid Z_{0:t-1}) = p(X_t \mid X_{t-1}) p(X_{t-1} \mid Z_{0:t-1}) \qquad \textbf{3. 16}$$

La densité $p(X_t \mid X_{t-1})$ représente la probabilité de transition de l'état X_{t-1} à l'état X_t, elle s'obtient à partir de l'équation d'évolution dynamique du système (3.4).

3.3.1.2 Correction Bayesienne

Cette étape se base sur la règle de Bayes, et consiste en la correction de la densité de probabilité conditionnelle compte tenu de la nouvelle observation Z_t.

$$p(X_t \mid Z_{0:t}) = \frac{p(Z_t \mid X_t) p(X_t \mid Z_{0:t-1})}{p(Z_t \mid Z_{0:t-1})} \qquad \textbf{3. 17}$$

Le terme $p(Z_t \mid Z_{0:t-1})$ peut se réécrire en se basant sur le caractère markovien :

$$p(Z_t \mid Z_{0:t-1}) = \int_{X_t} p(Z_t \mid X_t) p(X_t \mid Z_{0:t-1}) dX_t \qquad \textbf{3. 18}$$

La densité $p(Z_t \mid X_t)$ est déterminable à partir de l'équation d'observation (3.5).

3.3.2. Le Filtrage

Formuler les deux modèles du système analysé (modèle d'évolution d'état et celui de mesure) d'une façon probabiliste et les mettre à jour d'une façon continue fournit une technique de filtrage qui est une solution robuste pour traiter des mesures bruitées et séquentiellement reçues d'un ou de plusieurs capteurs pour effectuer une estimation récursive de l'état du système. Choisir un filtre approprié au système étudié doit prendre en compte les points suivants :

1. La linéarité ou la non linéarité du système.
2. Si le système étudié impose une distribution probabiliste uni-modale (Gaussienne) ou multimodale (non-paramétrique).
3. Si la dynamique du système comporte un régime multi-hypothèses dans son espace d'état, et si un filtre uni-modale mais multi-hypothèses est suffisant pour le modéliser, ou s'il faut un filtre qui représente la distribution probabiliste comme un ensemble d'échantillons aléatoires d'état pour chaque hypothèse proposée.

Par exemple, le filtre de Kalman est utilisé chez Bennewitz et al. [16] pour faire le suivi multi-cible. Ils ont créé un filtre indépendant pour suivre la trajectoire de chaque cible différente, où quelques patterns du mouvement ou des trajectoires sont a priori connus et appris. Dans un autre cas, une approche basée sur un modèle brownien du mouvement de cible humaine emploie le filtrage particulaire [81] pour effectuer le suivi des personnes mobiles et localiser simultanément le robot en basant sur des lectures de son télémètre laser. En outre, le filtre particulaire et le laser sont également employés dans [97] pour effectuer le suivi des cibles multiples à partir d'un robot en mouvement avec l'application de l'approche (Joint Probabilistic Data Association - JPDA) et l'adoption d'un modèle linéaire du mouvement humain. L'utilisation de JPDA comme un filtre peut être utile pour garantir un bon niveau d'association des données, mais il faut toujours faire attention à la complexité du calcul qu'il impose à cause de l'évaluation approfondie de plusieurs mesures et de différentes hypothèses. En tout cas, trois types fondamentaux de filtres ou leurs variantes sont en général proposés pour modéliser différents types de systèmes : le filtre de Kalman, le filtre de Kalamn étendu et le filtre particulaire.

3.3.2.1 Le Filtre de Kalman

Ce filtre suppose que le modèle d'évolution d'état et celui de mesure sont linéaires avec des erreurs gaussiennes. Les deux modèles sont respectivement les suivants :

$$X_t = A_t X_{t-1} + B_t U_t + v_t \qquad \textbf{3. 19}$$

X_t : le vecteur d'état, U_t : le vecteur de commande, A_t et B_t sont deux matrices de taille ($n \times n$) et ($n \times m$) respectivement, n : est la taille du vecteur d'état, m : est la taille du vecteur de commande. Le bruit gaussien de processus v_t modélise la transition aléatoire d'état, sa moyenne est nulle et sa covariance est une matrice R_t.

$$Z_t = C_t X_t + \omega_t \qquad \textbf{3. 20}$$

Z_t : le vecteur de mesure de taille k , C_t : est une matrice de taille ($k \times n$). Le bruit de mesure est modélisé par un vecteur ω_t, sa distribution est aussi une gaussienne de moyenne nulle et de covariance Q_t.

L'algorithme du Filtre du Kalman

L'entrée de ce filtre est la vraisemblance de l'état du système représentée par sa moyenne et sa covariance (μ_{t-1}, Σ_{t-1}) à l'instant ($t-1$). La première et la deuxième étape de l'algorithme déterminent les valeurs estimées ($\overline{\mu}_t, \overline{\Sigma}_t$) avant l'incorporation de la mesure. La troisième étape présente le calcul du gain de Kalman K_t qui décide le degré d'importance de l'incorporation du vecteur de mesure Z_t. Les quatrième et la cinquième étapes sont une détermination de l'état mis à jour représenté par ses deux composants (μ_t, Σ_t).

Tableau 1. **Algorithme du filtre de Kalman**

$(\mu_{t-1}, \Sigma_{t-1}, U_t, Z_t)$

1. $\bar{\mu}_t = A_t \, \mu_{t-1} + B_t \, U_t$
2. $\bar{\Sigma}_t = A_t \, \Sigma_{t-1} \, A_t^T + R_t$
3. $K_t = \bar{\Sigma}_t \, C_t^T (C_t \, \Sigma_t \, C_t^T + Q_t)^{-1}$
4. $\mu_t = \bar{\mu}_t + K_t (Z_t - C_t \, \bar{\mu}_t)$
5. $\Sigma_t = (I - K_t \, C_t) \bar{\Sigma}_t$

3.3.2.2 Le Filtre de Kalman Etendu

Un système décrit par un modèle linéaire est rare dans les applications robotiques. La plupart des systèmes et des phénomènes physiques sont non-linéaires. Le filtre de Kalman étendu permet de traiter ces systèmes. Les deux équations suivantes représentent respectivement un modèle d'évolution d'état et un modèle de mesure, où (g) et (h) sont deux fonctions non-linéaires :

$$X_t = g(X_{t-1}, U_t) + v_t \qquad \textbf{3. 21}$$

$$Z_t = h(X_t) + \omega_t \qquad \textbf{3. 22}$$

Ce filtre fournit une approximation gaussienne de la distribution probabiliste a posteriori de l'état du système représentée par sa moyenne μ_t et sa covariance Σ_t. La fiabilité de la dernière approximation dépend du degré de non-linéarité de la fonction approchée. Ce filtre peut aussi diverger si l'hypothèse d'indépendance entre les deux bruits n'est pas vérifiée. La modélisation d'un système non-linéaire en utilisant l'une des variantes de ce filtre comme (MH-EKF : multi-hypothesis Extended Kalman Filter) peut être utile avec un nombre borné d'hypothèses supposées dans l'espace d'état. Sinon, le coût de calcul quadratique devient prohibitif.

L'algorithme du Filtre de Kalman Etendu

Cet algorithme répète dans les cinq étapes illustrées dans le tableau (2) un déroulement semblable à celui qui est déjà présenté dans sa version linéaire (tableau 1). Les deux Jacobiennes G_t et H_t remplacent respectivement les matrice A_t, B_t et C_t qui apparaissent dans le modèle linéaire du filtre de Kalman.

Tableau 2. **Algorithme du filtre de Kalman Etendu**

$(\mu_{t-1}, \Sigma_{t-1}, U_t, Z_t)$

1. $\bar{\mu}_t = g(\mu_{t-1}, U_t)$
2. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$
3. $K_t = \bar{\Sigma}_t H_t^T (H_t \Sigma_t H_t^T + Q_t)^{-1}$
4. $\mu_t = \bar{\mu}_t + K_t (Z_t - h(\bar{\mu}_t))$
5. $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$

3.3.2.3 Le Filtrage Particulaire

Il s'agit d'une expression non-paramétrique du filtre de Bayes. L'idée principale [52] est de représenter la distribution probabiliste a posteriori par un ensemble d'échantillons d'état (particules) tirés aléatoirement dans une zone plus probable de l'espace d'état global. Ce filtre ne se fonde pas sur une forme fonctionnelle fixée du posteriori. Il est donc différent des filtres Gaussiens (Filtre de Kalman et Kalman Etendu) qui ont un simple maximum focalisé autour de l'état vrai avec une marge d'incertitude, les rendant peu adaptés pour traiter des problèmes globaux d'évaluation dans lesquels plusieurs d'hypothèses distinctes existent.

La technique du filtrage particulaire effectue l'estimation non-linéaire par un nombre fini d'échantillons (maillage particulaire), chacun correspondant à une région dans l'espace d'état. Elle est capable d'approximer une grande variété de distributions probabilistes, ce qui la rend efficace et très utilisée pour traiter des systèmes non linéaires complexes (mais le nombre de particules nécessaire pour atteindre une précision donnée peut être grand selon l'application). A chaque itération, cet échantillonnage évolue par simulation aléatoire du système, par la propagation des densités élémentaires soumise à des réalisations aléatoires des bruits de la dynamique du système et leurs corrections au vu des nouvelles observations. Régulièrement, les particules sont redistribuées pour éliminer celles de faibles poids (peu probables), au profit des plus massives, favorisant ainsi les régions d'état les plus pertinentes.

Equations du Filtrage Particulaire

Considérons un système non linéaire dont la dynamique et le modèle de mesure sont régit par les équations suivantes :

$$\mathrm{x}_{t+1} = f(\mathrm{x}_t, v_t) \qquad \textbf{3. 23}$$

$$\mathrm{z}_t = h(\mathrm{x}_t, \omega_t) \qquad \textbf{3. 24}$$

x_t : vecteur d'état du système, z_t : vecteur de mesures, f : modèle de processus (non linéaire), h : fonction d'observation, v_t et ω_t sont respectivement le bruit du processus et le bruit de mesure (les deux bruits sont supposés blancs et indépendants l'un de l'autre).

Soit,

$$p_N(\mathrm{x}_{t-1} \mid z_{0:t-1}) = \sum_{i=0}^{N-1} w_{t-1|t-1}^{i} \delta(\mathrm{x}_{t-1} - \mathrm{x}_{t-1}^{i}) \cong p(\mathrm{x}_{t-1} \mid z_{0:t-1}) \qquad \textbf{3. 25}$$

une approximation aléatoire de $p(\mathrm{x}_{t-1} \mid z_{0:t-1})$ par une somme de densités élémentaires $\delta(\mathrm{x}_{t-1} - \mathrm{x}_{t-1}^{i})$ qui ont des poids normalisés : $w_{t-1|t-1}^{i}$ et sont localisées dans l'espace d'état par leurs moyennes $\mathrm{x}_{t-1|t-1}^{i}$, $\delta(.)$ est la masse de Dirac.

Prédiction

Une approximation de $p(\mathrm{x}_t \mid z_{0:t-1})$ est alors donnée par :

$$p(\mathrm{x}_t \mid z_{0:t-1}) = \sum_{i=0}^{N-1} w_{t|t-1}^{i} \delta(\mathrm{x}_t - \mathrm{x}_t^{i}) \qquad \textbf{3. 26}$$

Où :

$\mathrm{x}_t^{i} = f(\mathrm{x}_{t-1}^{i}, v_t^{i})$

$\{v_t^{i}\}_{i=0}^{N-1}$, sont tirés indépendamment suivant la loi de v_t

Correction

De même, une approximation de $p(\mathrm{x}_t \mid z_{0:t})$ est donnée par :

$$p(\mathrm{x}_t \mid z_{0:t}) = \frac{p(z_t \mid \mathrm{x}_t) p(\mathrm{x}_t \mid z_{0:t-1})}{\int p(z_t \mid \mathrm{x}_t) p(\mathrm{x}_t \mid z_{0:t-1}) d\mathrm{x}_t}$$

$$\cong \sum_{i=0}^{N-1} w_{t|t}^{i} \delta(\mathrm{x}_t - \mathrm{x}_t^{i}) \qquad \textbf{3. 27}$$

Où :

$$w_{t|t}^{i} = \frac{w_{t|t-1}^{i} p(z_t \mid \mathrm{x}_t^{i})}{\sum_{j=0}^{N-1} w_{t|t-1}^{j} p(z_t \mid \mathrm{x}_t^{j})}$$

$p(z_t \mid \mathrm{x}_t^{j})$ est supposé connue a priori.

Les particules dans une distribution probabiliste a posteriori peuvent être représentées comme suit :

$$\chi_t = \mathrm{x}_t^{1}, \mathrm{x}_t^{2}, \ldots \mathrm{x}_t^{N}$$

Chaque particule x_t^i (avec $i = [1,....., N]$) représente un échantillon discret d'état à l'instant (t). χ_t représente un groupe de particules, alors que N est le nombre des particules, qui peut atteindre une valeur élevée (et augmenter en conséquence le temps de calcul).

L'algorithme du Filtre particulaire

Tableau 3. **Algorithme du filtrage particulaire**

(χ_{t-1}, u_t, z_t)

1. $\overline{\chi}_0 = \chi_0 = \phi$
2. *For* $i = 1$ *to* N *do*
3. *sample* $\mathrm{x}_t^i \sim p(\mathrm{x}_t | \mathrm{x}_{t-1}^i, u_t)$
4. $w_t^i = p(z_t | \mathrm{x}_t^i)$
5. $\overline{\chi}_t = \overline{\chi}_t + \langle \mathrm{x}_t^i, w_t^i \rangle$
6. *End For*
7. *For* $i = 1$ *to* N *do*
8. *draw* i *with probability* α w_t^i
9. *add* x_t^i *to* χ_t
10. *End For*
11. *Return* χ_t

L'entrée de ce filtre est un groupe des particules χ_{t-1} déjà construit à l'instant précédent ($t-1$) avec deux vecteurs, un de commande u_t et autre de mesure z_t. La première étape dans cet algorithme initialise un groupe temporaire des particules $\overline{\chi}_t$. Puis une étape de prédiction commence avec un tirage aléatoire des particules x_t^i, chacune d'entre elles dépendant de l'état des particules à l'instant précédent x_{t-1}^i et du vecteur de commande le plus récent. A chaque particule il faut assigner une valeur décrivant son importance qui est l'interprétation de la probabilité d'avoir un vecteur de mesures z_t en sachant l'état courant des particules x_t^i. Générer N particules pondérées sera la fin de l'étape de prédiction en formant le groupe $\overline{\chi}_t$. Ensuite, une étape de redistribution des particules produites dans la dernière étape sera effectuée, où les particules qui sont le moins importantes disparaissent du groupe résultant χ_t, alors que les autres qui ont un poids suffisamment élevé vont donner naissance (chacune selon son poids) à une ou plusieurs particules ayant le même poids unitaire (1/ N). Donc, à chaque itération la distribution des particules va être plus dense dans les régions de l'espace d'état où se trouvent les particules les plus lourdes. Autrement dit il s'agit

d'un processus de concentration (ou condensation) des particules qui évolue avec le temps sur l'état le plus probable.

3.3.2.4 MMPF (Multiple Model Particle Filter)

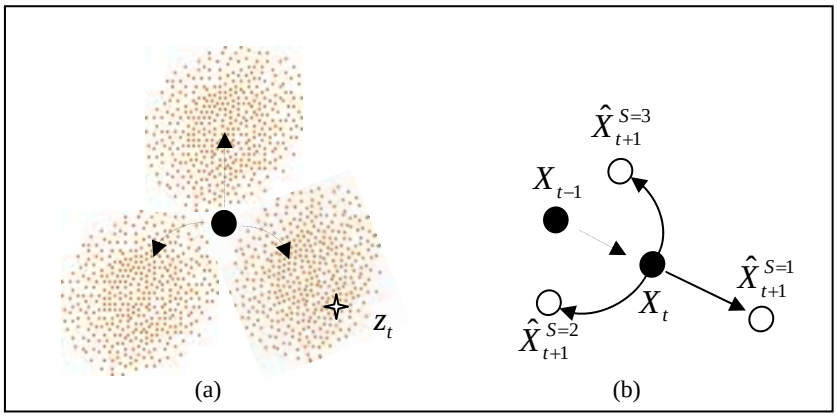


Figure (3.4). (a) cent particules dans chacun des trois nuages, avant qu'une mesure z_t soit prise; (b) après le choix de la particule la plus probable à l'instant (t), une nouvelle itération commence.

Le besoin d'une technique qui effectue le filtrage pour un système non linéaire modélisé par une dynamique changeante (Switching Dynamic Model), qui est le cas du problème traité dans ce chapitre, nous a amené à choisir le filtre particulaire (MMPF) [52]. Dans notre application, ce filtre couvre chacune des trois zones de prédiction (figure 3.4.a) par un nuage de N particules, où à chaque particule est assigné un poids qui représente la probabilité de la particule de décrire la vraisemblance de l'état :

$$\tilde{w}_t^i = p(z_t \mid \mathrm{x}_t^i(r_t)) * p(r_t^i \mid r_{t-1}) \qquad \textbf{3. 28}$$

$i = 1,....,N$, z_t : la mesure reçue de la cible suivie à l'instant (t), $\mathrm{x}_t^i(r_t)$: l'état estimé de la particule conditionné par le mode du régime r_t. $p(r_t^i \mid r_{t-1}) \in \Pi_{ab}$ représente la probabilité de transition de mode du régime d'une particule entre deux instants $[t-1, t]$. Pour un bruit de mesure additif gaussien, la distribution probabiliste $p(z_t \mid \mathrm{x}_t^i(r_t))$ peut être calculée comme suit :

$$p(z_t \mid \mathrm{x}_t^i(r_t)) = \eta \exp\left[-\frac{1}{2}\left(\frac{(\mathrm{x}_t^z - \mathrm{x}_t^i(r_t))^2}{\sigma_x^2} + \frac{(\mathrm{y}_t^z - y_t^i(r_t))^2}{\sigma_y^2}\right)\right] \qquad \textbf{3. 29}$$

η : est une constante de normalisation, (x_t^z et y_t^z) : sont les coordonnées mesurées de la cible suivie, ($\mathrm{x}_t^i(r_t)$ et $y_t^i(r_t)$) sont celles de la particule (i), (σ_x et σ_y) sont deux écarts types pré-choisis.

Calculer la moyenne des poids non normalisés (équations 3.28) dans chaque zone, sera utile pour indiquer laquelle d'entre elles est la plus probable. Puis, générer *N* particules et les redistribuer dans cette dernière zone choisie, sera suffisant pour avoir une zone de densité de probabilité maximale, où il faut appliquer une étape qui élimine toutes les particules les moins probables et qui choisit la plus probable. L'état estimé devient déterminé et une nouvelle étape d'estimation peut commencer figure (3.4.b).

Avoir une étape de redistribution peut garantir un bon niveau de robustesse du fonctionnement de ce filtre surtout avec des cibles de vitesse élevée (voiture) ou dans une application caractérisée par un bruit de moyenne non nulle (contrairement à l'hypothèse). Dans nos expériences pratiques, les cibles suivies sont relativement lentes. Nous avons donc tenté d'éviter cette étape en essayant de minimiser le temps de calcul. Dans ce cas, l'approche est réduite pour qu'elle devienne une sélection directe de la particule la plus probable. Ainsi, les étapes (4 et 5) seront annulées dans le tableau 4. L'approche réduite a prouvé notre point de vue, où elle nous a montré une bonne performance (la différence de qualité de trajectoire estimée est entre 2 et 3 cm, ce qui peut être négligeable dans un tel cas) pour une moyenne de temps de calcul égale à 10 ms.

Algorithme de (Multiple Model Particle Filter)

Les six premières étapes de l'algorithme présenté dans le tableau (4) ont été expliquées ci-dessus et illustrées dans la figure (3.4.a), la septième étape est une normalisation des poids des particules. La huitième étape calcule le nombre des particules efficaces $\hat{N}_{eff}$ qui sera utile dans la neuvième étape pour faire un test heuristique afin de savoir si ce nombre est inférieur à un seuil donné N_{thr}. Car si c'était vraiment le cas, l'algorithme va distinguer un phénomène de dégénérescence à éviter. Un phénomène de dégénérescence peut se produire à cause de l'effet des tirages aléatoires successifs du bruit de la dynamique qui cause une diminution de la capacité d'exploration de l'espace des trajectoires du filtre jusqu'à l'extinction. Cette dégénérescence se traduit par l'appauvrissement en particules de régions pertinentes de l'espace d'état et donc la convergence de presque tout les poids particulaires vers zéro. L'interprétation des poids particulaires comme les probabilités d'exploration de ses positions dans l'espace d'état induit une redistribution dont l'algorithme est illustré dans le tableau (5).

Tableau 4. **Algorithme de (Multiple Model Particle Filter)**

1. créer N particules

FOR $i = 1:N$

FOR $r_t = 1:S$

$$\mathrm{x}_t^i(r_t) = f(\mathrm{x}_{t-1}^i, \mathrm{x}_{t-1}^R, \mathrm{x}_t^R, r_t) + \Gamma_t v_t^i$$

$$\tilde{w}_t^i = p(z_t \mid \mathrm{x}_t^i(r_t))\, p(r_t \mid r_{t-1}^i) w_{t-1}^i$$

END FOR

END FOR

2. Calculer le poids moyen dans chaque zone

FOR $r_t = 1:S$

$$\overline{w}_t(r_t) = \sum_{i=1}^{N} (\overline{w}_t(r_t) + \tilde{w}_t^i) / N$$

END FOR

3. Indiquer la zone la plus probable

4. Générer N particules { $\mathrm{x}^{i^j}, w^{i^j} = 1/N$ } *dans la zone choisie en utilisant l'algorithme dans le tableau (5)*

5. Redistribuer les particules dans la zone et calculer leurs poids

FOR $j = 1:N$

$$\mathrm{x}_t^j = f(\mathrm{x}_{t-1}^{i^j}, \mathrm{x}_{t-1}^R, \mathrm{x}_t^R) + \Gamma_t v_t^j$$

$$\tilde{w}_t^j = p(z_t \mid \mathrm{x}_t^j)$$

END FOR

6. Choisir la particule la plus probable.

7. Normalisation de poids

FOR $j = 1:N$

$$w_t^j = \frac{\tilde{w}_t^j}{\sum_{i=1}^{N} \tilde{w}_t^i}$$

END FOR

8. Calculer $\hat{N}_{eff} = \dfrac{1}{\sum_{j=1}^{N} \left(w_t^j\right)^2}$

9. If $\hat{N}_{eff} < N_{thr}$

(Redistribution) en utilisant l'algorithme dans le tableau (5)

Tableau 5. **Algorithme de Redistribution [52]**
«Peigne de Kitagawa »

1. Initialisation d'une Somme Cumulative de Poids (SCP):

$c_1 = w_t^1$

FOR $i = 2 : N$
  2. Construire SCP: $c_i = c_{i-1} + w_t^i$
END FOR

3. Commencer d'en bas du SCP : $i = 1$

4. Dessiner le premier point: $u_1 \sim \upsilon\left[0, N^{-1}\right]$

  FOR $j = 1 : N$

    $u_j = u_1 + N^{-1}(j-1)$

    WHILE $u_j > c_i$

      $i = i + 1$

    END WHILE

    Assigner particule: $\mathrm{x}_t^{j^*} = \mathrm{x}_t^i$
    Assigner poids: $w_t^j = 1/N$
    Assigner parent : $i^j = i$

  END FOR

A la fin de l'algorithme de redistribution, (tableau 5), toutes les particules ont le même poids $w_k^j = 1/N$. Où $\upsilon\left[0, N^{-1}\right]$ Représente une distribution uniforme dans l'intervalle $\left[0, N^{-1}\right]$. Pour chaque particule redistribuée $\mathrm{x}_t^{j^*}$ l'algorithme va assigner l'indice de son parent qui la produit, dénoté par (i^j).

3.4. Simulations

Dans une première phase de nos travaux et afin de tester la validité de l'approche de suivi discutée dans ce chapitre, nous avons réalisé quelques simulations sous MATLAB. Cent particules sont tirées aléatoirement dans chacune des trois zones de prédiction, la fréquence d'acquisition des mesures est de 20 Hz, la vitesse de la cible 0.4 m/s, la vitesse de l'observateur (Robot) 0.1 m/s, et le bruit de mesure supposé blanc et additif.

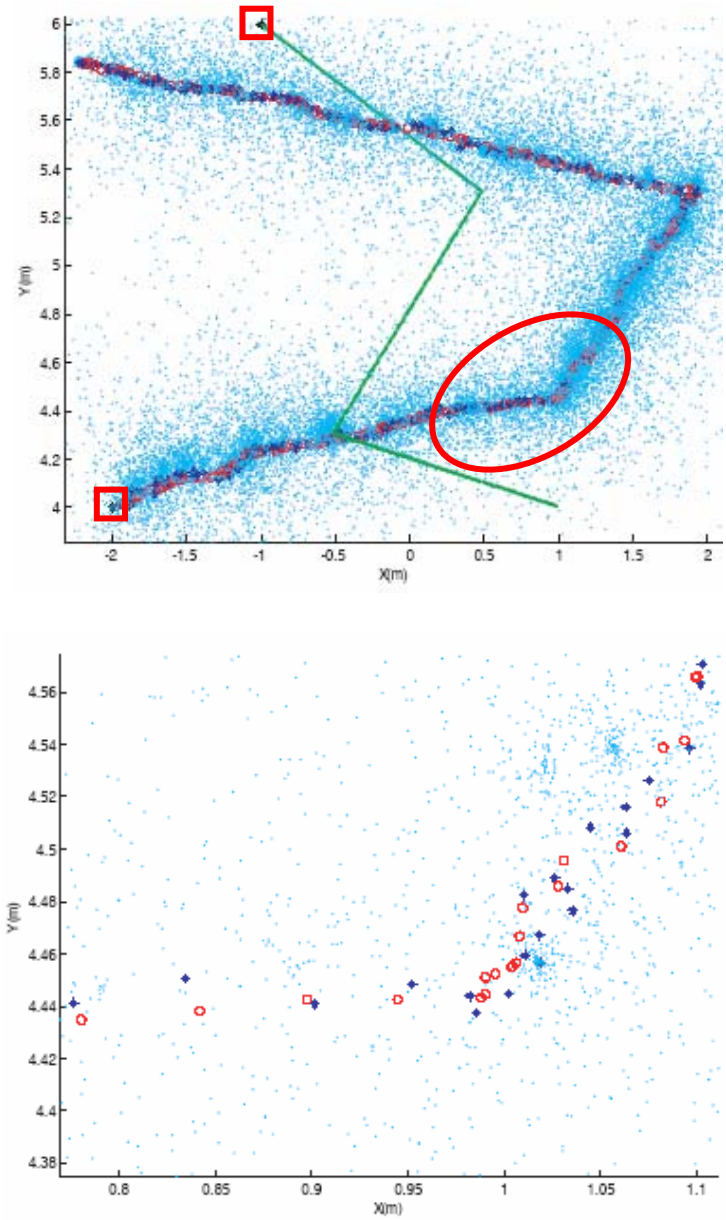
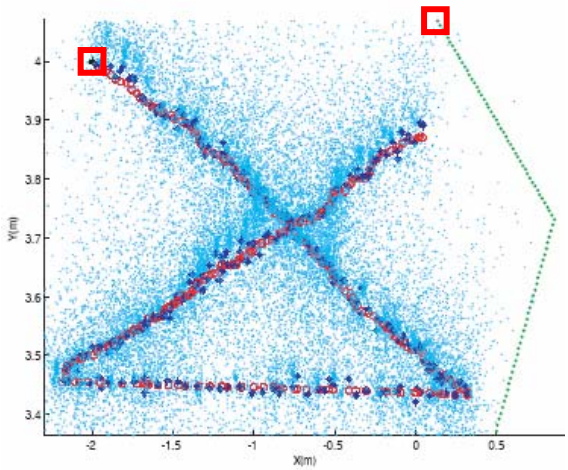
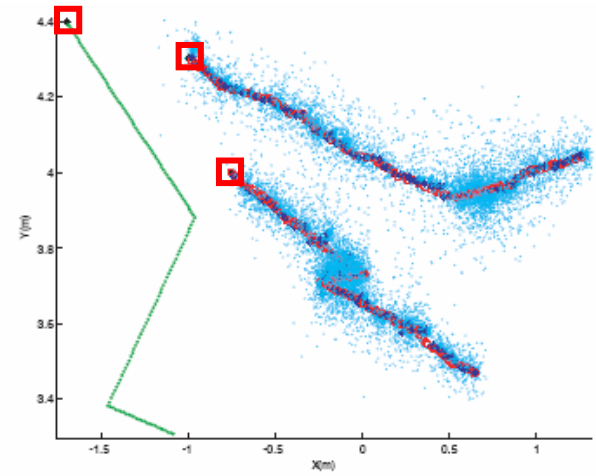


Figure (3.5) (A). L'observateur (le robot dans notre cas) se déplace suivant la ligne droite (en vert), pendant que la cible suit sa trajectoire; les gros carrés rouges sont les positions initiales du robot et de la cible. (B). Représente un zoom pour une zone de trajectoire entourée par l'ellipsoïde rouge dans la figure (A) ; les petits cercles rouges représentent la trajectoire réelle de la cible, les très petits points en bleu ciel sont les particules alors que les croix bleues représentent la trajectoire estimée.

La figure (3.5) montre l'évolution du processus de suivi en simulation. La trajectoire de l'observateur se compose de trois parties en supposant qu'il est capable de recevoir des mesures de la cible suivie pendant toute la durée du processus. La cible est manœuvrante, elle change sa direction deux fois, son mouvement le long de sa trajectoire réelle est randomisé pour compliquer les contraintes du test effectué. D'autres simulations avec d'autres choix pour la trajectoire de la cible suivie et celle de l'observateur sont montrées figure (3.6.A). La figure (3.6.B) décrit un processus de suivi pour deux cibles à la fois en respectant toujours les mêmes contraintes prises en compte dans les tests précédents.



A.



B.

Figure (3.6) (A). Une trajectoire plus compliquée, (B). Suivi de deux cibles à la fois

Toutes les simulations que nous avons réalisées ont toujours montré une bonne correspondance entre la trajectoire réelle et celle estimée en utilisant le filtre proposé (MMPF), avec un système non linéaire modélisé par une dynamique changeante (Switching Dynamic Model). La moyenne d'erreur entre les deux trajectoires est de 2 cm. Nous avons tenté d'aller plus loin en appliquant cette approche sur le robot pour faire le suivi de personnes mobiles dans son entourage pour réaliser

notre objectif de faire le (Simultaneous Mapping And Tracking). Ceci est décrit dans le paragraphe suivant.

3.5. Expérimentations

Ce paragraphe explique notre approche basée sur la grille d'occupation construite simultanément pour faire le groupage de cellules voisines occupées, la détection des groupes mobiles dans le voisinage du robot, faire le suivi pour une cible détectée. Enfin nous illustrons notre procédure qui peut gérer le processus du suivi multi-cibles avec des résultats expérimentaux.

3.5.1. Groupage de Cellules Voisines Occupées

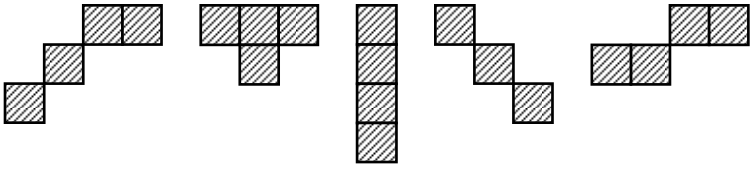


Figure (3.7). Quelques exemples des groupes considérés

Dans un environnement dynamique représenté par une grille d'occupation, le groupage de cellules voisines occupées évoluant de manière similaire dans la zone de perception du robot (figure 3.7), permet que chaque groupe soit considéré comme une seule cible représentée par les coordonnées centrales du groupe dynamique détecté. Cette étape de groupage permet de réduire le nombre de traqueurs, et compense également des mesures très bruyantes, car traiter un seul groupe est beaucoup plus robuste que le traitement de plusieurs cellules. Le fait que l'environnement contient des entités mobiles exige que les groupes qui sont toujours dans la zone d'acquisition de données soient mis à jour (au niveau de leurs formes et de leur nombre) de manière récursive effectuée à chaque balayage du laser.

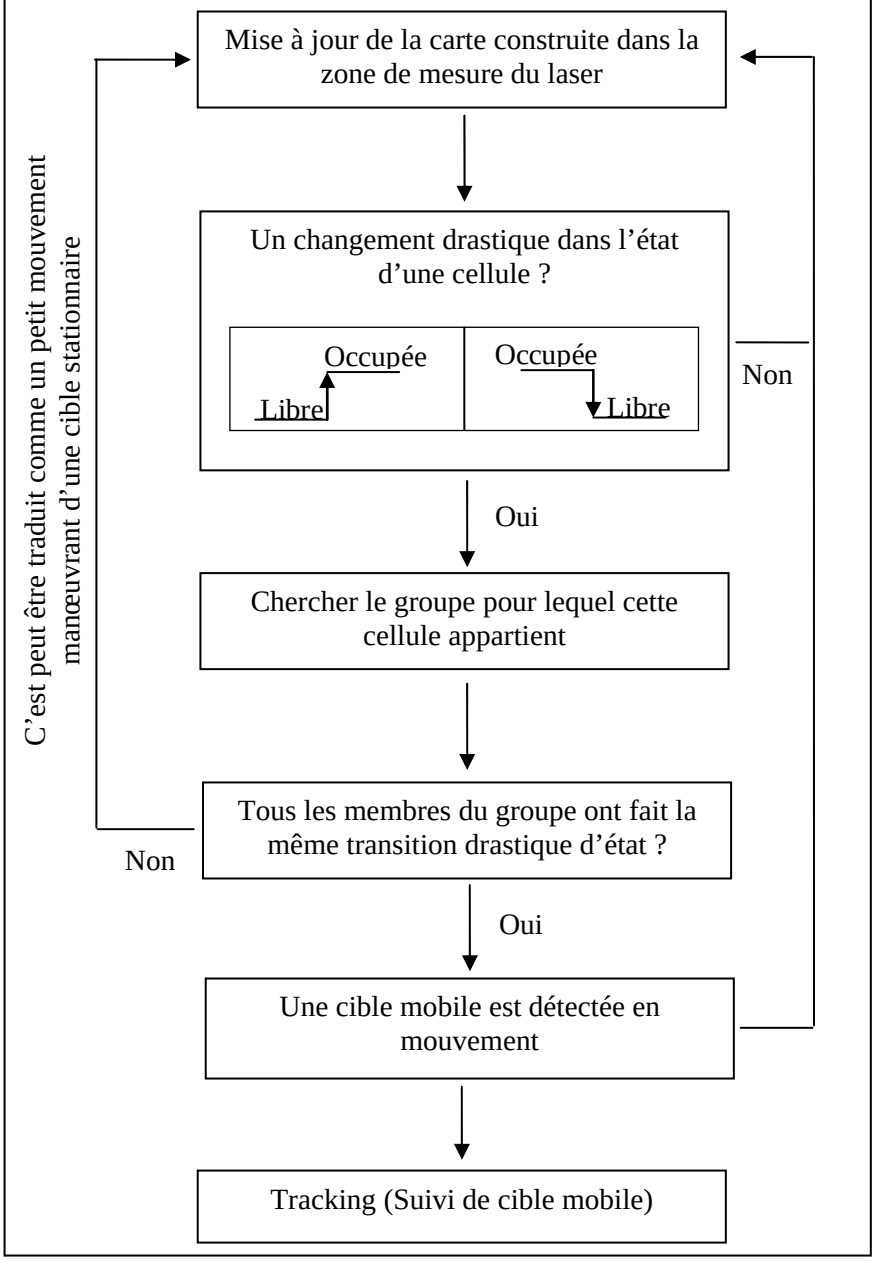
3.5.2. Détection de Cibles Mobiles

Quand le robot fait la mise à jour de sa carte, il compare l'état courant de chaque cellule évaluée avec son état à l'instant précédent. Un changement drastique de l'état d'une cellule donnée (de l'état occupé à l'état libre ou l'inverse sans passer par l'état inconnu) entre deux balayages successifs du laser sera suffisant pour déclencher une procédure de détection (tableau 7), qui cherche tout d'abord le groupe auquel cette cellule appartient, puis il évalue l'état d'occupation pour tous les membres inclus dans ce groupe. S'ils ont tous subi (entre les deux instants) le même changement d'occupation, le groupe peut être considéré comme un objet dynamique en mouvement, ses coordonnées centrales sont calculées et envoyées pour une étape de suivi. La raison pour laquelle nous avons mis la contrainte d'avoir un changement drastique et global d'état sur tous les membres qui composent un groupe vient du fait que le laser fait son balayage au niveau des jambes des personnes mobiles. Donc, pour éviter de faire la détection de petits déplacements qui peuvent caractériser le mouvement naturel des jambes d'un être humain même s'il est toujours fixe dans sa position, il a fallu prendre en compte cette condition. Notre méthode est uniquement sensible aux

vrais déplacements d'une cible mobile, mais pas pour les très petites manœuvres effectuées par une cible stationnaire.

La figure (3.8) illustre un groupe de cellules se déplaçant vers la gauche, la probabilité d'occupation des cellules qui le composent change entre les deux positions considérées dans les deux situations (1) et (2) selon le modèle du capteur déjà discuté dans le chapitre précédent. De plus, les cellules où la cible était stationnaire dans la situation (1) vont subir une transition (occupé – libre) de leurs états ce qui est suffisant pour déclencher l'étape de détection ci-dessus. Nous pouvons noté comment les cellules cachées derrière la cible ont jusqu'à l'instant *t* l'état inconnu. Dès que la cible disparaît complètement de la scène dans la troisième situation elles deviennent libres.

Tableau 6. **La détection d'une cible mobile**



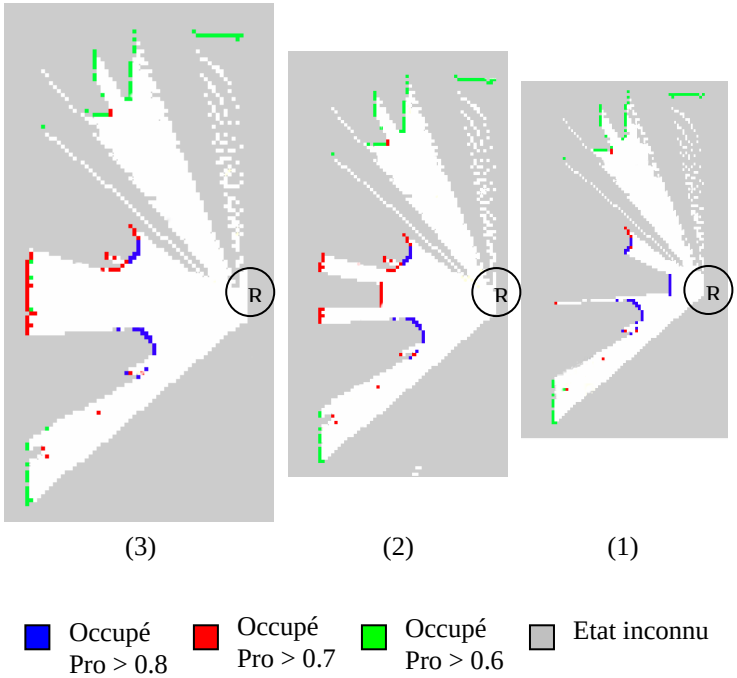


Figure (3.8). Exemple d'une grille d'occupation avec un objet se déplaçant vers la gauche. La probabilité d'occupation de cible change rigoureusement quand l'objet se déplace.

3.5.3. Suivi de Cible Mobile

Le principe de notre approche peut être représenté par deux couches virtuelles illustrées dans la figure (3.9). Nous exploitons la grille d'occupation élaborée dans une première couche pour faire la détection d'objets mobiles et l'acquisition des mesures qui doivent être fournies au filtre, et dans une deuxième couche, pour réaliser le processus de suivi déjà expliquée.

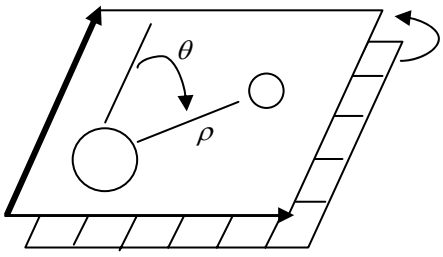


Figure (3.9). Deux couches virtuelles : la première représente la grille d'occupation construite, alors que la seconde représente le niveau où le suivi des cibles mobiles s'effectue, la géométrie (Robot –Cible) est illustrée.

Dans nos expériences, 100 particules sont tirées aléatoirement dans chacune des trois zones d'estimation illustrées dans la figure (3.4.a). Le dernier nombre de particules est choisi d'une façon expérimentale, puisqu'il nous a donné une bonne performance au niveau de la précision de la trajectoire suivie, avec un temps moyen nécessaire pour effectuer une étape complète d'estimation dans une telle trajectoire égal à 50 ms. Les figures (3.10) et (3.13) montrent une bonne adéquation (quelques centimètres) entre la trajectoire estimée et celle mesurée.

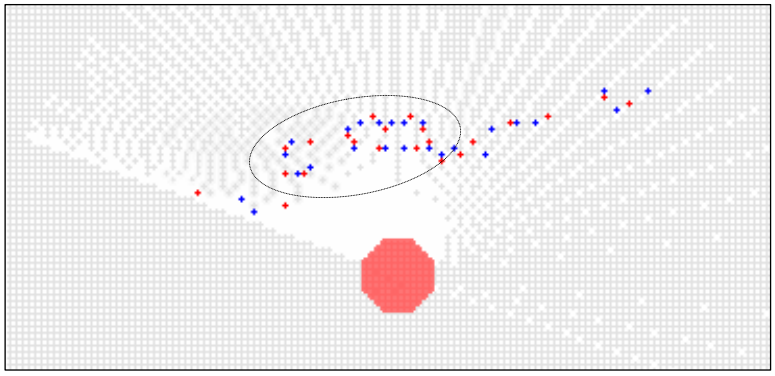
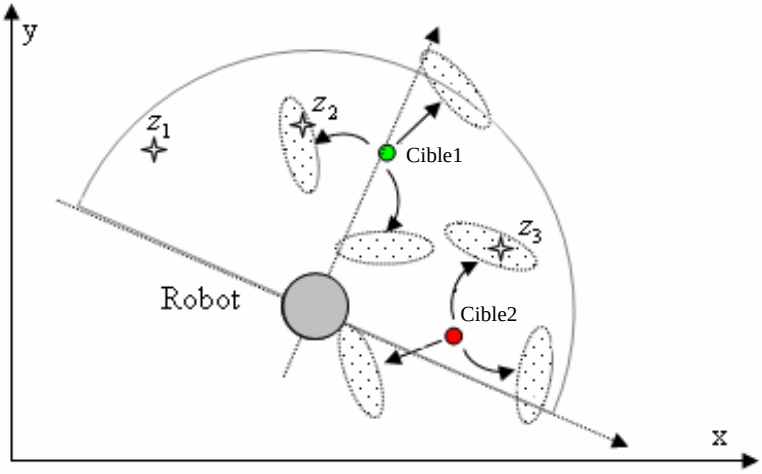


Figure (3.10). Suivi d'une personne se déplaçant à proximité du robot (cellules bleues : trajectoire mesurée ; cellules rouges : trajectoire estimée). Dans la zone entourée par l'ellipsoïde la stratégie que nous avons utilisée pour faire l'association de données, montre la capacité à attribuer les deux jambes d'une personne mobile à une seule cible, et non comme deux cibles indépendants.

Suivi Multi Cibles (Association de données)



A.

Cibles Actives	Balayage No
……	……..
Cible1, Cible2	31110
Cible1, Cible2	31111
Cible1, Cible2, Cible3	31112
……..	……

B.

Figure (3.11). (A) Deux cibles suivies à proximité du robot, la première mesure a une petite probabilité, insuffisante pour qu'elle soit attribuée; dans le balayage suivant elle sera donc considérée comme une nouvelle cible entrant dans la scène. La deuxième mesure reçue z_2 dans le même balayage est attribuée à la cible 1 car la particule la plus probable lui appartient, et la troisième mesure z_3 est attribuée à la cible 2 en utilisant le même principe. (B) Un exemple illustrant des balayages successifs du laser et les cibles qui sont considérées actives dans chaque itération.

Le suivi peut être fait pendant que le robot construit sa propre carte (grille d'occupation), ce qui permet de détecter les cibles mobiles et de fournir leurs coordonnées centrales au traqueur. Dans la zone de perception du robot, le laser fait son balayage de 180 degrés avec une fréquence de 20 Hertz. L'algorithme montré dans le tableau (6) peut distinguer les groupes mobiles de ceux qui sont statiques. Pour chaque nouvelle détection, il faut lancer une étape de vérification pour savoir s'il y a

des cibles qui sont déjà détectées dans les balayages précédents et toujours existantes dans la zone d'acquisition de données (ce sont des cibles actives). Si c'est vraiment le cas, figure (3.11), une étape de pondération doit être effectuée sur toutes les particules diffusées dans les régions d'estimation pour toutes les cibles impliquées afin d'attribuer la mesure reçue à une cible active qui est le propriétaire de la particule la plus lourde.

A ce niveau là, il faut faire attention au fait qu'une telle particule déterminée peut avoir, d'un point de vue comparatif, un poids potentiel par rapport aux autres particules, alors que d'un point de vue absolu, elle représente une petite probabilité, insuffisante pour que la mesure reçue ne puisse pas être attribuée (dans ce cas une méthode simple de seuillage peut être acceptée). Une telle mesure, dans ce cas, sera considérée comme une première détection provoquée par une nouvelle cible entrant à proximité du robot. S'il y a encore des mesures reçues dans le même balayage, d'autres étapes de pondération doivent être faites pour toutes les cibles actives pour attribuer chaque mesure courante à sa propre cible. Dans le cas où une cible n'a plus de mesures à attribuer pour deux balayages successifs, elle sera éliminée. Une fois le voisinage du robot libre, la liste des cibles suivies doit être re-initialisée. En fait, l'application de cette stratégie nous a montré une capacité imprévue a priori. Elle est capable de résoudre le problème du mouvement de deux jambes d'une personne, pour qu'elles soient considérées comme une seule cible mobile, figure (3.10).

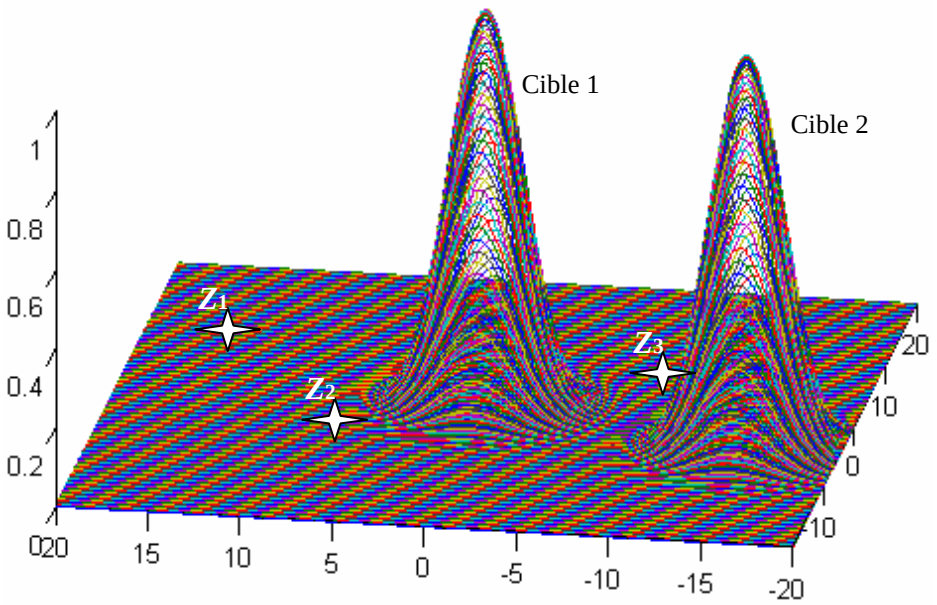


Figure (3.12). Représentation de la figure (3.11) en 3D.

La figure (3.13) représente l'évolution d'une expérience développée dans notre laboratoire. Quand le robot se déplace pour construire sa carte, il détecte et suit simultanément les trajectoires de deux personnes se déplaçant dans sa proximité (numéro un et numéro deux). Et dans l'autre partie de sa trajectoire, le robot suit une nouvelle cible (numéro trois). Les cellules vertes représentent la trajectoire mesurée ; les cellules rouges représentent la trajectoire estimée et les cellules bleues sont des parties statiques dans l'environnement. Les cellules occupées par le robot pendant son mouvement sont illustrées par les cellules rouges.

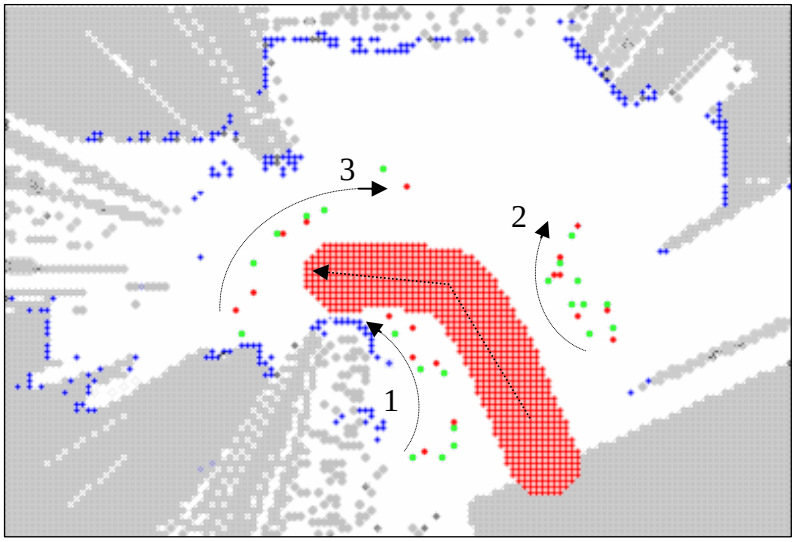
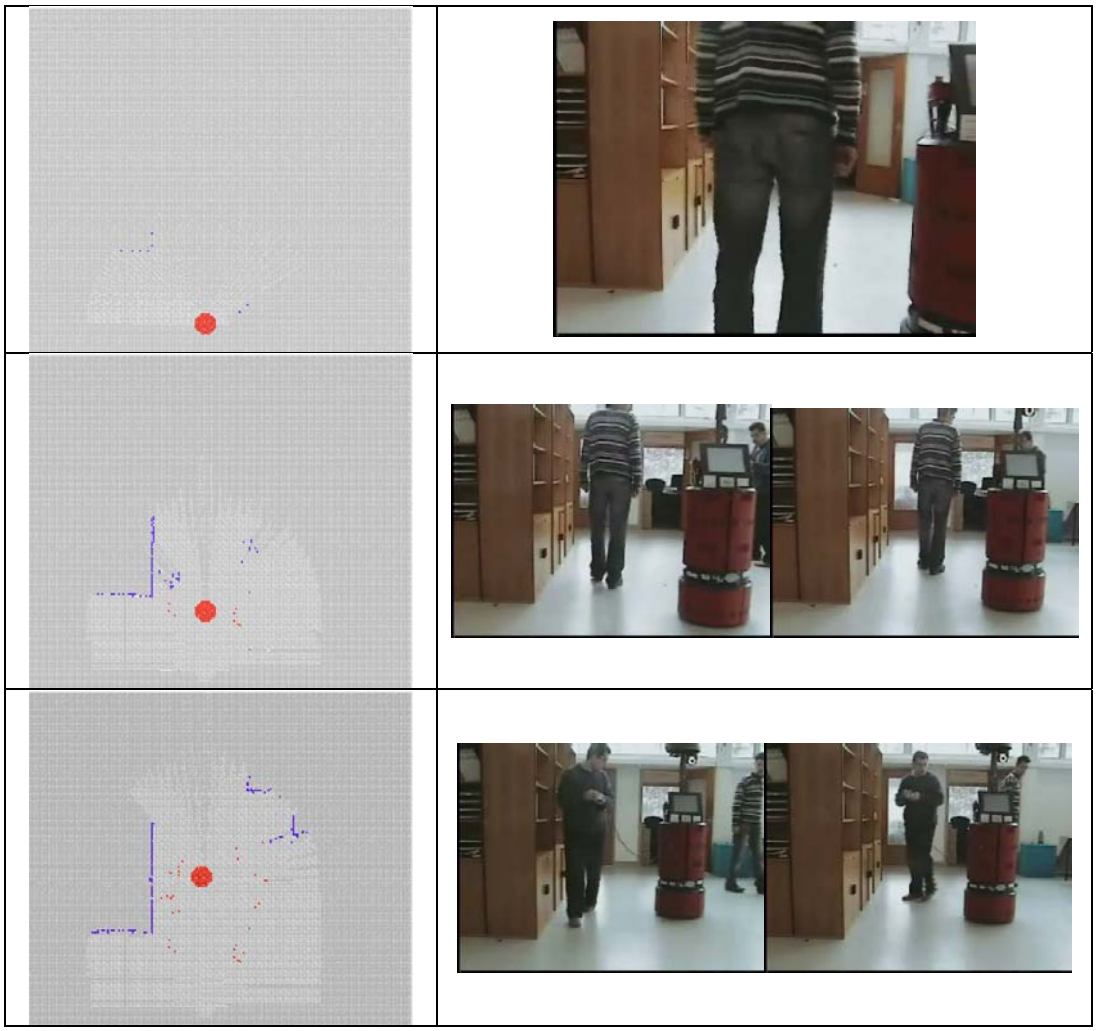


Figure (3.13). Suivi Multi Cible. Trois trajectoires peuvent être distinguées (1, 2 et 3). Les parties en bleu sont statiques et incluses dans la carte élaborée. Les cellules vertes représentent les trajectoires mesurées et les cellules rouges représentent les trajectoires estimées.



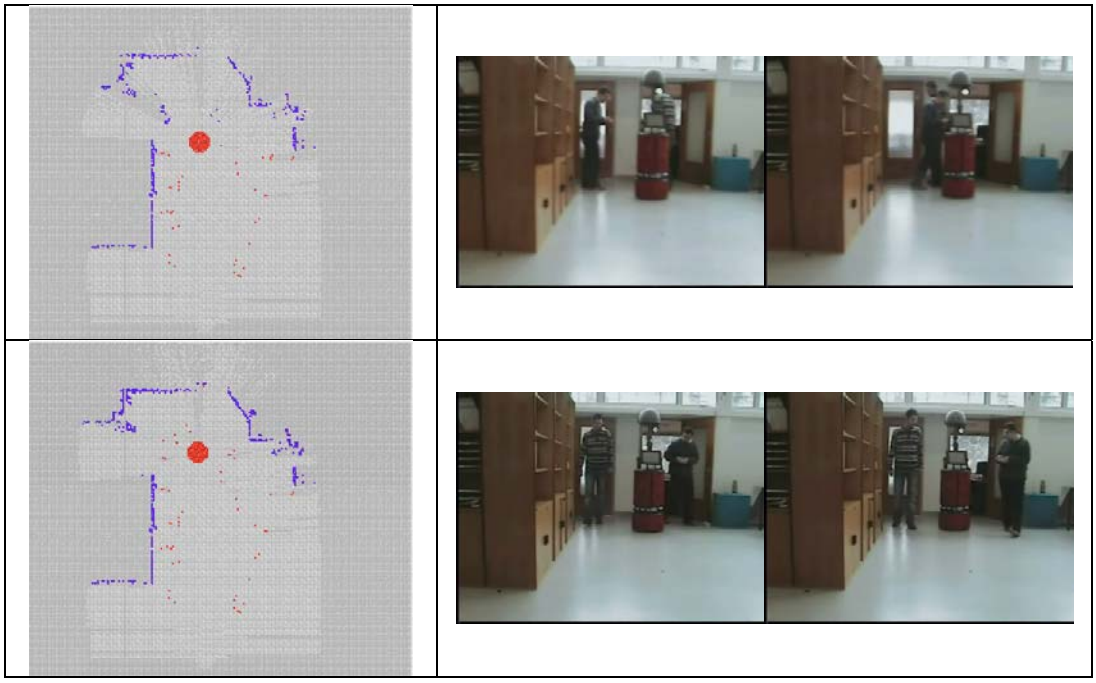


Figure (3.14). Cartographie et suivi simultanés pendant le mouvement du robot dans son environnement. Les traces des trajectoires des cibles mobiles sont distinguées (en rouge) de la carte construite (en bleu). Lors d'une première perception, la détection est toujours exprimée en cellules bleues, qui deviennent rouges à partir des perceptions suivantes s'il s'agit de cibles mobiles.

3.6. Conclusion et Discussion

Dans ce chapitre nous avons décrit géométriquement le système (cible - observateur), nous avons proposé un modèle pour la dynamique du système, puis nous avons présenté une approche modifiée pour faire le suivi d'objets mobiles pendant que le robot construit une carte de son environnement représentée par une grille d'incertitude. Nous avons montré de façon détaillée une technique de filtrage particulaire (MMPF) qui convient bien au modèle de dynamique proposé. En suite, nos stratégies pour regrouper les cellules voisines et occupées sous forme de « masses » ou de « segments » sont discutées. La détection des groupes mobiles et le suivi multi cibles sont également illustrés. Finalement nous avons discuté quelques expériences réalisées dans notre laboratoire et illustrées par les figures correspondantes.

L'approche suivie dans ce chapitre, montre souvent une bonne performance avec quelques exceptions. Par exemple, dans un environnement très peuplé où il y a beaucoup de personnes se déplaçant dans la zone de mesure, la procédure proposée pour faire l'association de données échoue parfois et spécialement avec des cibles qui ont des trajectoires croisées. En plus, une erreur de localisation du robot pendant son mouvement, égale au moins à la moitié de la dimension d'une cellule (2.5 cm), va déclencher une fausse étape de détection et de suivi pour des segments qui sont en réalité des parties statiques de la carte élaborée. Pour essayer d'améliorer le comportement de l'approche suivie, nous avons décidé de fusionner une source visuelle avec le laser pour améliorer la qualité de l'information sensorielle afin que le robot appréhende mieux son univers, et qu'il soit mieux capable de détecter des cibles mobiles. C'est ce que nous allons présenter en détails dans le chapitre suivant où nous proposons d'intégrer un système visuel catadioptrique.

CHAPITRE IV

Détection Visuelle d'Objets Mobiles
(En Utilisant une Caméra Catadioptrique)

4.1. Introduction

4.2. Caméra Catadioptrique
- **4.2.1. Combinaison miroir parabolique / objectif télécentrique**
- **4.2.2. Modèle de Projection**
- **4.2.3. Redressement Cylindrique d'une Image Panoramique**

4.3. Histogramme

4.4. Extraction du Fond d'image
- **4.4.1. La Matrice de Convolution**
- **4.4.2. Gaussian Average Operator**
- **4.4.3. Le Médian**
  - **4.4.3.1. Filtrer une Image en Utilisant (Median Operator)**
  - **4.4.3.2. Extraction du Fond d'image en Utilisant (Temporal Median Operator)**

4.5. Détecter une Cible Dynamique en Utilisant le Flot Optique
Une Approche Différentielle

4.6. Déterminer la Direction de Cible Détectée

4.7. Déterminer les Directions de Plusieurs Cibles

4.8. Détection des Cibles Mobiles à Partir d'Images Panoramiques non Déployées (Images Brutes)

4.9. Comparaison Entre les Deux Techniques Suivies Pour Déterminer les Directions des Cibles Détectées

4.10. Conclusion

4.1. Introduction :

L'exploitation des capteurs visuels est de plus en plus répandue dans différentes applications (systèmes de surveillance et de sécurité, robots mobiles, voitures intelligentes, systèmes du guidage de missiles orientés, …etc.). L'image est en effet une source d'information beaucoup plus riche que la télémétrie. Elle nécessite cependant des traitements plus complexes et plus coûteux en temps de calcul.

Dans ce chapitre, nous proposerons l'exploitation d'une image panoramique pour améliorer la détection et le suivi de cibles mobiles au prix d'une réduction de la résolution. Nous commençons par une brève présentation de la caméra catadioptrique. Ensuite, le modèle de projection et le redressement cylindrique d'une image panoramique sont discutées. Le médian qui représente un filtre efficace pour améliorer la qualité d'une image bruitée est exploité dans sa version (Temporal Median Operator) pour extraire le fond de scène (background pixels) à partir de séquence d'images acquises de la même position. Les cibles mobiles (les personnes) sont distinguées comme « foreground pixels » dans une image binaire construite en utilisant une approche de flot optique modifiée qui prend en compte les changements possibles de l'intensité de la source lumineuse. Les directions des cibles détectées sont déterminées en utilisant deux stratégies. La première utilise les images panoramiques déployées; elle cherche les régions les plus probables dans l'image binaire élaborée en calculant un histogramme sur ses colonnes. La deuxième approche est applicable pour les images panoramiques brutes; elle regroupe les « foreground pixels » en utilisant une technique qui produit une nouvelle intensité d'un pixel selon les intensités de ses voisins. Une discussion conclut le chapitre.

4.2. Camera Catadioptrique :

Grâce à son champ de vision de 360°, une caméra catadioptrique est très avantageuse pour certaines applications comme le suivi visuel de cible sur une longue distance ou la localisation par rapport à des amers visuels, en comparaison avec une caméra perspective.

Un système catadioptrique est une combinaison d'un miroir (par exemple parabolique, hyperbolique ou elliptique) avec un système de capture d'images classique. Plusieurs approches existent pour obtenir des images avec un champ de vision étendu:

1. Utilisation d'images multiples.
2. Utilisation d'objectifs spéciaux.
3. Utilisation de miroirs convexes.

Le choix entre les trois approches citées résulte du compromis résolution - vitesse d'acquisition. L'utilisation d'images multiples a l'avantage de fournir une résolution potentiellement grande, mais au prix du temps d'acquisition. Certaines applications telles que la navigation du robot, nécessitent un temps d'acquisition quasi instantané pour informer rapidement le système sur l'état de son environnement et ne requièrent pas nécessairement une haute résolution d'image; pour cela les caméras catadioptriques sont plus utilisées.

Parmi les différents systèmes qui emploient les miroirs convexes, ceux qui présentent un centre de projection unique (figure 4.1) ont l'avantage de permettre la reconstruction d'images selon différents types de projection et sur différents types de surfaces. Une caméra catadioptrique munie d'un miroir hyperbolique et d'un objectif conventionnel dont le centre optique est situé au foyer de l'hyperbole a cette propriété, mais il n'est pas facile de garantir un montage mécanique qui satisfasse cela. Les miroirs paraboliques permettent de construire une caméra avec un centre de projection unique sans avoir de contrainte forte sur la précision mécanique de l'assemblage. D'autre part, la connaissance des paramètres géométriques de formation d'une image panoramique permet d'en générer une autre en projection perspective.

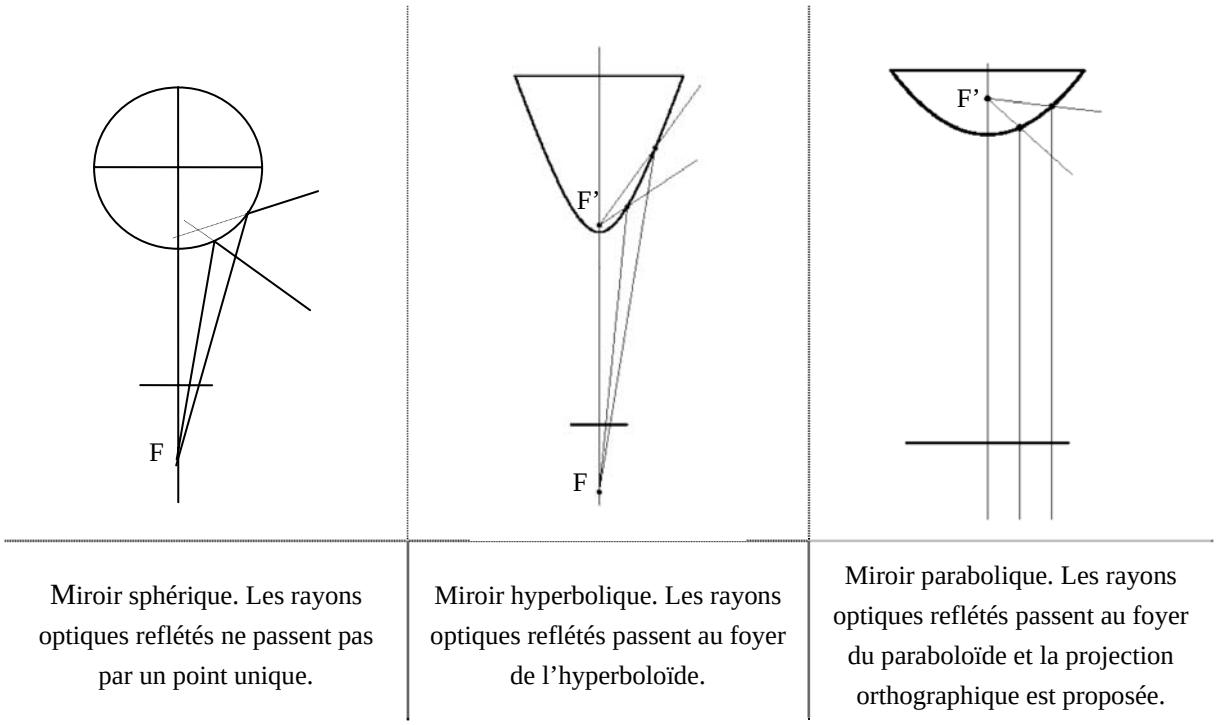
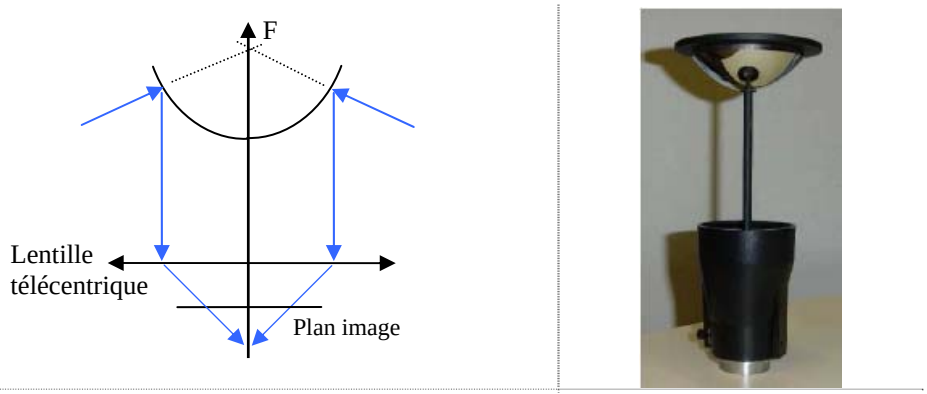


Figure (4. 1), trois types de miroirs convexes. Un miroir hyperbolique ou un miroir parabolique présentent un centre de projection unique.

4.2.1. Combinaison miroir parabolique / objectif télécentrique

Dans notre application, nous utilisons une caméra omnidirectionnelle (figure 4.2) composée d'une caméra conventionnelle, d'un miroir parabolique et d'une lentille télécentrique qui est une lentille spéciale formée d'éléments multiples qui permettent de sélectionner un faisceau lumineux de rayons parallèles à l'axe optique de la lentille. Avec une telle lentille alignée sur l'axe du miroir, seuls les rayons lumineux incidents qui arrivent sur le miroir et qui traversent le foyer du paraboloïde sont transmis. Une lentille télécentrique permet une relative flexibilité sur la position du jeu (lentille, miroir), et élimine les réflexions internes du cylindre ou de la sphère qui supporte le miroir.



Tous les rayons traversent virtuellement le foyer du miroir. La lentille télécentrique sélectionne seulement les rayons parallèles à l'axe du miroir.

Caméra omnidirectionnelle avec miroir parabolique et lentille télécentrique.

Figure (4. 2), Le capteur omnidirectionnel utilisé.

4.2.2. Modèle de Projection

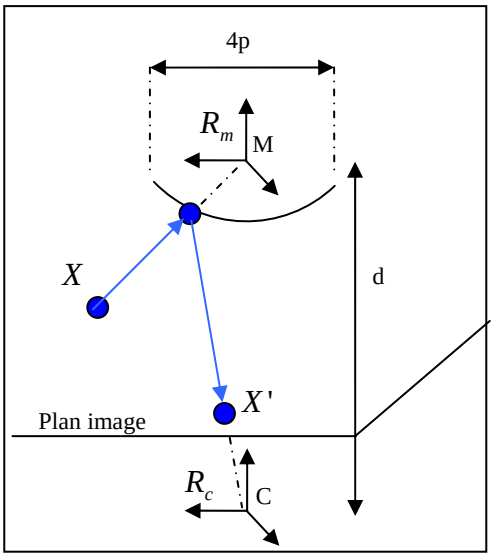


Figure (4. 3), Caméra catadioptrique constituée d'un miroir parabolique et d'une caméra classique.

	Parabolique	Hyperbolique	Elliptique	Plan
(ξ, ψ)	(1,1+2*p*)	$\left(\frac{d}{\sqrt{d^2+4p^2}}, \frac{d+2p}{\sqrt{d^2+4p^2}}\right)$	$\left(\frac{d}{\sqrt{d^2+4p^2}}, \frac{d-2p}{\sqrt{d^2+4p^2}}\right)$	(0,1)

Tableau (4.1), les paramètres (ξ, ψ) changent selon le type du miroir d'un système catadioptrique. d et $4p$ illustrés dans la figure (4.3), sont respectivement la distance entre les deux repères et le (Latus Rectum).

Nous pouvons donc constater, qu'un centre de projection unique est une propriété intéressante car cela exige que les droites joignant un point de l'espace et sa projection sur le miroir passent par un point central unique de projection. Donc, chaque pixel sur le plan image mesure la luminance du rayon qui passe par le point central dans une direction particulière et connue. Baker et al. [12] montrent qu'un système catadioptrique peut être construit à partir d'un miroir parabolique avec une caméra orthographique ou un miroir (hyperbolique, elliptique ou plan) avec une caméra perspective. Geyer et al [50] proposent un modèle unifié de projection pour les caméras panoramiques à point central unique. Ces caméras sont modélisées par une projection centrale sur une sphère suivie d'une autre projection perspective sur le plan image. Ce modèle générique est réglable [62] par un couple de paramètres (ξ, ψ) selon le type de miroir utilisé, (tableau 4.1).

Soit R_c et R_m deux repères attachés à la caméra et au miroir respectivement. Les centres C et M sont respectivement le centre optique et le point principal. Soit X un point de l'espace de coordonnées $X = (X, Y, Z)$ dans le repère R_m. D'après le modèle de projection proposé dans [50] : la projection de point X dans le plan d'image est un point $X' = (x, y, 1)$ avec :

$$X' = KMf(X) \qquad \textbf{4. 1}$$

K est une matrice triangulaire contenant les paramètres de calibrage interne de la caméra,

$$M = \begin{pmatrix} \psi - \xi & 0 & 0 \\ 0 & \psi - \xi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Et :

$$f(X) = \begin{pmatrix} \dfrac{X}{Z + \xi\sqrt{X^2 + Y^2 + Z^2}} & \dfrac{Y}{Z + \xi\sqrt{X^2 + Y^2 + Z^2}} & 1 \end{pmatrix}^T$$

4.2.3. Redressement Cylindrique d'une Image Panoramique

Un point de la surface du miroir peut être défini par rapport au pixel de l'image par la relation suivante :

$$p(x, y, z) = \left[u, v, \frac{h^2 - u^2 - v^2}{2h} \right] \qquad \textbf{4. 2}$$

h : est un paramètre dont la valeur dépend du miroir parabolique, et (u, v) représentent la position du pixel par rapport à l'axe du miroir.

A partir de la dernière équation, une relation entre un point sur une surface cylindrique entourant le paraboloïde et un pixel de l'image peut être écrite comme suit :

$$p(\Theta, Z) = \left[\arctan\left(\frac{u}{v}\right), \frac{R(h^2 - u^2 - v^2)}{2h\sqrt{u^2 + v^2}} \right] \qquad \textbf{4. 3}$$

R est le rayon du cylindre de projection.

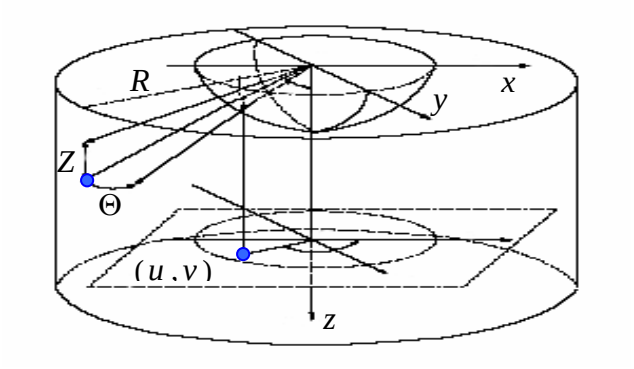


Figure (4. 4), Construction d'un point sur un cylindre à partir d'un pixel d'une image panoramique [51]. Θ et Z sont calculés par rapport aux cordonnées locale de la caméra (x, y, z).

Pour déployer une image panoramique vers un autre cylindrique, nous avons exploité une fonction qui fait partie d'une bibliothèque de fonctions de traitement d'image (JAFAR) développée au LAAS. Le calibrage de la caméra panoramique est fait en utilisant (OcamCalib Toolbox) [96]. Mais, la projection finale sur le plan image exige d'effectuer une étape d'interpolation qui suit le principe que nous proposons dans la figure (4.5). L'intensité du pixel inséré dans le plan image sera générée à partir des pixels contigus dans le plan cylindrique en utilisant l'algorithme du tableau (4.2).

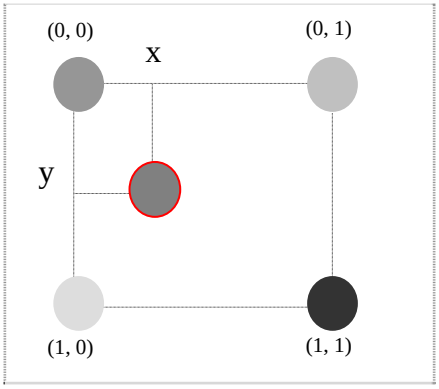


Figure (4. 5), Le principe suivi pour faire l'interpolation. Le pixel entouré en rouge représente le pixel généré dans le plan image, alors que les autres sont les pixels dans le plan cylindrique.

Ce calcul considère que les distances entre les pixels dans le plan cylindrique sont unitaires, leur arrangement est matriciel. A chaque pixel parmi les quatre qui forment un carré autour de celui généré (figure 4.5), il faut attribuer un poids qui représente le degré de sa participation à décider le niveau de gris du pixel produit selon sa position relative par rapport à chacun entre eux.

$$\{P(0,0),\ P(0,1),\ P(1,0),\ P(1,1)\}$$
$$IPI = 0$$
$$for(j = 0:1)$$
$$\quad for(i = 0:1)$$
$$\quad poids(j,i) = (1 - abs(x - i)) * (1 - abs(y - j))$$
$$\quad IPI = IPI + poids(j,i) * pixel(j,i)$$
$$\quad end$$
$$end$$
$$return(IPI)$$

Tableau (4.2). *IPI*: (Interpolated Pixel's Intensity), (x , y) la position relative de pixel généré par rapport à chacun des quatre qui l'entourent.

Exemple : Pour calculer l'intensité d'un pixel inséré dans un plan d'image à une position relative ($x = 0.3$, $y = 0.2$) par rapport au pixel *P(0,0),* et à partir de quatre pixels dans un plan cylindrique ayant les intensités suivantes : {*P(0,0)* = 200, *P(0,1)* = 180, *P(1,0)*=100, *P(1,1)*=80}. En utilisant l'algorithme précédent illustré dans le tableau (4.2), les poids des quatre pixels seront : { $poids(0,0) = 0.56$, $poids(0,1) = 0.24$, $poids(1,0) = 0.14$, $poids(1,1) = 0.06$ }.
L'intensité calculée du pixel résultant sera 174.

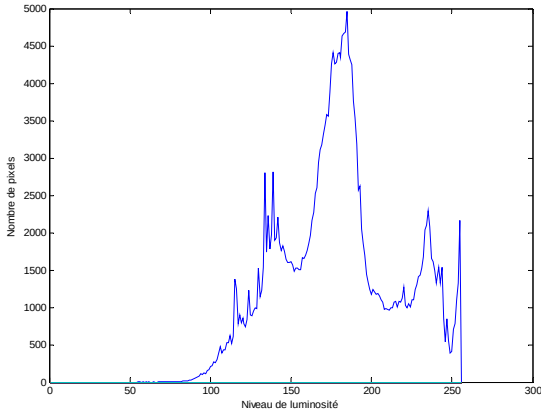
4.3. Histogramme

L'histogramme de l'intensité de luminance montre la distribution de différents niveaux de luminosité dans une image en comptant du nombre de pixels qui ont une certaine valeur entre 0 (noir) et 255 (blanc), pour un pixel de 8 bits. La figure (4.6) montre deux images avec leurs histogrammes. Nous pouvons noter que l'image (4.6.A) est très éclairée et pour cela son histogramme ne couvre pas toute la plage de 256 niveaux de luminosité. Pour régler une telle situation nous pouvons appliquer une étape de normalisation comme suit :

$$N_{x,y} = \frac{N_{\max} - N_{\min}}{A_{\max} - A_{\min}} (A_{x,y} - A_{\min}) + N_{\min} \qquad \textbf{4. 4}$$

$N_{x,y}$ est l'intensité du pixel dans la nouvelle image. La plage des niveaux de luminosité dans l'ancienne image (4.6.A) commence à un niveau minimal $A_{\min}$ et se termine au niveau maximal $A_{\max}$. Pour que l'histogramme couvre toute la plage dans la nouvelle image N , il faut choisir $N_{\min} = 0$ et $N_{\max} = 255$ qui représentent le niveau minimal et maximal respectivement.

A.



B.

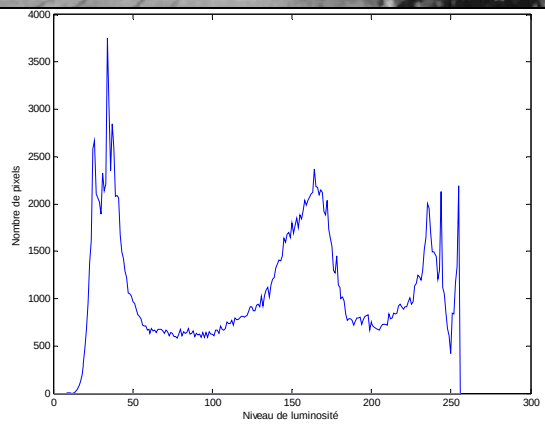


Figure (4. 6), A. histogramme pour une image qui ne couvre pas toute la plage de 256 niveaux d'intensité lumineuse, B. une image normalisée ; son histogramme couvre toute la plage.

4.4. Traitement d'image (Extraction du Fond d'image)

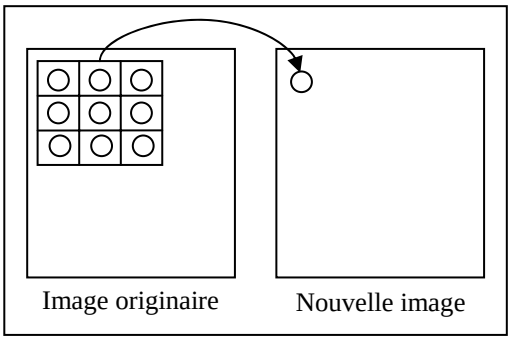
Dans le domaine du traitement d'image, plusieurs opérations sur le voisinage du pixel peuvent être applicables (Gaussian Average operator, Laplacian opérator, Median opertator,..etc) pour extraire une information utile (e.g. des segments horizontaux ou verticaux) ou avoir une qualité améliorée d'une image donnée. Ces opérateurs emploient généralement des matrices de convolution.

4.4.1. Matrice de Convolution

Une matrice de convolution est normalement une matrice carrée de taille 3x3 ou 5x5 (figure 4.7). Le mécanisme avec lequel cette méthode peut être exploitée est de scanner successivement tous les pixels dans l'image originaire. Pour chaque pixel appelé « pixel initial », il faut multiplier sa valeur et celles de 8 pixels (pour une matrice de taille 3x3) qui l'entourent par les valeurs correspondantes dans le noyau (matrice des cœfficients de pondération) dont les valeurs varient selon l'opérateur utilisé. Puis il faut additionner l'ensemble des résultats de multiplication (équation 4.5) pour qu'un nouveau pixel soit finalement produit :

$$\begin{aligned} N_{x,y} = {} & P_0 * A_{x-1,y-1} + P_1 * A_{x,y-1} + P_2 * A_{x+1,y-1} + \\ & P_3 * A_{x-1,y} + P_4 * A_{x,y} + P_5 * A_{x+1,y} + \\ & P_6 * A_{x-1,y+1} + P_7 * A_{x,y+1} + P_8 * A_{x+1,y+1} \end{aligned} \qquad \textbf{4. 5}$$

$N_{x,y}$: intensité du pixel dans la nouvelle image, $A_{x,y}$: intensité du pixel dans l'image d'origine.



P_0	P_1	P_2
P_3	P_4	P_5
P_6	P_7	P_8

Le noyau : Cœfficients de pondération

Figure (4. 7), une matrice de convolution appliquée sur un groupe de pixels pour calculer l'intensité du nouveau pixel dans une nouvelle image. Les valeurs des coefficients de pondération entre P_0 et P_8 dépendent de l'opérateur utilisé.

4.4.2. Gaussian Average Operator

Cet opérateur se représente comme un rectificateur optimal qui emploie la matrice de convolution pour nettoyer efficacement le bruit d'une image numérique. Afin de déterminer les coefficients de pondération, montrées dans la figure (4, 7) il faut utiliser l'algorithme suivant :

$$\{winsize,\ \sigma\}$$
$$centre = (winsize/2)+1$$
$$sum = 0$$
$$for\ i = 1 : winsize$$
$$\quad for\ j = 1 : winsize$$
$$\quad template(j,i) = \exp - \left(\frac{(j-centre)^2 + (i-centre)^2}{2\,\sigma^2} \right)$$
$$\quad sum = sum + template(j,i)$$
$$\quad endfor$$
$$endfor$$
$$template = template / sum$$
$$return(template)$$

Tableau (4.3), détermination des coefficients de pondération de l'opérateur Gaussien.

L'entrée de l'algorithme illustrée dans le tableau (4.3) est la taille de la matrice de convolution ($winsize$) qui doit garder une taille impaire (e.g. (3x3), (5x5), (7x7),….etc.) pour garantir qu'un seul pixel va être à chaque fois le centre de matrice carrée, alors que (σ) représente la valeur de variance choisie a priori par l'utilisateur.

Après l'initialisation des deux variables, l'algorithme calcule les valeurs des coefficients selon l'équation suivante :

$$g(x,y) = \exp\ - \left(\frac{x^2 + y^2}{2\sigma^2} \right) \qquad \textbf{4. 6}$$

(x, y) : Sont les coordonnées relatives d'une case dans la matrice par rapport à la case centrale.

Ensuite, une étape finale de normalisation est importante. Par exemple, une matrice de coefficients de pondération de taille (5x5) avec une variance σ =1, sera la suivante :

0.02	0.08	0.14	0.08	0.02
0.08	0.37	0.61	0.37	0.08
0.14	0.61	1.0	0.61	0.14
0.08	0.37	0.61	0.37	0.08
0.02	0.08	0.14	0.08	0.02

Une matrice des coefficients de pondération de l'opérateur Gaussien

A.

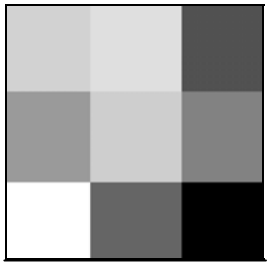
B.

Figure (4. 8), A. Image bruitée. B. La même image filtrée en utilisant l'opérateur Gaussien avec une matrice de convolution de taille (5x5).

4.4.3. Le Médian

Mettre les valeurs des pixels voisins dans l'ordre

191	193	166
180	190	175
200	170	150



150	166	170	175	180	190	191	193	200

Figure (4. 9), Une matrice de convolution centrée sur un pixel d'intensité lumineuse (190), le niveau de gris du pixel et ceux des pixels qui l'entourent sont illustrés. Le médian est le membre central, ici déterminé à une intensité égale à (180).

C'est la valeur centrale extraite d'une série des valeurs ordonnées. D'où, chercher le médian dans une matrice de convolution représente une fonction plus simple à appliquer que d'autres opérateurs et moins coûteuse en temps de calcul, car il n'est pas nécessaire d'effectuer le calcul de l'équation (4.5). Il faut simplement que les pixels qui remplissent la matrice soient ordonnés afin de déterminer le médian, figure (4.9). L'utilisation d'une matrice de taille impaire garantit qu'un seul pixel ait l'ordre du centre. Sinon, le médian peut être calculé comme la moyenne de deux valeurs centrales.

4.4.3.1 Filtrer une Image en Utilisant (Median Operator)

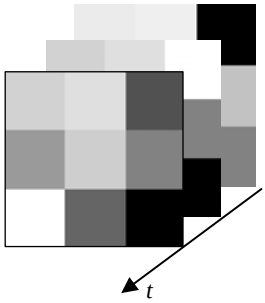
Filtrer une image en utilisant le médian (figure 4.10) paraît classique et efficace à la fois pour nettoyer une image de son bruit qui peut dégrader sa qualité à cause d'erreurs de décodage numérique dans un système de transmission d'images.

A.

B.

Figure (4. 10), A. Image bruitée, B. La même image après le filtrage en appliquant l'opérateur médian (taille de matrice de convolution (3x3)).

4.4.3.2 Extraire le Fond d'image en Utilisant le (Temporal Median Operator)



$$N_{x,y} = Méd(A^0_{x,y}, A^1_{x,y}, ..., A^n_{x,y}) \qquad \textbf{4. 7}$$

n : nombre de toutes les images.

Figure (4. 11), choisir le médian parmi les pixels qui ont les mêmes coordonnées matricielles dans les images successives.

A.

B.

C.

D.

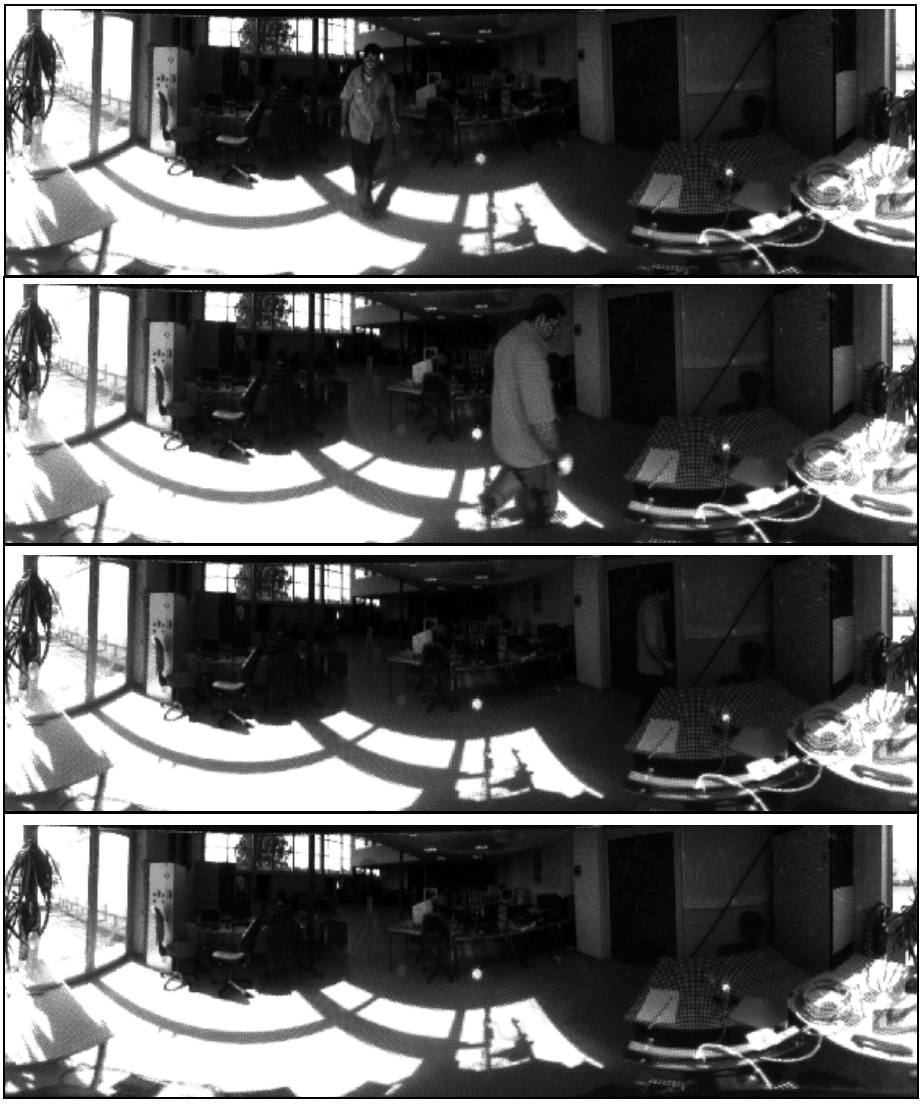


Figure (4. 12), Les images A., B., et C. sont trois images successives acquises à un intervalle de temps de 5 secondes, l'image D. représente le fond de scène extrait en utilisant (temporal median operator).

Utiliser un capteur visuel pour détecter les objets mobiles dans une scène nécessite d'en extraire les parties statiques (le fond de scène) dans une première étape afin d'isoler les cibles dynamiques dans une deuxième. A partir d'une séquence d'images acquises de la même position à un intervalle de temps Δt (constant), il est possible d'extraire un modèle décrivant le fond de scène en cherchant le médian parmi les pixels qui ont les mêmes coordonnées matricielles dans toutes les images successives, figure (4.11). Plus la scène est occupée plus Δt doit être grand. Cet opérateur choisit seulement les composants du modèle global qui sont corrélés. Une mise à jour répétitive (ou presque répétitive) de l'extraction du fond de scène en utilisant cette approche résout le problème de la relocalisation d'objets qui appartiennent vraiment aux parties statiques afin d'améliorer le niveau de la robustesse du modèle extrait. Dans notre application, le modèle du fond de scène est extrait à partir de trois images successives, comme illustré dans la figure (4.12).

4.5. Détecter une Cible Dynamique en Utilisant le Flot Optique

Une séquence d'images acquises à des instants différents dans une scène dynamique, permet de détecter le mouvement de n'importe quelle entité mobile, qui sera présentée comme un groupe de pixels se déplaçant avec le temps. L'idée ici est de profiter du calcul proposé par l'approche du flot optique pour déterminer la vitesse du pixel entre une image courante et un fond de scène déjà connu ou extrait. L'intensité lumineuse des pixels dans l'image résultante sera l'interprétation de l'amplitude de leurs vitesses. Après une étape de seuillage qui considère que chaque pixel ayant une vitesse nulle sera noir alors que les autres qui ont des vitesses différentes mais plus que la valeur du seuil seront au niveau (255 ou blanc), une image binaire sera construite.

La relation entre les pixels dans une zone d'image peut être écrite comme suit [88]:

$$P(t+1)_{x+\delta x, y+\delta y} = P(t)_{x,y} + H(t)_{x,y} \qquad \textbf{4. 8}$$

$H(t)_{x,y}$ compense la différence d'intensité entre deux images, et $(\delta x, \delta y)$ représente le déplacement du pixel dans l'intervalle $[t, t+1]$. Donc, l'intensité du pixel situé à l'instant $(t+1)$ dans la position $(x+\delta x, y+\delta y)$ est égale à celle du pixel situé à l'instant (t) dans la position (x, y) affectée par un petit changement qui vient d'une différence (entre les deux images dans lesquelles le pixel est apparu) provoquée par plusieurs phénomènes physiques possibles (e.g. changement d'illumination, réflexion spéculaire, …etc.).

Une Approche Différentielle

Cette approche se focalise sur le changement différentiel de valeur du pixel. Des travaux qui ont employé le flot optique (Horn et al. [58] ; Nagel [86] ; Barron et al. [14]), considèrent l'hypothèse que l'intensité du pixel dans une image est toujours constante [88]. Donc : $H(t)_{x,y} \approx 0$.

Dans notre travail, nous avons un problème de détection d'objets mobiles dans une application robotique qui représente une application qui peut être effectuée sur une longue durée pendant laquelle plusieurs phénomènes physiques peuvent avoir lieu. Pour cela, nous avons négligé cette hypothèse afin d'avoir une approche qui prend en compte tous les effets physiques possibles, ce qui

permet en conséquence de réduire leurs influences indésirables sur la qualité de l'information déduite à partir de l'image traitée.

En utilisant un développement de *Taylor* au premier ordre, le terme $P(t+1)_{x+\delta x, y+\delta y}$ peut être exprimé par l'équation suivante :

$$P(t+1)_{x+\delta x, y+\delta y} = P(t)_{x,y} + \delta x \frac{\partial P(t)_{x,y}}{\partial x} + \delta y \frac{\partial P(t)_{x,y}}{\partial y} + \delta t \frac{\partial P(t)_{x,y}}{\partial t} \qquad \textbf{4. 9}$$

En remplaçant l'équation (4.8) dans l'équation (4.9), nous avons :

$$\frac{H(t)_{x,y}}{\delta t} - \frac{\partial P(t)_{x,y}}{\partial t} = \frac{\delta x}{\delta t}\frac{\partial P(t)_{x,y}}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial P(t)_{x,y}}{\partial y} \qquad \textbf{4. 10}$$

Autrement écrite :

$$w_{x,y} - \nabla t_{x,y} = u_{x,y}\nabla x_{x,y} + v_{x,y}\nabla y_{x,y} \qquad \textbf{4. 11}$$

La différence d'intensité du pixel sur les colonnes, $\nabla x_{x,y}$, est la suivante:

$$\nabla x_{x,y} = \frac{\partial P(t)_{x,y}}{\partial x} = (P^{im}_{x+1,y} - P^{im}_{x,y} + P^{fond}_{x+1,y} - P^{fond}_{x,y} + P^{im}_{x+1,y+1} - P^{im}_{x,y+1} + P^{fond}_{x+1,y+1} - P^{fond}_{x,y+1})/4 \qquad \textbf{4. 12}$$

La différence d'intensité du pixel sur les lignes $\nabla y_{x,y}$ est la suivante:

$$\nabla y_{x,y} = \frac{\partial P(t)_{x,y}}{\partial y} = (P^{im}_{x,y+1} - P^{im}_{x,y} + P^{fond}_{x,y+1} - P^{fond}_{x,y} + P^{im}_{x+1,y+1} - P^{im}_{x+1,y} + P^{fond}_{x+1,y+1} - P^{fond}_{x+1,y})/4 \qquad \textbf{4. 13}$$

Les deux derniers gradients ($\nabla x_{x,y}, \nabla y_{x,y}$) peuvent être directement utilisés pour extraire les segments verticaux et horizontaux respectivement dans une image numérique. Le taux du changement d'intensité $\nabla t_{x,y}$ provoqué par le mouvement d'un objet mobile peut être écrit comme suit :

$$\nabla t_{x,y} = \frac{\partial P(t)_{x,y}}{\partial t} = (P^{fond}_{x,y} - P^{im}_{x,y} + P^{fond}_{x,y+1} - P^{im}_{x,y+1} + P^{fond}_{x+1,y} - P^{im}_{x+1,y} + P^{fond}_{x+1,y+1} - P^{im}_{x+1,y+1})/4 \qquad \textbf{4. 14}$$

Alors que le terme $w_{x,y} = \frac{H(t)_{x,y}}{\delta t}$ représente le taux du changement d'intensité du pixel à cause des raisons physiques extérieures, où :

$$H(t)_{x,y} = \frac{(P^{im}_{x,y} - P^{fond}_{x,y})}{(moy^{im} - moy^{fond})} \qquad \textbf{4. 15}$$

Donc plus la différence $(moy^{im} - moy^{fond})$ entre les moyennes de deux images est grande, moins les effets indésirables sont remarquables.

Il nous reste à calculer $u_{x,y}$ et $v_{x,y}$ (les composantes horizontale et verticale de vitesse du pixel) qui peuvent minimiser une erreur $e_{x,y}$ pour tous les pixels dans une image donnée :

$$e_{x,y} = \sum_x \sum_y (u_{x,y} \nabla x_{x,y} + v_{x,y} \nabla y_{x,y} + \nabla t_{x,y} - w_{x,y})^2 \qquad \textbf{4. 16}$$

D'autre part, si l'hypothèse qui considère que les pixels voisins qui sont en mouvement ont la même vitesse, est prise en compte, c'est-à-dire : l'hypothèse de lissage qui suppose que l'intensité lumineuse de ce voisinage varie doucement avec le temps sans aucune interruption surprise est respectée, une autre équation qui minimise le changement de flux optique pour maximiser la dernière contrainte du lissage peut être écrite :

$$e^l_{x,y} = \sum_x \sum_y \frac{1}{4}((u_{x+1,y} - u_{x,y})^2 + (u_{x,y+1} - u_{x,y})^2 + (v_{x+1,y} - v_{x,y})^2 + (v_{x,y+1} - v_{x,y})^2) \qquad \textbf{4. 17}$$

Donc, l'erreur totale à minimiser sera :

$$e^t_{x,y} = \sum_x \sum_y (\lambda\, e_{x,y} + e^l_{x,y}) \qquad \textbf{4. 18}$$

λ Est un paramètre de régularisation.

Les différentiations d'erreur totale $e^t_{x,y}$ par rapport à $u_{x,y}$ et à $v_{x,y}$ sont respectivement :

$$\frac{\partial e^t_{x,y}}{\partial u_{x,y}} = (2\lambda(u_{x,y} \nabla x_{x,y} + v_{x,y} \nabla y_{x,y} + \nabla t_{x,y} - w_{x,y}) \nabla x_{x,y} + 2(u_{x,y} - \bar{u}_{x,y}) = 0 \qquad \textbf{4. 19}$$

$$\frac{\partial e^t_{x,y}}{\partial v_{x,y}} = (2\lambda(u_{x,y} \nabla x_{x,y} + v_{x,y} \nabla y_{x,y} + \nabla t_{x,y} - w_{x,y}) \nabla y_{x,y} + 2(v_{x,y} - \bar{v}_{x,y}) = 0 \qquad \textbf{4. 20}$$

$\bar{u}_{x,y}$ et $\bar{v}_{x,y}$ sont deux moyennes calculables dans les équations (4.25) et (4.26).

D'où nous avons deux équations à deux inconnues $u_{x,y}$ et $v_{x,y}$:

$$(1 + \lambda(\nabla x_{x,y})^2) u_{x,y} + \lambda \nabla x_{x,y} \nabla y_{x,y} v_{x,y} = \bar{u}_{x,y} - \lambda \nabla x_{x,y} \nabla t_{x,y} + \lambda \nabla x_{x,y} w_{x,y} \qquad \textbf{4. 21}$$

$$(1 + \lambda(\nabla y_{x,y})^2) v_{x,y} + \lambda \nabla x_{x,y} \nabla y_{x,y} u_{x,y} = \bar{v}_{x,y} - \lambda \nabla y_{x,y} \nabla t_{x,y} + \lambda \nabla y_{x,y} w_{x,y} \qquad \textbf{4. 22}$$

Donc, nous avons finalement ce que nous cherchons :

$$u_{x,y} = \bar{u}_{x,y} - \lambda\left(\frac{\nabla x_{x,y}\bar{u}_{x,y} + \nabla y_{x,y}\bar{v}_{x,y} + \nabla t_{x,y}}{1+\lambda((\nabla x_{x,y})^2 + (\nabla y_{x,y})^2)}\right)\nabla x_{x,y} + \frac{\lambda\ w_{x,y}\ \nabla x_{x,y}}{1+\lambda((\nabla x_{x,y})^2 + (\nabla y_{x,y})^2)} \qquad \textbf{4. 23}$$

$$u_{x,y} = \bar{v}_{x,y} - \lambda\left(\frac{\nabla x_{x,y}\bar{u}_{x,y} + \nabla y_{x,y}\bar{v}_{x,y} + \nabla t_{x,y}}{1+\lambda((\nabla x_{x,y})^2 + (\nabla y_{x,y})^2)}\right)\nabla y_{x,y} + \frac{\lambda\ w_{x,y}\ \nabla y_{x,y}}{1+\lambda((\nabla x_{x,y})^2 + (\nabla y_{x,y})^2)} \qquad \textbf{4. 24}$$

$$\bar{u}_{x,y} = (u_{x-1,y} + u_{x-1,y-1} + u_{x-1,y+1} + u_{x+1,y} + u_{x+1,y-1} + u_{x+1,y+1} + u_{x,y-1} + u_{x,y+1})/8 \qquad \textbf{4. 25}$$

$$\bar{v}_{x,y} = (v_{x-1,y} + v_{x-1,y-1} + v_{x-1,y+1} + v_{x+1,y} + v_{x+1,y-1} + v_{x+1,y+1} + v_{x,y-1} + v_{x,y+1})/8 \qquad \textbf{4. 26}$$

A.

B.

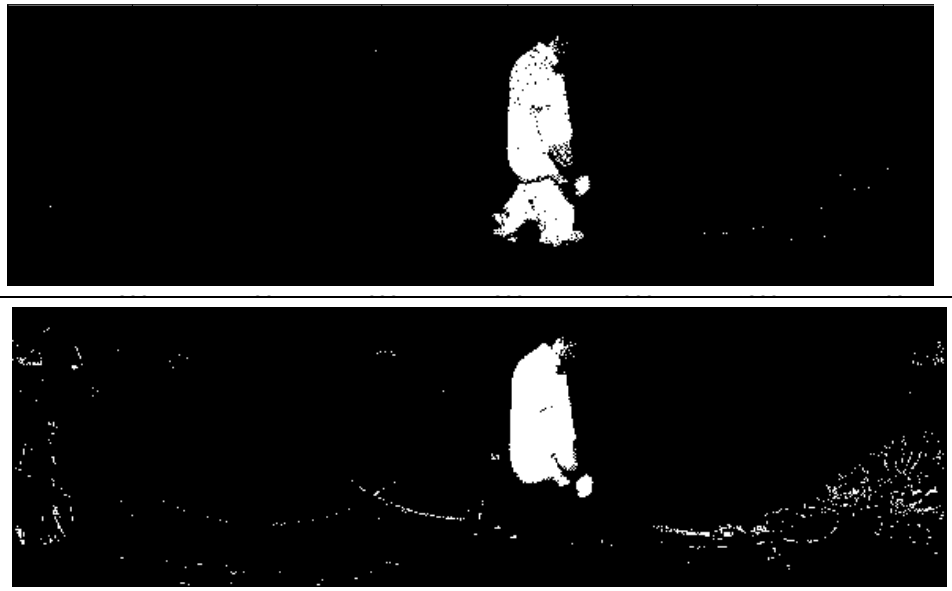


Figure (4. 13), A. une personne détectée dans une image binaire résultant de l'application de l'approche discutée sur l'image (4.12.B) et le fond extrait (4.12.D), B. la même personne détectée en faisant une soustraction directe entre les deux images citées.

La composante présentée dans les équations (4.23) et (4.24) :

$$\frac{\lambda\ w_{x,y}\ (\nabla x_{x,y}\ ou\ \nabla y_{x,y})}{1+\lambda((\nabla x_{x,y})^2 + (\nabla y_{x,y})^2)}$$

Représente le nouveau terme ajouté pour compenser les effets indésirables produits par les phénomènes physiques externes déjà expliqués.

L'amplitude de vitesse du pixel qui est en mouvement sera :

$$V_{x,y} = \sqrt{(u_{x,y})^2 + (v_{x,y})^2} \qquad \textbf{4. 27}$$

A la fin du calcul, une image de vitesses des pixels sera construite. Chaque pixel a une amplitude supérieure à un seuil égal à 10% de $V_{x,y}^{\max}$ ($V_{x,y}^{\max}$: l'amplitude maximale du pixel dans l'image) prend la valeur (255 blanc), sinon le pixel est noir.

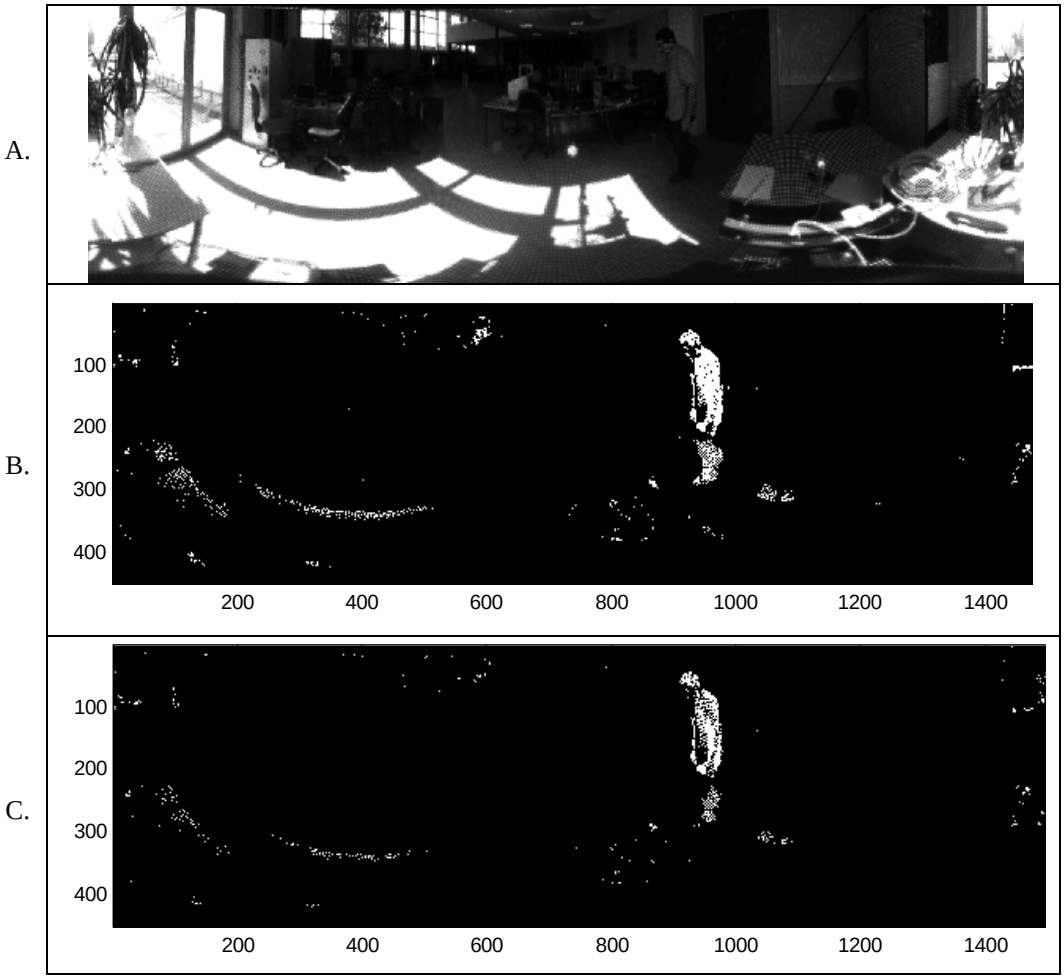


Figure (4. 14), Une personne détectée avec un changement de lumière entre le fond extrait dans la figure (4.12.D) et l'image courante (4.14.A). B. Une image qui résulte de l'application d'une approche qui ne respecte pas notre hypothèse. C. une image qui résulte de l'application de l'approche qui respecte bien notre hypothèse

La figure (4.13), montre la différence de qualité entre une image binaire résultant de l'application de l'approche du flot optique expliquée dans ce chapitre et celle qui résulte d'une soustraction directe entre l'image courante et le fond de scène. Toutes les deux sont construites avec le même niveau de seuillage. Nous pouvons constater que l'image (4.13.A) est moins bruitée que l'image (4.13.B), en plus les pixels qui appartiennent à la cible détectée sont bien distribués sur

toutes les parties de cible, alors que dans l'image (4.13.B) il y a un manque dans la partie inférieure de cible.

D'autre part, l'image (4.14.C) qui est construite en appliquant l'approche qui respecte notre hypothèse ($H(t)_{x,y} \neq 0$ pour compenser les changements physiques extérieurs) est moins bruitée que l'image (4.14.B) qui ne la respecte pas. Les "fantômes" montrés dans les deux images (4.14. B et C) sont formés à cause du mouvement de l'ombre des cadres des fenêtres, ce mouvement est lié au changement de direction des rayons du soleil entre l'instant d'extraction du fond de scène et l'instant d'acquisition de l'image courante. Pour réduire ce qui reste comme bruit dans l'image (4.14.C), il faut mettre à jour le fond de scène de façon répétitive qui semble une solution utile sauf s'il y a un changement de lumière drastique et brutal. Par contre, faire la mise à jour de façon permanente permet d'éliminer tout le bruit qui peut arriver (ce qui peut donner de très bons résultats), mais sera coûteux au niveau du temps de calcul car il faut dans ce cas ajouter une fonction simultanément aux autres fonctions. Pour cela, il faut choisir une fréquence qui garantit une meilleure performance en temps réel, sur une base expérimentale.

Dans nos expériences, la mise à jour du fond de scène est faite pour chaque 12 images acquises dans un environnement où il y a une ou deux cibles mobiles se déplaçant avec une vitesse moyenne égale à 0.4 m/s. En outre, le fond de scène sera renouvelé à chaque fois que la zone de perception du robot devient complètement vide. Dans les deux cas, les algorithmes qui sont utilisés pour faire la détection et le suivi de cibles mobiles seront initialisés. Lorsque le robot s'arrête après n'importe quel petit mouvement (linéaire ou angulaire), une mise à jour pour le fond de scène sera obligatoirement effectuée.

4.6. Déterminer la Direction de la Cible Détectée

Afin de déterminer la direction de cible détectée, il faut calculer un histogramme sur les colonnes de l'image binaire, figure (4.15.A). Pour une image de taille (*NbRows, NbColumns*), la probabilité qu'une colonne appartienne à une cible détectée se calcule comme suit :

$$
\begin{array}{l}
\{image\} \\
for(i = 0; i < NbColumns; i++) \\
\quad for(j = 0;\ j < NbRows;\ j++) \\
\quad\quad if\ (image(j,i) == 255) \\
\quad\quad ColumnCounter++; \\
\quad endfor \\
\quad Column\Pr o = \eta\ \dfrac{ColumnCounter}{NbRows} \\
endfor \\
return(Columns\Pr o)
\end{array}
$$

Tableau (4.4), Un algorithme qui calcule les probabilités des colonnes dans une image binaire

L'algorithme illustré dans le tableau (4 .4) accumule sur chaque colonne le nombre de pixels qui ont la luminosité maximale (255), puis il attribue à chaque colonne son degré d'intérêt représenté par une probabilité proportionnelle aux nombre de pixels accumulés.

Dans l'histogramme illustré dans la figure (4.15.B), notre approche va chercher la zone la plus intéressante qui se compose des colonnes côte à côte, car elles ont des probabilités élevées. Le nombre de colonnes qui indique le commencement de cette zone ($ColumnStart$) et celui qui indique sa fin ($ColumnEnd$) sont importantes pour déterminer la moyenne de cette zone (μ) qui peut être calculée dans l'équation suivante:

$$\mu = ColumnStart + \frac{ColumnEnd - ColumnStart}{2} \qquad \textbf{4. 28}$$

Alors que la direction de cible détectée est calculable comme suit :

$$ThetaDetect = 2\pi\left(\frac{NbColumns - \mu}{NbColumns}\right) \qquad \textbf{4. 29}$$

Par exemple : dans la figure (4.15), Pour une image de taille (1500 x 450) nous avons une zone d'importance isolée : ($ColumnStart$ = 778, $ColumnEnd$ = 928). La moyenne sera : (μ = 853). Alors que la direction de cible détectée est : (thêta = 2.72 rad).

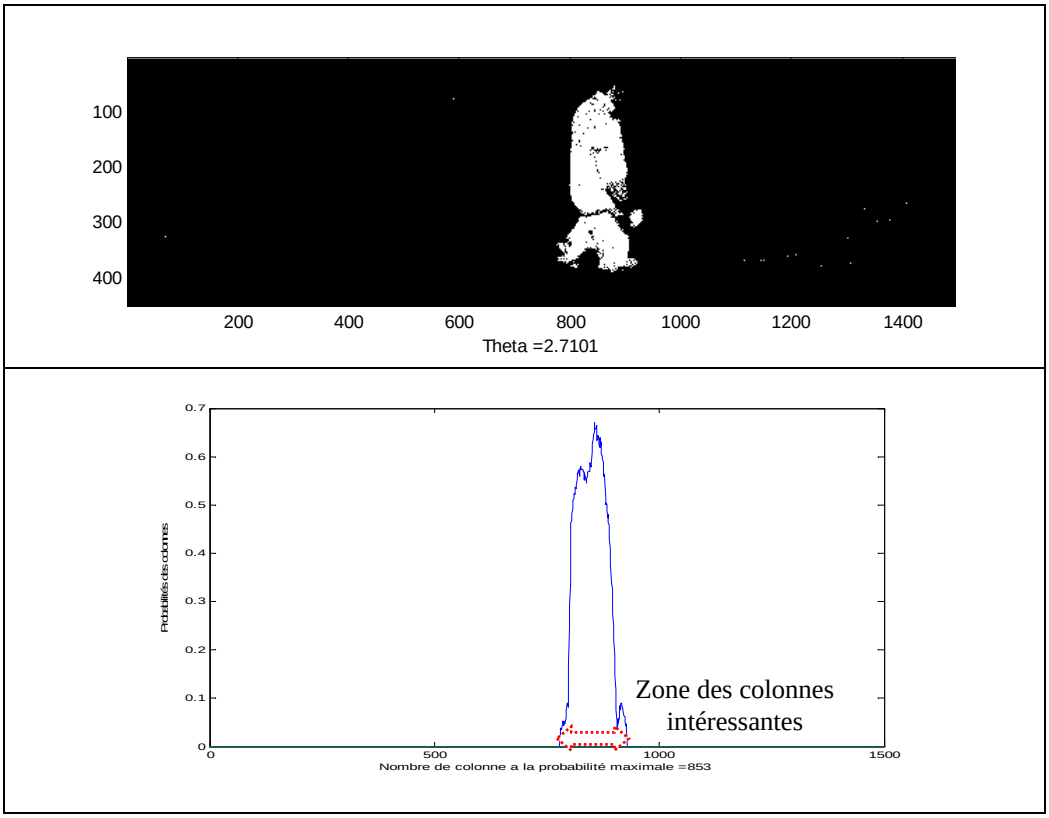


Figure (4. 15), Une zone des colonnes intéressantes (sont côte à côte et elles ont des probabilités supérieures)

4.7. Déterminer les Directions de Plusieurs Cibles

Pour détecter plusieurs cibles mobiles dans une scène, la même technique du flot optique sera appliquée. La méthode déjà utilisée pour déterminer la direction d'une seule cible va être employée aussi. La figure (4.16) illustre deux cibles détectées dans les deux directions (2.36 et 3.45 radian) respectivement.

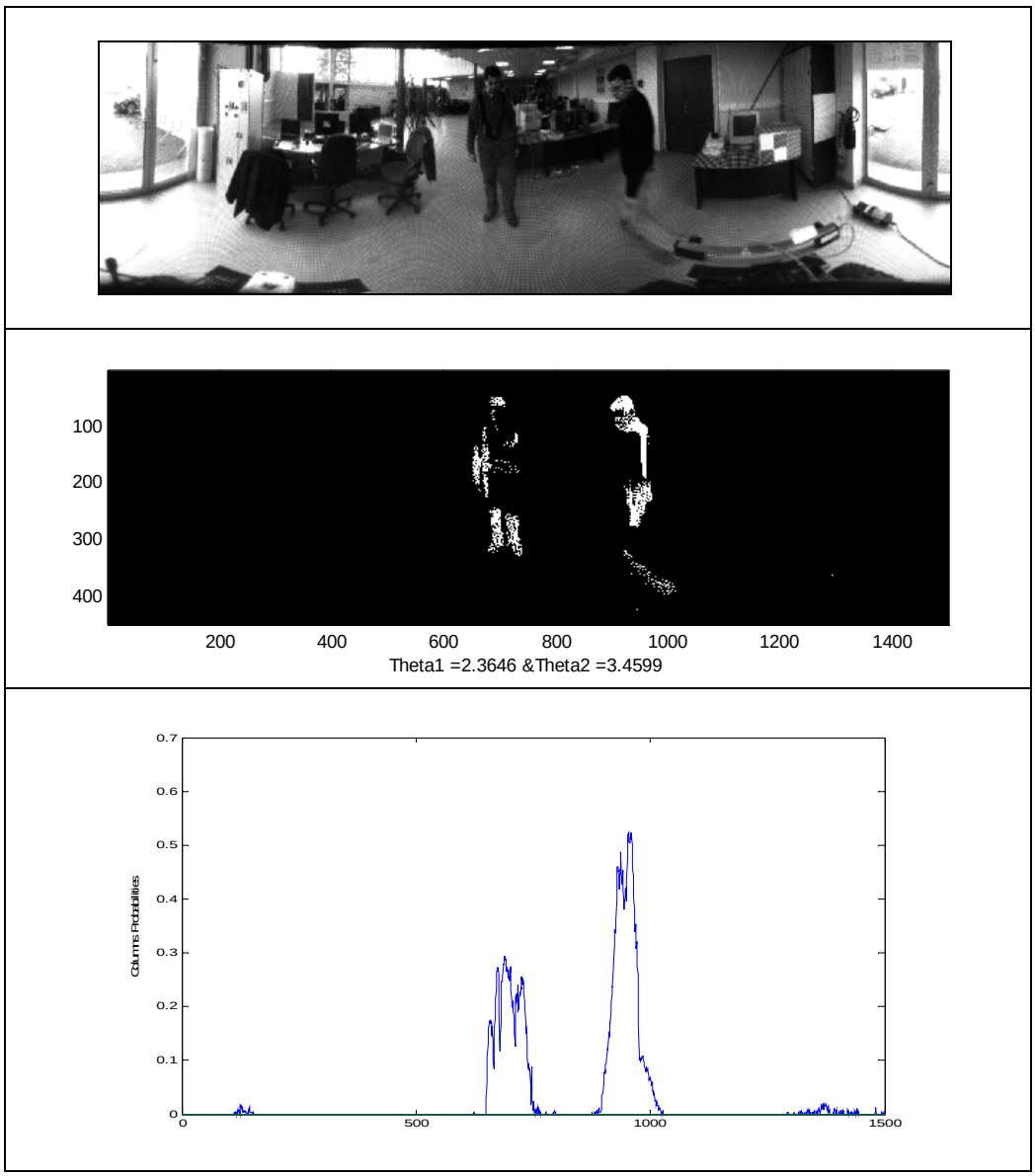


Figure (4. 16), A. deux cibles se déplaçant dans la zone de perception du robot, B. une image binaire montrant les deux cibles détectées, C. histogramme montrant deux zones de colonnes intéressantes (ont des probabilités supérieures).

4.8. Détecter les Cibles mobiles à Partir d'Images Panoramiques non Déployées (Images Brutes)

La méthode utilisée dans ce cas, utilise la matrice de convolution expliquée dans le paragraphe (4.4.1). La matrice employée est de taille (51x51), ses cœfficients de pondération sont égaux et égalent à un. En appliquant cette méthode, nous pouvons construire une nouvelle image dans laquelle chaque pixel a accumulé une nouvelle valeur à partir des valeurs de ses voisins. D'où, un pixel qui appartient à un groupe de pixels représentant une cible va accumuler une grande intensité, et les intensités maximales (les sommets) seront attribuées aux pixels qui sont au centre du groupe, figure (4.19, B).

$$P_{x,y}^{acc} = \sum_{i=-N}^{N} \sum_{j=-N}^{N} P_{x+i,y+j} \qquad \textbf{4. 30}$$

$P_{x,y}^{acc}$: L'intensité de pixel dans l'image résultant de l'application de la matrice de convolution. Et :

$$N = (TemplateDim - 1)/2$$

$TemplateDim$: est la dimension de matrice de convolution soit sur les colonnes ou sur les lignes (ici c'est 51).

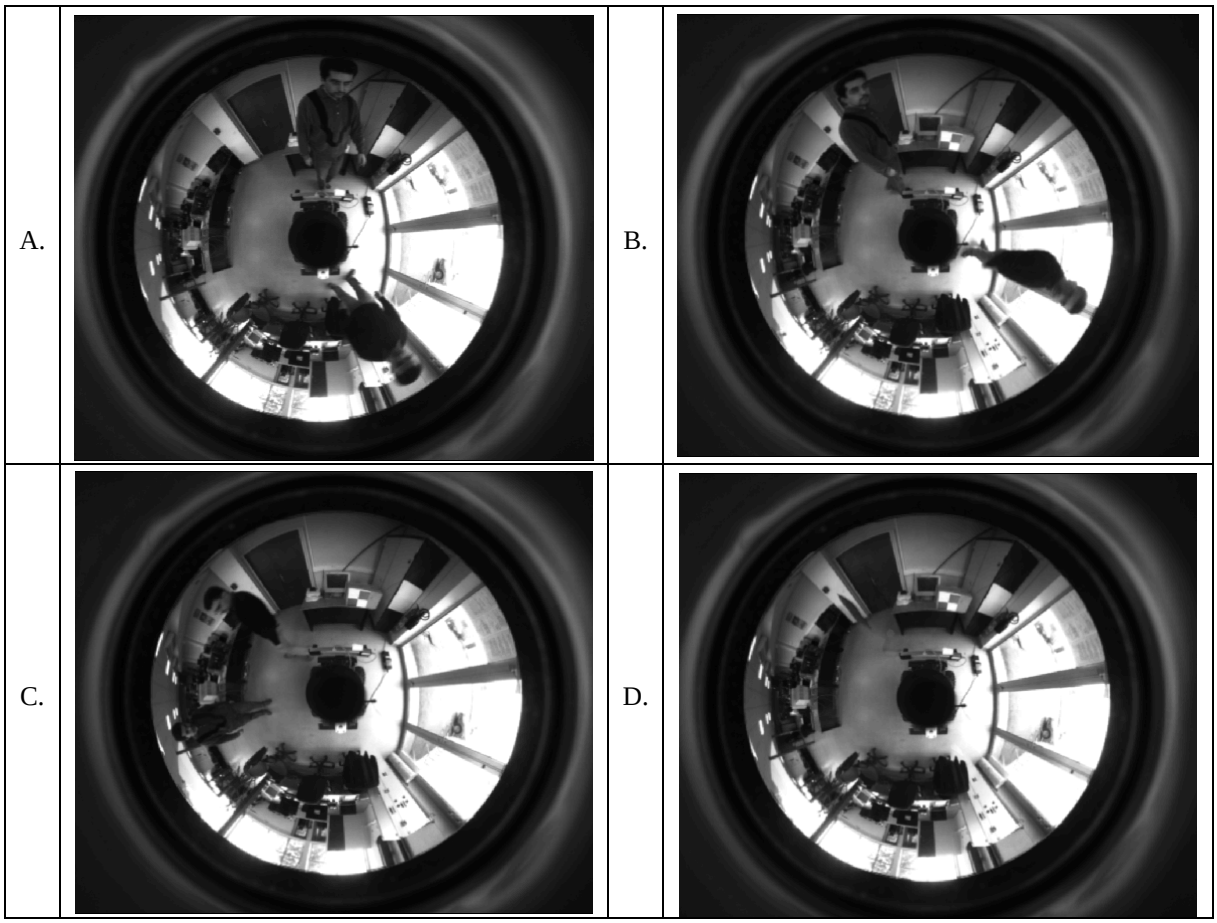


Figure (4. 17) A partir de trois images panoramiques brutes (A, B, C) prises à des moments différents, le fond de scène (D) est extrait.

Après une étape de normalisation les pixels entourés par des voisins qui sont souvent noirs vont être éliminés, alors que les pixels qui portent des intensités accumulées maximales, vont prendre de nouveau la valeur (255), figure (4.19, A). Pour normaliser :

$$P_{x,y}^{new} = 255 * \frac{P_{x,y}^{acc}}{MAX(P_{x,y}^{acc})} \qquad \textbf{4. 31}$$

P^{new} : La nouvelle valeur du pixel après la normalisation. $MAX(P_{x,y}^{acc})$: Représente la valeur maximale accumulée dans cette image.

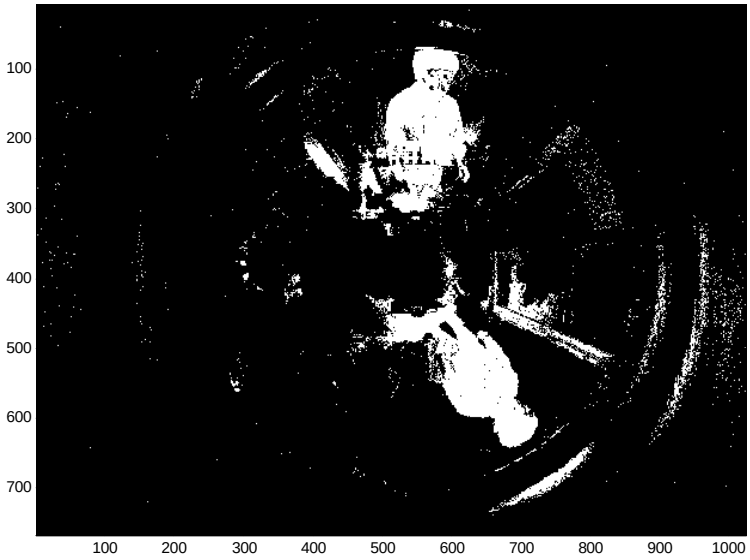


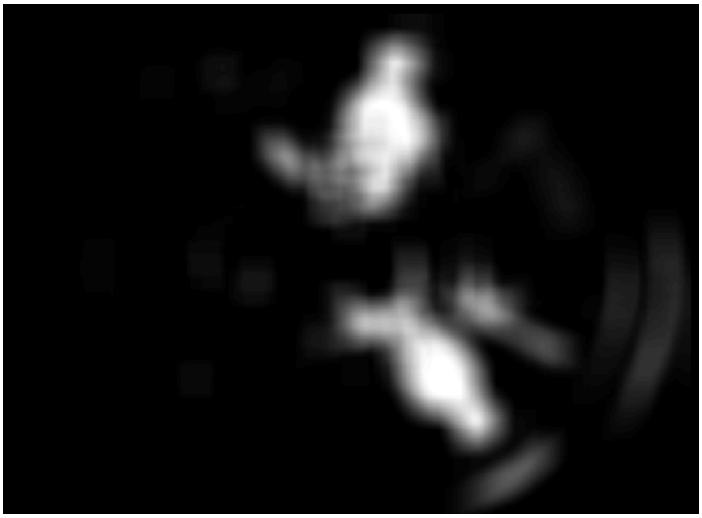
Figure (4. 18), une image binaire résultant de l'application de flot optique sur l'image (4.17.A) et le fond extrait (4.17.D).

L'histogramme illustré dans la figure (4.19, D) montre comment la stratégie que nous avons suivie a réussi à créer une nouvelle image avec des niveaux de luminosité groupés et bien séparés, où les pixels de sommets sont minoritaires, ce qui permet de les isoler par une simple étape de seuillage avec un seuil élevé (e.g. $s \approx 200$), pour obtenir une image finale comme celle illustrée dans la figure (4.19, C) où nous avons finalement deux zones qui représentent deux cibles détectées dans la scène. Cette dernière image est bien nettoyée du bruit et des fantômes. Les directions des cibles détectées se calculent facilement comme suit :

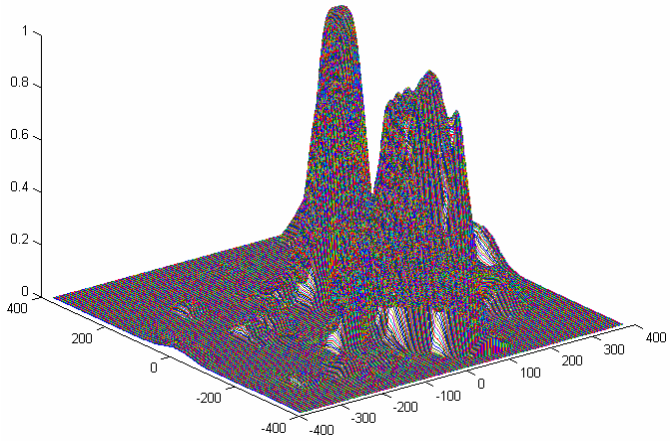
$$ThetaDetect = arctg\left(\frac{y^t}{x^t}\right) \qquad \textbf{4. 32}$$

x^t et y^t sont les coordonnées centrales de cible détectée, elles sont représentées par les nombres de lignes et de colonnes occupées par la cible dans l'image binaire. Par exemple dans la figure (4.19, C) il y a deux cibles détectées dans deux directions : (1. 5 rads et 5. 27 rads).

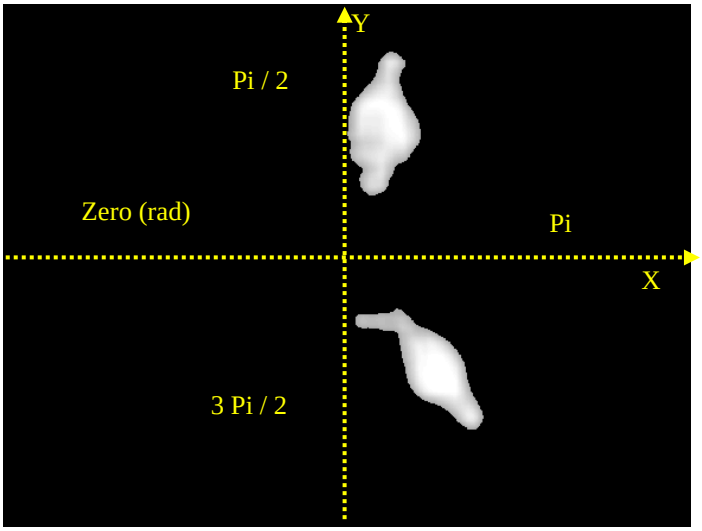
A.



B.


1
0.8
0.6
0.4
0.2
0
400
200
0
-200
-400
-400
-300
-200
-100
0
100
200
300
400

C.


Y
Pi / 2
Zero (rad)
Pi
X
3 Pi / 2

D.

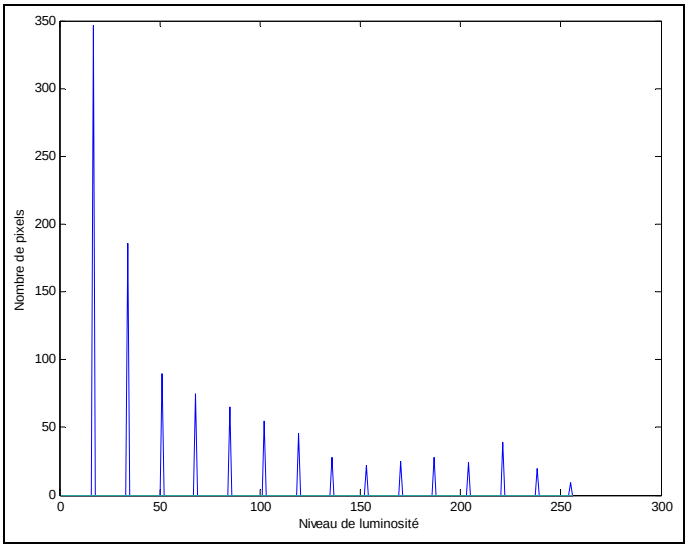


Figure (4. 19), A. Image après l'application d'une étape de normalisation, B. Histogramme 3D de l'image (4.18) avant normalisation, C. image finale avec deux cibles détectées, D. Histogramme 2D de l'image (4.19. A).

4.9. Comparaison Entre les Deux Techniques Suivies Pour Déterminer les Directions des Cibles Détectées

La deuxième approche discutée dans le (8ème) paragraphe est plus fiable que la première discutée dans le (7ème). En revanche, même si elle n'a pas besoin d'une image déployée, la deuxième approche est plus coûteuse au niveau du temps du calcul à cause de la grande dimension de la matrice de convolution utilisée. La matrice est choisie de taille (51x51) pour qu'elle soit décisive pour la distinction entre un pixel (bruit) et un autre qui appartienne vraiment à une cible. Autrement dit, il joue le rôle d'un « groupeur » et d'un nettoyeur à la fois.

Quand les images panoramiques sont utilisées comme des sources importantes d'information pour être fusionnées avec d'autres capteurs (comme le laser), la première approche semble plus adéquate car elle est plus rapide, et pour de petites résolutions d'image (250x75) elle peut fonctionner en temps réel et sa perte de fiabilité peut être compensée par les données fournies par l'autre capteur. Mais, pour un système de vision employé seul pour faire la détection à un degré de fiabilité élevé, la deuxième approche devient meilleure.

4.10. Conclusion

Dans ce chapitre, nous avons introduit brièvement la camera catadioptrique : la combinaison du système catadioptrique (miroir parabolique - objectif télécentrique), son modèle de projection, et le redressement cylindrique d'une image panoramique. Quelques concepts essentielles avec quelques techniques de base de traitement d'images sont aussi discutées (l'histogramme, matrice de convolution, les opérateurs : médian et Gaussien).

Ensuite, nous avons détaillé avec nos expériences la détection des cibles mobiles en utilisant les images panoramiques. Le fond de scène (Background) est extrait à partir de trois images successives (prises avec un délai du temps constant) en employant le (Temporal Median Operator). Le calcul de l'image binaire est basé sur une approche modifiée du flot optique où un nouveau terme qui compense les changements d'illumination et leurs influences indésirables est trouvé et pris en compte. Deux stratégies pour déterminer les directions des cibles détectées à partir d'images déployées (sur une forme cylindrique) et à partir d'images non déployées, sont respectivement présentées et comparées.

Le chapitre suivant va discuter la fusion de données sensorielles (télémètre laser et une caméra catadioptrique) afin d'améliorer la fiabilité de détection du système expliqué dans le chapitre précédent. Mais dans ce cas, la synchronisation des données fournies par les deux capteurs devient un sujet crucial, sur lequel nous allons nous concentrer.

CHAPITRE V

Fusion de Capteurs
(Et Alignement des Données):

5.1. Introduction

5.2. Fusion de Capteurs

5.3. Considérations Techniques

5.4. Résultats Expérimentaux

5.5. Architecture Algorithmique

5.6. Conclusion

5.1. Introduction

Chez l'homme conscient une expérience simple de palper un objet sans le regarder peut être suffisante pour rendre compte de sa température, de sa forme. Mais afin d'avoir plus d'informations sur l'objet (aspect, couleur, etc.), il faut fusionner la vision avec le toucher.

En robotique, la fusion de capteurs est notamment proposée pour compenser les insuffisances des uns par rapport aux autres afin d'obtenir le maximum d'informations pour améliorer la perception du robot et la compréhension de son environnement. Nous avons développé le système illustré dans le chapitre III qui fait le suivi multi-cibles et la cartographie simultanée, mais ce système a des limitations pour la détection de plusieurs personnes se déplaçant aléatoirement dans le champ de perception du robot. Fusionner le télémètre laser avec un capteur visuel est alors une possibilité pour augmenter l'efficacité du système. Pour cela, nous avons choisi les images panoramiques comme source riche d'information pour compléter le laser. Mais fusionner des données, en particulier en temps réel, rend cruciale la synchronisation des capteurs qui les fournissent, car si un événement a lieu et est détecté à l'instant (t) par un capteur, le même événement peut être détecté à un instant ($t + \delta t$) par l'autre capteur, ce qui peut être interprété comme un autre événement indépendant. Elaborer un système capable de distinguer et coupler les signaux reçus par les capteurs et provoqués par le même événement est donc indispensable.

Dans ce chapitre nous présentons la technique suivie pour résoudre le problème d'alignement de données fournies par les deux capteurs utilisés (laser et caméra catadioptrique). Les considérations techniques que nous avons prises en compte pour mener nos expériences sont aussi illustrées. Les résultats expérimentaux sont détaillés et illustrés par des tableaux et des figures. Enfin, une architecture algorithmique organisant les relations entre les différentes fonctions discutées dans tous les chapitres précédents ainsi qu'avec d'autres modules préexistants est présentée. Une conclusion générale résumant le travail effectué clôt le chapitre.

5.2. Fusion de Capteurs

Dans notre application, la détection et le suivi de cible mobile en utilisant le laser, expliqués au chapitre III, sont plus lourds en temps de traitement que la détection visuelle proposée au chapitre IV qui peut fonctionner en temps réel pour la résolution d'image considérée. Pour aligner les données reçues par les deux capteurs nous proposons une technique probabiliste. En considérant que le détecteur visuel peut seulement fournir la direction de la cible détectée, alors que le laser peut fournir la position de la cible en distance et en direction à la fois, la fusion sera faite au niveau de l'information commune représentée par la direction de la cible détectée. Le principe de la technique que nous proposons (figure 5.1) est le suivant. Le détecteur visuel réserve pour chaque cible détectée un registre FIFO, (First Input First Output) pour enregistrer ses dix dernières directions détectées; chacune d'entre elles sera appelée "cas". Chaque direction thêta fournie par le laser va déclencher une étape de pondération sur toutes les valeurs qui sont déjà stockées dans tous les registres. Ensuite, nous recherchons le "cas" avec le poids le plus élevé pour coupler les deux signaux décrivant le même événement qui les a produits, et attribuer cet événement à la cible appropriée s'il y en a plusieurs.

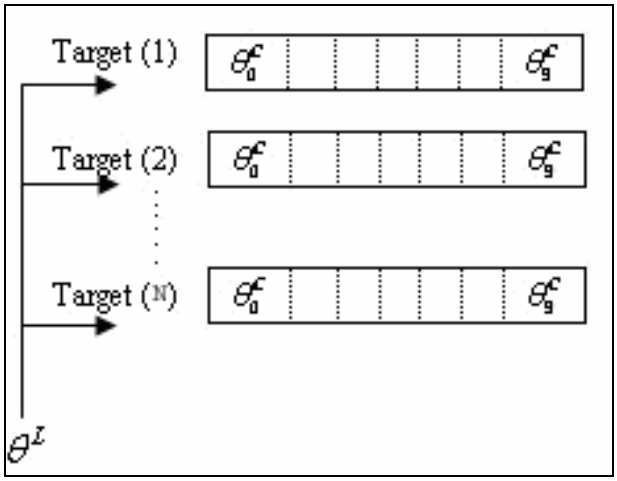


Figure (5.1), N registre pour N cibles, les dix dernières positions angulaires (déterminées par le détecteur visuel) de chaque cible sont enregistrées. Une étape de pondération va être effectuée sur toutes les valeurs de N registres chaque fois que le système reçoit une direction détectée par le laser.

Le poids de chaque "cas" peut être évalué comme suit :

$$w_n^N = \eta * \exp\left(-\left(\frac{\theta_n^C - \theta^L}{\sigma}\right)^2\right) \qquad \textbf{5. 1}$$

Avec : N le nombre de cibles, $n = 0,\ldots,9$ les dix derniers cas dans chaque registre, θ_n^C et θ^L : les directions de cible fournies respectivement par la caméra et le laser. σ : l'écart type, η : est une constante de normalisation.

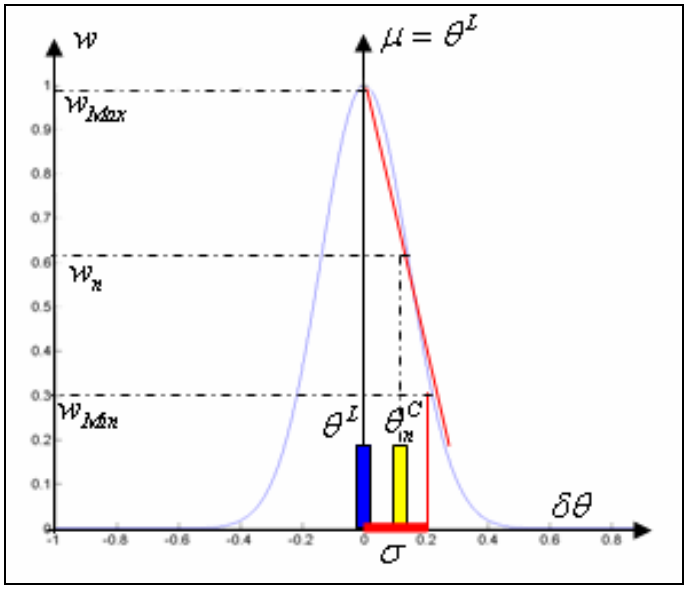


Figure (5. 2), Distribution Gaussienne, (modèle de pondération).

Le modèle de pondération proposé suit une distribution gaussienne, dans laquelle la direction de la cible détectée par le laser est toujours supposée représenter la moyenne. Ainsi, le poids de chaque « cas » changera entre sa valeur minimale jusqu'à sa valeur maximale selon la différence entre θ_n^C et θ^L, figure (5.2). Avec une approximation qui peut rester acceptable, cette distribution gaussienne peut être linéarisée pour déterminer la direction de la cible détectée et faire la fusion comme suit :

$$\theta_{Fusion}^N = \theta^L + \delta\theta_n^N \qquad \textbf{5. 2}$$

Où

$$\delta\theta_n^N = \pm\frac{(1-w_n^N)\sigma}{(w_{Max}-w_{Min})}$$

(Pour σ = 0.2 radian et η = 0.99) la Gaussienne est celle qui est illustrée figure 5.2.

La direction calculée dans l'équation (5.2) peut être aussi considérée comme une simple moyenne de deux directions :

$$\theta_{Fusion}^N = (\theta^L + \theta_n^N)/2 \qquad \textbf{5. 3}$$

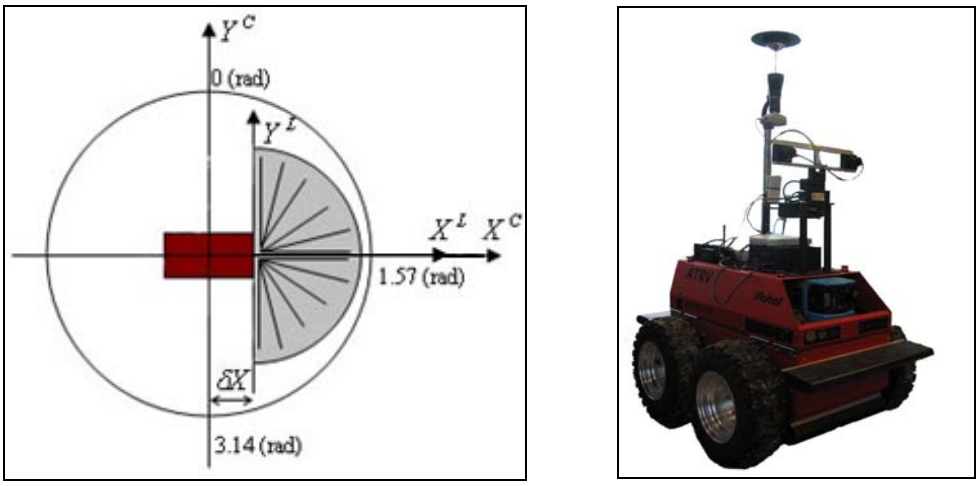
Bien que l'équation (5.2) semble plus compliquée que la dernière équation (5.3), elle propose à l'utilisateur du système une méthode utile pour analyser les cas de divergence entre les deux directions détectées (θ^L et θ_n^N) même si elles sont fusionnées. C'est ce qui se passe par exemple si l'état choisi après l'étape de pondération exprime une probabilité légère même si elle est la plus importante parmi les autres. Dans ce sens, la valeur de $\delta\theta_n^N$ devient un moyen de diagnostiquer la divergence. Plus elle est proche de zéro, plus il y a de convergence entre les deux données angulaires, et plus la probabilité de l'état choisi aura un poids significatif, et vice-versa. Un cas de divergence peut arriver si l'un des deux capteurs ne fonctionne pas correctement. C'est par exemple le cas s'il y a un changement drastique d'illumination de la scène rendant la détection visuelle de cible mobile impossible à un instant donné.

5.3. Considérations Techniques

Le robot "DALA" illustré dans la figure (5.3. B.) équipé d'un télémètre laser et d'une caméra catadioptrique est utilisé dans nos expériences. Le secteur d'intersection entre les deux zones de perception pour les deux capteurs utilisés est illustré dans la figure (5.3. A.). Le laser couvre seulement 180° en semi cercle devant le robot (son rayon utile est choisi ici pour être de trois mètres), tandis que la caméra catadioptrique couvre 360° autour du robot. En plus, s'il existe une différence (δX = 34 cm) entre la référence du laser (X^L, Y^L) et celle de la caméra (X^C, Y^C). Pour interpréter la direction d'une cible donnée détectée dans la zone d'intersection il faut naturellement faire une calibration entre les 2 capteurs de manière à avoir une transformation de repère correcte entre eux. Pour ce faire, le calcul de la position angulaire de la cible dans la référence du laser doit prendre en compte une transformation sur l'axe X^L :

$$\theta^L = arctg\left(\frac{X^L + \delta X}{Y^L}\right) \qquad \textbf{5. 4}$$

La fusion sera réalisée seulement dans la zone de 180° couverte le laser. Une partie de l'image panoramique décrivant la moitié de son champ, avec une résolution d'image de (75, 250) sera suffisante pour jouer le rôle d'un détecteur visuel. Son principe a déjà été expliqué au chapitre 4.



A B

Figure (5. 3), A. La zone d'intersection de deux capteurs est illustrée en gris. B. le robot Dala équipé par un télémètre laser et une caméra catadioptrique.

5.4. Résultats Expérimentaux

Le tableau 1 illustre dans sa première ligne l'évolution temporelle du suivi d'une cible se déplaçant dans la zone d'intersection des capteurs, en respectant la fréquence de détection visuelle. Les directions de la cible détectée sont présentées dans la deuxième ligne. Ainsi les première et deuxième lignes représentent un registre pour une cible. La troisième ligne fournit les directions détectées par le laser qui arrivent toujours avec quelque retard pour un événement déjà détecté par l'image panoramique. Certaines des lectures du laser fournissent au système deux valeurs proches qui sont en fait les deux jambes d'une personne détectée, mais elles sont attribuées à la même cible grâce à la procédure d'association de données expliqué au le chapitre III. En appliquant l'approche de fusion présentée dans ce chapitre, chaque θ^L reçu sera fusionné avec le θ_n^C adéquat. La valeur qui interprète la fusion des deux capteurs qui détectent le même événement est calculée en utilisant l'équation (5.2) et est donnée dans la dernière ligne.

La figure (5.4), illustre la trajectoire d'une cible se déplaçant à proximité du robot, les points rouges étant la trajectoire de cible détectée par le laser. Les points bleus sont les parties fixes de l'environnement construits simultanément en utilisant la technique de grille d'occupation expliquée dans le chapitre II. Les points jaunes sont les directions détectées en utilisant le capteur visuel et qui sont couplées avec ceux qui sont fournis par le laser. Enfin, les points noirs représentent les moments de fusion. Pour tracer ces deux derniers types de points (jaunes et noirs) nous avons utilisé la distance

fournie par les mesures du laser qui sont couplées avec eux. La figure (5.5), montre deux trajectoires de deux cibles détectées dans la zone de perception du robot. Plus cette zone est occupée plus le temps du calcul pour chaque état de détection (par le laser) sera grand. D'où, moins de nombre d'états détectés.

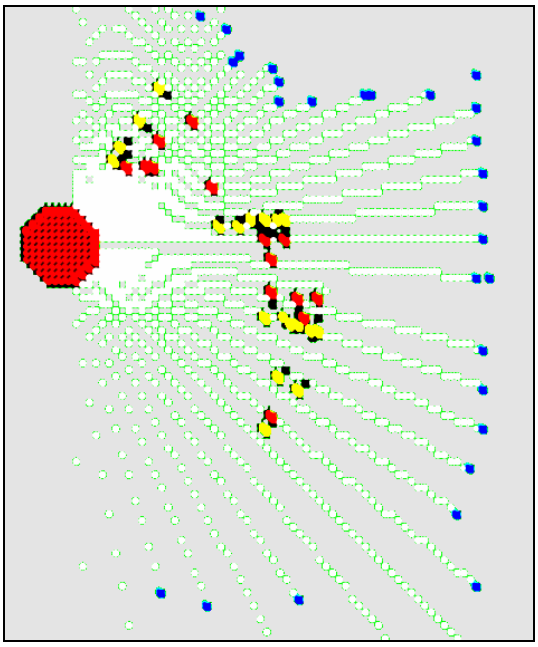


Figure (5. 4), A. Une cible se déplaçant à proximité du robot; les points rouges représentent la trajectoire détectée par le laser ; les cellules bleues sont les parties statiques de l'environnement. Les points jaunes sont les directions détectées par la caméra et qui sont couplées à ceux qui sont détectées par le laser. Alors que, les points noirs représentent les moments de fusion.

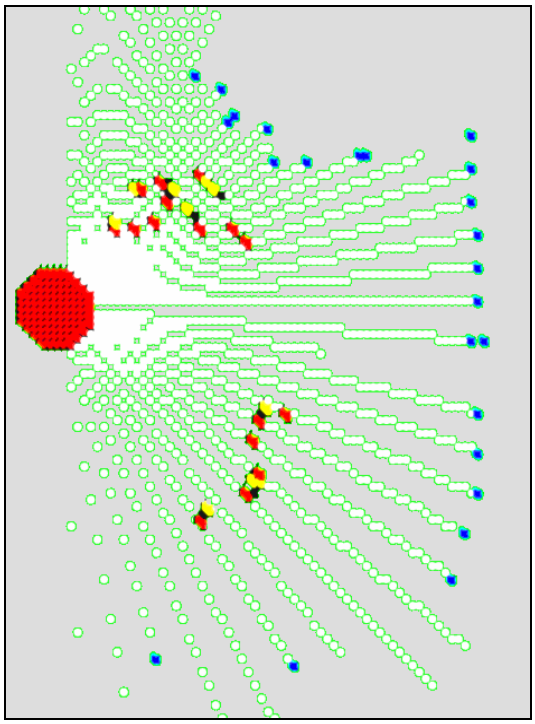


Figure (5.5), Deux cibles se déplaçant à proximité du robot ; le code de couleurs est identique à celui de la figure (5.4). Dans les figures (5.4) et (5.5), un petit nombre de points jaunes peuvent être couverts par les points rouges.

Comme nous l'avons déjà signalé, la technique de fusion que nous proposons dans ce chapitre est applicable pour un robot stationnaire. Pour un robot en mouvement continu, le suivi de cibles mobiles est possible seulement en utilisant le laser comme capteur. Dans les deux cas (robot stationnaire ou en mouvement) le laser est utilisé pour construire la carte de l'environnement de façon simultanée.

TABLEAU 1

ECHANTILLONS DE TRAJECTOIRE D'UNE CIBLE ILLUSTRANT LA NECESSITE D'ALIGNER LES DONNEES DES DEUX CAPTEURS UTILISEES

Pas de Temps	*n= 0*	*n= 1*		*n= 2*		*n= 3*		*n= 4*	
θ_n^C	2.211	1.595		1.533		1.33		1.19	
θ^L fusionnée avec			2.32, 2.35 θ_0^C		2.15 θ_0^C		2.06, 2.1 θ_0^C		1.88, 1.71 θ_1^C
θ_{Fusion}			2.267		2.169		2.155		1.626

Pas de Temps	*n= 5*		*n= 6*		*n= 7*	*n= 8*	*n= 9*		*n= 10*		
θ_n^C	0.841		0.565		0.49	0.35	0.23		0.075		
θ^L fusionnée avec		1.63 θ_1^C		1.54 θ_2^C				0.84 θ_5^C		0.73, 0.78 θ_5^C	0.55 θ_6^C
θ_{Fusion}		1.62		1.546				0.84		0.814	0.551

5.5. Architecture Algorithmique

Nous décrivons ici la structuration de notre architecture algorithmique en termes de composants et d'organisation de ses fonctions. Dans la figure (5.6), les fonctions illustrées en bleu et expliquées chacune dans son propre chapitre, représentent les briques essentielles pour réaliser la performance souhaitée. Toutes les fonctions sont écrites en langage C dans l'environnement de développement GenoM [46] (Generator of Module) du LAAS, permettant de définir et de produire des modules qui encapsulent des algorithmes. Un module GenoM est une entité logicielle standardisée qui gère les traitements et offre des services associés : démarrer ou arrêter un traitement, transmettre des paramètres, envoyer des requêtes, récupérer des résultats, exporter des structures de données (appelées posters), . . . etc.

Le contrôleur écrit comme un script Tcl, lit un poster produit par le détecteur visuel pour gérer les instants où le module caméra doit capturer une seule ou une série d'images, et à quel moment il faut arrêter (e. g. si le robot commence à se déplacer). L'affichage de la carte construite et des cibles suivies comme celles illustrées dans les figures (5.4) et (5.5) sont effectuées par un script Tk. Les

autres modules préexistants sont le module (SICK) qui s'occupe de gérer le balayage laser 2D et de fournir les informations portées par les retours de ses rayons ; le module (POM) qui s'occupe de fournir une meilleure localisation du robot en fusionnant toutes les informations disponibles pour estimer la position du robot (odométrie, localisation sur segments d'un modèle connu… etc.) et le module (CAMERA) qui s'occupe de l'acquisition des images et de les fournir dans un (Buffer) des matrices prêtes à être lues.

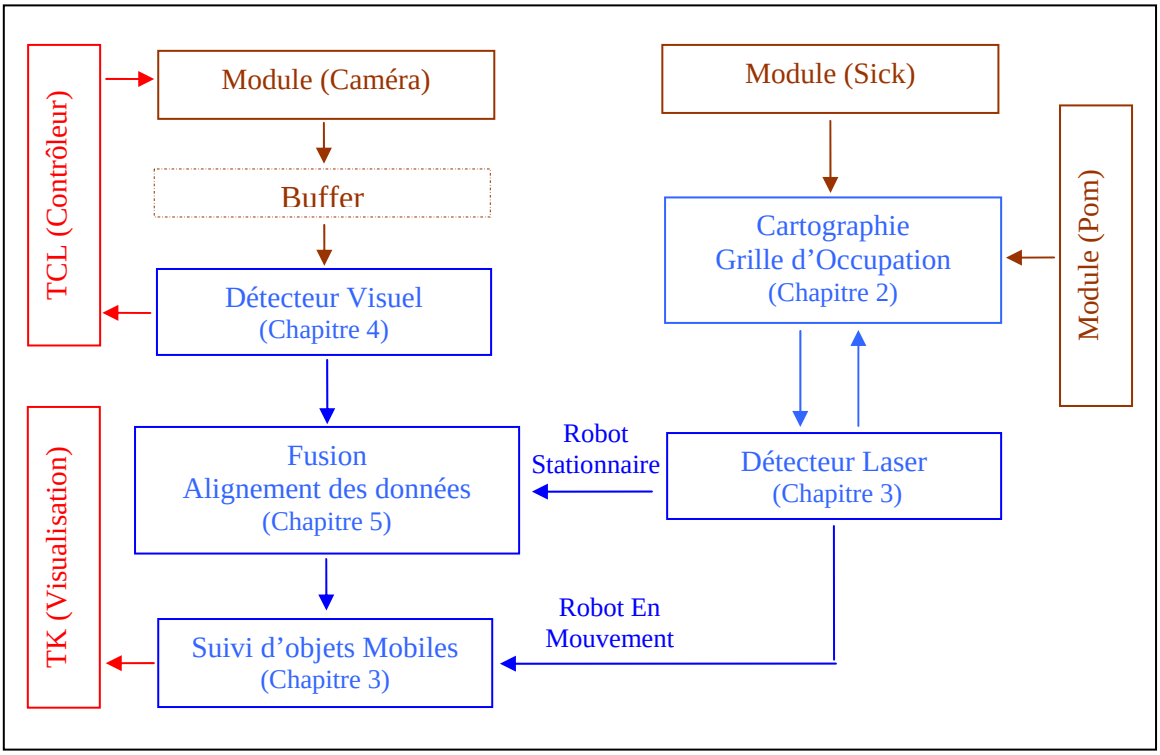


Figure (5.6), Les composants de notre architecture algorithmique et l'organisation de ses fonctions. Les modules préexistant (camera, sick, pom) sont en marron, les fonctions qui représentent les composantes essentielles de notre module (SMT, Simultaneous Mapping and Tracking) sont en bleus, et nos scripts (Tcl/Tk) sont en rouge.

5.6. Conclusion

Dans ce chapitre nous avons fusionné les données de deux sources sensorielles (laser et vision panoramique) pour effectuer la détection et le suivi des cibles mobiles. L'alignement de données représente le sujet sur lequel nous avons focalisé notre intérêt. Afin de réaliser une bonne performance, nous avons proposé une technique probabiliste qui utilise, à chaque fois que le système reçoit une direction d'une cible fournie par le laser, un modèle de pondération gaussien pour choisir le « cas » de poids le plus important parmi les dernières dix mesures captées par la caméra, pour chaque cible. Le modèle gaussien est aussi utile pour calculer la direction finale de la cible détectée après avoir effectué la fusion. Ce dernier modèle nous propose en même temps une bonne méthode pour analyser les situations de convergence et celles de divergence entre chaque couple de directions fusionnées, ce qui nous donne en conséquence un certain indice pour analyser la qualité de la fusion et caractériser le système. Les considérations expérimentales techniques ainsi que des résultats sont aussi expliqués et commentés.

Conclusion Générale

Les travaux présentés dans cette thèse traitent du problème de cartographie et localisation simultanées (SLAM). L'idée principale est de réaliser une carte pouvant être mise à jour à une fréquence suffisamment élevée pour bien décrire un environnement dynamique. Puisqu'une bonne carte doit être stable, les traces des cibles mobiles doivent en être retirées. La détection des mouvements des cibles et le suivi de leurs trajectoires devient donc indispensable. Plusieurs approches qui utilisent plusieurs types des capteurs sont expérimentées, et les résultats sont illustrés dans tous les chapitres qui composent ce mémoire.

Chapitre 1 :

Pourquoi faire la cartographie dans une application robotique ? Cette question trouve peut être sa réponse dans ce chapitre qui montre l'importance de disposer d'une carte de l'environnement correcte pour qu'un robot mobile puisse se localiser effectuer une navigation efficace.Trois types des cartes sont discutées (géométrique, topologique, grille d'occupation). Une étude bibliographique pour chacune d'entre elle est présentée. Nous avons choisi la grille d'occupation parce qu'elle nous apparaît comme un modèle capable de décrire un environnement dynamique. Un état de l'art portant sur les sujets discutés dans cette thèse est aussi présenté.

Chapitre 2 :

Ce chapitre rappelle les règles probabilistes de base, l'hypothèse de Markov et le filtre de Bayes. Il présente aussi le principe et le modèle du capteur utilisé (le télémètre laser). Une grille d'occupation construite pour une partie de notre laboratoire est illustrée (dans un environnement statique). Le filtre, DBF (Discrete Bayes Filter DBF) est employé pour calculer et mettre à jour la probabilité d'occupation des cellules de la grille. Afin d'éviter l'ambiguïté qui peut associer le calcul d'une seule valeur probabiliste comprise entre zéro et un pour décider de l'état d'occupation d'une cellule (libre, inconnue ou occupée), nous avons introduit une pondération entre trois états probabilistes partiels se partageant l'état de la cellule et tels que leur somme est égale à un. L'approche discutée ne prend pas en compte l'occupation des cellules voisines, ce qui est bien adapté à l'utilisation du laser dont le faisceau fin ne couvre pas deux cellules (chacune étant de 5x5 cm^2).

Chapitre 3 :

La capacité à extraire les parties statiques de son entourage afin de construire une carte propre dans un environnement inconnu qui contient des entités mobiles, représente une capacité essentielle pour garantir une bonne fonctionnalité pour un robot mobile. Il est toujours nécessaire pour le robot, de détecter les objets se déplaçant dans sa zone de perception, de les suivre, et éventuellement de prédire leur mouvement pour améliorer son interaction avec eux.

Dans ce chapitre, nous avons introduit le problème du suivi en présentant la géométrie classique d'un système se composant d'une cible et d'un observateur (en gardant la généralité du problème). Ensuite, nous avons proposé un modèle pour la dynamique du système. Puis, nous avons adoptée une approche modifiée pour faire le suivi d'objets mobiles pendant que le robot construit sa carte a également été présentée. Les techniques de filtrages qui sont essentielles pour traiter ces problèmes sont brièvement exposées, et nous avons détaillé une variante de filtrage particulaire (MMPF, Multiple Model Particle Filter) qui est adéquate pour le modèle de dynamique que nous avons proposé. Nos stratégies pour faire le groupage des cellules voisines et occupées sous forme des masses ou de segments pour les considérer comme une seule cible, puis faire la détection des groupes mobiles, ainsi que notre approche pour le suivi multi-cible, ont été aussi discutées. Finalement, nous avons montré quelques résultats d'expérimentations effectuées dans notre laboratoire.

Chapitre 4 :

Grâce à leur richesse, l'utilisation de capteurs visuels est très fréquente en robotique. Afin d'améliorer les performances de l'approche expliquée dans le troisième chapitre qui souffre d'un certain manque de fiabilité pour la détection de cibles mobiles dans des environnements très peuplés, nous avons proposé des techniques basées sur des images panoramiques pour réaliser une étape de détection visuelle. Nous avons présenté les expériences que nous avons effectuées pour la détection d'objets mobiles en utilisant un système de vision catadioptrique. Le fond de scène (Background) est extrait à partir de trois images successives (prises avec un pas de temps constant entre elles) en employant le (Temporal Median Operator). L'image binaire est calculée en utilisant une une approche modifiée de flot optique où un nouveau terme qui compense les changements d'illumination est pris en compte, de manière à tenir compte de l'évolution de l'environnement. La détermination des directions des cibles détectées à partir d'images déployées (sur une forme cylindrique) et à partir d'images panoramiques brutes, est effectuée en employant deux stratégies qui sont respectivement présentées et comparées.

Chapitre 5 :

D'un point de vue général, chaque capteur a son propre mécanisme de fonctionnement et ses propres traitements qui aident à traduire les signaux qu'il reçoit en information utile. Garantir au robot une meilleure compréhension des événements qui se déroulent autour de lui nécessite la fusion des données de ses capteurs pour dépasser leurs limites respectives.

La question cruciale (sur laquelle ce chapitre s'est concentré) est celle de la synchronisation des capteurs pour que la fusion porte sur les mêmes événements. Ce chapitre a présenté une technique probabiliste que nous avons proposée pour la fusion des données fournies par le laser et la caméra catadioptrique. A chaque fois que le système reçoit une direction d'une cible fournie par le laser, une étape de pondération se déclenche pour évaluer la possibilité de la fusionner avec l'une des dix dernières mesures captées par la caméra et stockées dans un registre qui concerne une seule cible. Le modèle gaussien de pondération est utilisé pour calculer la direction finale de la cible détectée et faire la fusion.

Plusieurs directions peuvent être considérées pour améliorer les résultats de nos travaux ou pour avancer vers de meilleures performances de la cartographie et de la détection de cibles mobiles. Ainsi, l'utilisation d'images couleur permettrait une meilleure identification des cibles mobiles et une meilleure différenciation entre elles, au prix de traitements plus complexes, mais tout à fait envisageables, y compris en temps réel. Comme il s'agit de détecter et de suivre ces cibles globalement et non de les modéliser dans le détail, les traitements d'image peuvent se limiter à l'extraction de points d'intérêts ou de formes simples et à rechercher les corrélations d'une image à la suivante. D'autre part, la cartographie et le suivi de cible devraient s'effectuer pendant le mouvement continu du robot. Une extension directe de nos travaux serait d'intégrer le mouvement du robot pour le suivi dans l'image panoramique, de la même manière que nous le faisons pour le suivi avec le télémètre laser. Parmi les principales applications qui peuvent bénéficier de nos travaux ont peut citer par exemple les véhicules intelligents. Le filtre présenté dans le troisième chapitre est utile par exemple pour informer l'observateur (dans ce cas une voiture) de la probabilité d'avoir un point d'intersection avec la trajectoire d'un cible mobile et donc permettre d'éviter une collision possible. Une autre application est la robotique domestique qui nécessite d'effectuer une cartographie dans un environnement dynamique parce que peuplé et de détecter les cibles mobiles que sont les humains, de suivre leurs trajectoires, en vue de permettre une meilleure interaction homme-robot en particulier. Les travaux et les résultats exposés dans ce mémoire sont un pas dans une longue démarche de recherche et de développements engagée par la communauté robotique dans cette direction.

Références:

[1] G. C. Anousaki and K. J. Kyriakopoulos. Simultaneous localization and map building for mobile robot navigation. In IEEE Robotics and Automation Magazine, septembre 1999.

[2] E.G. Araujo and R.A. Grupen. Feature detection and identification using a sonar Array. In Proc. Of 1998 IEEE Intl. Conf. on Robotics and Automation, May 1998.

[3] D. Arbuckle, A. Howard and M. Mataric. Temporal Occupancy Grids: a Method for Classifying the Spatio-Temporal Properties of the Environment, In Proc. of IEEE/RSJ Intrl. Conf. on Intelligent Robots and Systems (IROS'04), 2004.

[4] A. Arleo, J. del R. Millan, and D. Floreano. Ecient learning of variable-resolution cognitive maps for autonomous indoor navigation. In IEEE Transactions on Robotics and Automation, d_ecembre 1999.

[5] K.O. Arras, R. Philippsen, M. de Battista, M. Schilt, and R. Siegwart. A navigation framework for multiple mobile robots and its applications at the Expo.02 exhibition. In Proc. of the IROS-2002 Workshop on Robots in Exhibitions, 2002.

[6] N. Ayache and O.D. Faugeras. Maintaining representations of the environment of a mobile robot. IEEE Transactions on Robotics and Automation, 1989.

[7] A. Baba , R.Chatila. "Data fusion of panoramic images and laser scans for dynamic targets detection" Rapport LAAS N°07173, Avril 2007, 6p. (Soumis à ICRA 2008)

[8] A. Baba and R.Chatila "Dynamic Targets Detection for Robotic Applications Using Panoramic Vision System". Rapport LAAS No07126, The 13th International Conference on Advances Robotics (ICAR 2007), Jeju (Corée), 21-24 Août 2007, pp.308-313. Selected to be published in the Springer-Verlag Lecture Notes in Control and Information Sciences (LNCIS); the Edition is entitled "Recent Progress in Robotics; Viable Robotic Service to Human".

[9] A. Baba and R. Chatila. Experiments with simultaneous environment mapping and multi-target tracking. 10th International Symposium on Experimental Robotics 2006.

[10] A. Baba and R. Chatila. Simultaneous Environment Mapping And Mobile Target Tracking. IAS-9, IOS Press, 2006, ISBN 1-58603-595-9, pp.145-152.

[11] P. Bai, H. Qiao, A. Wan and Y. Liu, Person-Tracking with Occlusion Using Appearance Filters. In Proc. of the IEEE Inrl. Conf. on Intelligent Robots and Systems (IROS), 2006.

[12] S. Baker et S. K. Nayar. A theory of single-viewpoint catadioptric image formation, Int. Journal of Computer Vision, 35(2) :1–22, Novembre 1999.

[13] J. Barreto et H. Araujo. Geometric properties of central catadioptric line images. 7th European Conf. on Computer Vision, ECCV'02, pages 237–251, Copenhagen, Danemark, Mai 2002.

[14] J.L. Barron, D.J.Fleet and S.S. Beauchemain, Performance of Optical Flow Technic. International Journal of computer vision, 12(1), pp. 43-77,1994.

[15] N. Bellotto and H. Hu "Vision and Laser Data Fusion for Tracking People with a Mobile Robot " IEEE ROBIO 2006.

[16] M. Bennewitz, W. Burgard, G. Cielniak and S. Thrun. Learning Motion Patterns of People for Compliant Robot Motion, International Journal of Robotics Research, Vol. 24, No. 1, pp. 31-48, 2005.

[17] M. Bennewitz,W. Burgard, and S. Thrun. Using EM to learn motion behaviours of persons with mobile robots. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002.

[18] M. Bennewitz, G. Cielniak, and W. Burgard, "Utilizing learned motion patterns to robustly track persons". In Proc. of Joint IEEE Int. Workshop on VS-PETS, Nice, France, 2003, pp. 102.109.

[19] S. Betgé-Brezetz, R. Chatila and M. Devy. Natural scene understanding for mobile robot navigation, In Proc. of IEEE Intrl. Conf. on Robotics and Automation (ICRA'94), 1994

[20] D. Beymer and K. Konolige, "Tracking people from a mobile platform,. in Proc. of IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics," Seattle, WA, USA, 2001.

[21] D. Beymer and K. Konolige, "Real-time tracking of multiple people using continous detection". in Proc. of the Int. Conf. on Computer Vision, Kerkyra, Greece, 1999.

[22] A. Birk, S. Carpin. Merging occupancy grid maps from multiple robots, In Proc. of the IEEE Publication, Volume: 94, Issue: 7, ISSN: 0018-9219, July 2006

[23] J. Borenstein and Y. Koren. Histogramic in-motion mapping for mobile robot obstacle avoidance.In IEEE Transactions on Robotics and Automation, aôut 1991.

[24] G. A. Borges and M.-J. Aldon. Optimal pose estimation using geometrical maps. In IEEE Transactions on Robotics and Automation, février 2002.

[25] R.A. Brooks, Aspects of mobile robot visual map making. In Proc. of 2nd International Symposium on Robotics Research, Kyoto, Japan, 1984.

[26] A.M. Bruckstein & T.J. Richardson. Omniview Cameras with Curved Surface Mirrors. In OMNIVIS00, pages 79–84, 2000.

[27] H. Bulata and M. Devy. Incremental construction of a landmark-based and topological model of indoor environments by a mobile robot. In IEEE International Conference on Robotics and, Automation (ICRA'96), 1996.

[28] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences, with an interactive museum tour-guide robot. Artificial Intelligence, 114(1-2), 2000.

[29] L. Cahut, K.P. Valavanis, and H. Delic. Sonar resolution-based environment mapping. In Proc. Of 1998 IEEE Intl Conf. on Robotics and Automation, May 1998.

[30] J. A. Castellanos, J. M. M. Montiel, J. Neira, and D. Tardos. The spmap : a probabilistic framework for simultaneous localization and map building. In IEEE Transactions on Robotics and Automation, octobre 1999.

[31] J.A Castellanos, J.D. Tardos, and G.Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In Proc. Of 1997 IEEE Intl. Conf. on Robotics and Automation, May 1997.

[32] P. Chakravarty and R. Jarvis "Panoramic Vision and Laser Range Finder Fusion for Multiple Person Tracking" IROS 2006.

[33] R. Chatila and J.P. Laumond. Position referencing and consistent world modelling for mobile robots, In Proc. of IEEE Intrl. Conf. on Robotics and Automation (ICRA'85), 1985.

[34] Y. Cheng, C.-M. Huang, and L.-C. Fu1, Multiple People Visual Tracking in a Multi-Camera System for Cluttered Environments. In Proc. of the IEEE Inrl. Conf. on Intelligent Robots and Systems (IROS), 2006.

[35] K.S. Chong and L.Kleeman. Sonar based map building of a mobile robot. In Proc IEEE Intl. Conf. on Robotics and Automation, 1997.

[36] C. Coué, C. Pradalier, C. Laugier, T. Fraichard and P. Bessière. Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application. The International Journal of Robotics Research Vol. 25, No. 1, January 2006.

[37] J.L Crowley. Navigation for an intelligent mobile robot. IEEE Journal of Robotics and Automation, 1985.

[38] L. Delahoche, C. Pegard, E. M. Mouaddib, and P.Vasseur. Incremental map building for mobile robot navigation in an indoor environment. In Proc. Of IEEE Intl. Conf. on Robotics and Automation, May 1998.

[39] W. Ding, Z. Gong, S. Xie and H. Zou, Real-time Vision-based Object Tracking from a Moving Platform in the Air. In Proc. of the IEEE Inrl. Conf. on Intelligent Robots and Systems (IROS), 2006.

[40] F.Dirk, H. N. Peter, and E. Wolfgang. Robust Background Estimation for Complex Video Sequences. 2001.

[41] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. In IEEE Transactions on Robotics and Automation, volume 17(3), pages 229{241, juin 2001.

[42] N.Dohuu , W.Paquier and R. Chatila. Combining structural description and image-based representation for image, object, and scene recognition, 19th International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh (GB), 30 Juillet - 5 Août 2005, pp.1452-1457.

[43] D. Dufourd, Des cartes combinatoires pour la construction automatique de modèles d'environnement par un robot mobile, thèse, 2006.

[44] A. Elfes. Dynamic control of robot perception using multi-property inference grids, In Proc. of IEEE Intrl. Conf. on Robotics and Automation (ICRA’92), 1992.

[45] A. Elfes. A sonar based mapping and navigation system, In Proc. of IEEE Intrl. Conf. on Robotics and Automation (ICRA’86), 1986.

[46] S. Fleury et M. Herrb, GenoM : Manuel d'utilisation, Rapport LAAS N°01203, Mai 2001.

[47] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. Journal of Artificial Intelligence Research (JAIR), 11:391427, 1999.

[48] J. S. Franco and E. Boyer. Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid. In Tenth IEEE International Conference on Computer Vision, Volume: 2, ISSN: 1550-5499, ISBN: 0-7695-2334-X, 2005.

[49] J. Gasos and A. Martin. Mobile robot localization using fuzzy maps. In Proceedings of the IJCAI'95 workshop - Lecture Notes in Artificial Intelligence, pages 207{224. Springer-Verlag, 1997.

[50] C. Geyer et K. Daniilidis. A unifying theory for central panoramic systems and practical implications. European Conf. on Computer Vision, ECCV’00, volume 29, pages 159–179, Dublin, Irelande, Mai 2000.

[51] Jose-Joel Gonzalez-Barbosa, Vision panoramique pour la robotique mobile : stéréovision et localisation par indexation d'images. Thèse, Rapport LAAS No : 04434, 2004.

[52] N. Gordon, B. Ristic, S. Arulampalam,. Beyond the Kalman filter, particle filter for tracking applications. Artech House, 685 Canton Street, Norwood, MA 02062, 2004.

[53] N J. Gordon, D J. Saolmond and AFM Smith, Novel Approach to non-linear/non-Gaussien bayesian state estimation. In IEE-Proceedings-F, volume 140, pages 107-113, 1993.

[54] D. Hähnel, D. Schulz, and W. Burgard. Mapping with mobile robots in populated environments. In Proc. Of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002.

[55] D. Hähnel, R. Triebel, W. Burgard, S. Thrun. Map Building with Mobile Robots in Dynamic Environments. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2003.

[56] I. Haritaoglu, D. Harwood, and L.S. Davis. "W4: Real time, "surveillance of people and their activities". IEEE Trans. on PAMI, 22(8):809–830, 2000.

[57] M. Hashimoto, S. Ogata, F. Oba and T. Murayama. A Laser Based Multi-Target Tracking for Mobile Robot. IAS-9, IOS Press, 2006, ISBN 1-58603-595-9, pp.135-144.

[58] B. K. P. Horn and B. G. Schunk, Determining Optical Flow, Artificial Intelligence, 17, pp. 185-203, 1981.

[59] I. Horswill, Artificial intelligence and mobile robots: case studies of successful robot systems. The AAAI Press/The MIT Press, Chapter: The Polly System,1998.

[60] A. Howard. Multi-robot mapping using manifold representations. In IEEE International Conference on Robotics and Automation, pages 4198{4203, New Orleans (Louisianne, Etats-Unis), avril 2004.

[61] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. "Detection and location of people using adaptive fusion of color and edge information". In Proceedings of international Conference on Pattern Recognition, 2000.

[62] R. Jain, D. Militzer, and H. Nagel. Separating nonstationary from stationary scene components in a sequence of real world tv-images. IJCAI, pages 612–618, 1977.

[63] J. A. Janet, S. M . Scoggins, M. W. White, J. C. Sutton, E. Grant, and W. E. Snyder. Self-organising geometric certainty maps: A compact and multifunctional approach to place recognition and motion planning. In Proc. IEEE Intl. Conf. on Robotics and Automation, volume 4, 1997.

[64] O. Javed, K.Shafique and M.Shah," A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information" Workshop on Motion and video computing, 2002.

[65] G. Kitagawa, Monte Carlo filter and smoother for non-gaussian non-linear state space models. Journal of computational and Graphical Satitistics, vol. 5, pages 85-90, 2002.

[66] B. Kluge, C. Koehler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2001.

[67] S. Koenig and R. G. Simmons, Artificial intelligence and mobile robots: case studies of successful robot systems. The AAAI Press/The MIT Press, Chapter: A robot navigation Architecture based on partially observable Markov Process Models, 1998.

[68] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In Proc. of the National Conf. on Artificial Intelligence (AAAI-94), 1994.

[69] K. M. Krishna and P. K. Kalra. Detection, Tracking and Avoidance of Multiple Dynamic Objects. Journal of Intelligent and Robotic Systems 33: 371–408, 2002.

[70] B. Kuipers and Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Journal of Robotics and Autonomous Systems, 1991.

[71] Y. D. Kwon and J. S. Lee. A stochastic environment modelling method for mobile robot by using a 2-d laser scanner. In Proc. IEEE Intl. Conf. on Robotics and Automation, April 1997.

[72] J.-C. Latombe. Robot motion planning. Kluwer Academic Publisher, Norwell (Massachusetts), Etats-Unis, 1991.

[73] J.-P. Laumond, editor. La Robotique Mobile. Hermès Sciences Publications, Paris, 2001.

[74] J. H. Lee,T. Tsubouchi, K. Yamamoto and S. Egawa. People Tracking Using a Robot in Motion with Laser Range Finder. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2006.

[75] J. J. Leonard and H. F. Durrant-Whyte. Directed Sonar Sensing for Mobile Robot Navigation. Kluwer Academic Publisher, Cambridge, MA, 1992.

[76] T. S. Levittand D. T. Lawton. Qualitative navigation for mobile robots, AI journal, Vol 44, n° 3, 1990.

[77] M. Lindstr¨om and J.-O. Eklundh, “Detecting and tracking moving objects from a mobile platform using a laser range scanner”. In Proc. Of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2001, pp. 1364.1369.

[78] H. Liu, W.Pi and H. Zha, Motion Detection for Multiple Moving Targets by Using an Omnidirectional Camera, 2001.

[79] M. S. Marzouqi and R. A. Jarvis, Distance Transform Based Gaussian Distribution for Probabilistic Target Tracking. In Proc. of the IEEE Inrl. Conf. on Intelligent Robots and Systems (IROS), 2006.

[80] P. J. McKerrow. Echolocation – from range to outline segments. Journal of Robotics and Automation Systems, 1993.

[81] M. Montemerlo, S. Thrun, W. Whittaker, Conditional particle filters for simultaneous mobile robot localization and people-tracking. Proceedings. ICRA '02. IEEE International Conference on Robotics and Automation, Volume: 1, pp. 695- 701 vol.1, 2002.

[82] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma, Experiences with a Mobile Robotic Elderly Guide for the Elderly. National Conference on Artificial Intelligence, AAAI, August, 2002.

[83] P. Moutarlier and R. Chatila. Incremental free space modelling from uncertain data by an autonomous mobile robot, In Proc. of IEEE/RSJ Intrl. Conf. on Intelligent Robots and Systems (IROS'91), 1991.

[84] P. Moutarlier. Modélisation autonome de l'environnement par un robot mobile. PhD thesis, Laboratoire d'Automatique et d'Analyse des Systèmes du CNRS, Toulouse, France, octobre 1991.

[85] M. Mucientes and W. Burgard. Multiple Hypothesis Tracking of Clusters of People. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2006.

[86] H.H. Nagel, On the Estimation of Optical Flow: Relations Between Different Approaches and Some New Results, Artificial Intelligence, 33, pp. 299-324, 1987.

[87] J. Neira, J. D. Tardos, J. Horn, and G. Schmidt. Fusing range and intensity images for mobile robot localisation. In IEEE Transactions on Robotics and Automation, février 1999.

[88] M. Nixon, and, A. Aguada , Feature Extraction & Image Processing. Newnes (Elsevier) Linacre House, Jordan Hill, Oxford OX2, 8DP 30 Corporate Drive, Burlington, MA 01803. (2002)

[89] G. Oriolo, G. Ulivi, and M. Vendittelly. Fuzzy maps : a new tool for mobile robot perception and planning. In Journal of Robotic Systems, volume 14(3), pages 179{197, 1997.

[90] D. Pagac, E. M. Nebot, and H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. In IEEE Transactions on Robotics and Automation, volume 14(4), pages 623{629, 1998.

[91] Y. Pang, Q. Huang, W. Zhang, Z. Hu, A. H. Rajpar, and K. Li, Real-time Object Tracking of a Robot Head Based on Multiple Visual Cues Integration. In Proc. of the IEEE Inrl. Conf. on Intelligent Robots and Systems (IROS), 2006.

[92] R. Reulke, A. Wehr, R. Klette, M. Scheele, K. Scheibe, Panoramic Mapping using CCD-Line Camera and Laser Scanner with Integrated Position and

Orientation System. Image and Vision Computing NZ, Palmerston North, November 2003.

[93] M. Ribo and A. Pinz. A comparison of three uncertainty calculi for building sonar-based occupancy grids. In Robotics and Autonomous Systems, volume 35(3-4), pages 201{209, 2001.

[94] A. Robert de Saint Vincent. Visual navigation for a mobile robot : Building a map of the occupied space from sparse 3D stereo data, In Proc. of IEEE Intrl. Conf. on Robotics and Automation (ICRA'87), 1987.

[95] G. Salut, Le thème non_linéaire en automatique et traitement du signal. Journées Nationales du GdR "Traitement du signal et images", Chantilly, 25-27 septembre 1989.

[96] D.Scaramuzza, A.Martinelli, et R.Siegwart, A Toolbox for Easy Calibrating Omnidirectional Cameras, Proceedings of the IEEE Inrl. Conf. on Intelligent Robots and Systems (IROS), 2006.

[97] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, ".People tracking with mobile robots using sample-based joint probabilistic data association filters". Int. Journal of Robotic Research, vol. 22, no. 2, pp. 99.116, 2003.

[98] R. Siegwart and I. R. Nourbakhsh. Introduction to autonomous mobile robots. The MIT Press, Cambridge (MA, Etats-Unis), 2004.

[99] C. Stachniss and W. Burgard. An integrated Approach To Goal-directed Obstacle Avoidance Under Dynamic Constraints For Dynamic Environments. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002.

[100] C. Stauffer and W. E. L. Grimson. "Learning patterns of activity using real-time tracking". IEEE Trans. On PAMI, 22(8):747–757, 2000.

[101] P. Stepan, M. Kulich and L. Pireucil. Robust data fusion with occupancy grid, IEEE Transactions on systems, MAN, and Cybernetics-Part C: Applications and Reviews, Vol 35, N° 1, 2005.

[102] R. Szabo. Topological Navigation of Simulated Robots using Occupancy Grid, pp. International Journal of Advanced Robotic Systems, Volume 1, Number 3 (2004), ISSN 1729-8806.

[103] S. Thrun, W. Burgard and D. Fox. Probabilistic Robotics, Cambridge, Mass. MIT Press (20 septembre 2005).

[104] S. Thrun, Learning occupancy grids with forward sensor models, Autonomous Robots, vol. 15, 2003.

[105] T. Veit, F. Cao and P. Bouthemy. Space-time A Contrario Clustering for Detecting Coherent Motions. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2007.

[106] A. Victorino, P. Rives, and J.-J. Borrelly. Localisation d'un robot mobile et cartographie par télémétrie laser. In Journées des Jeunes Chercheurs en Robotique (JJCR'00), Bourges, février 2000.

[107] C.-C.Wang and C. E. Thorpe. Simultaneous Localization And Mapping With Detection And Tracking of Moving Objects. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2002.

[108] D. Wolf and G. S. Sukhatme. Online Simultaneous Localization and Mapping in Dynamic Environments. In Proc. of the Intl. Conf. on Robotics and Automation ICRA, 2004

[109] C. Wren, A Azarbayejani, T. Darrel, and A. Pentland. “Pfinder, real time tracking of the human body”. IEEE Trans. on PAMI, 19(7), 1997.

[110] B. Yamauchi. A frontier-based approach for autonomous exploration. In Proc. of the IEEE Interl. Symposium on Computational Intelligence in Robotics and Automation, 1997.

[111] A. Zelinski. A mobile robot exploration algorithm. In IEEE Transactions on Robotics and Automation,pages 707{717, décembre 1992.

[112] L. Zhang and B. K. Ghosh. Geometric feature based 2 1/2 map building and planning with laser, sonar and tactile sensors. In International Conference on Intelligent Robots and Systems (IROS'00),2000.