Navigation visuelle de robots mobiles dans un environnement d'intérieur

Haythem Ghazouani

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II

– SCIENCES ET TECHNIQUES DU LANGUEDOC –

THÈSE

Pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ MONTPELLIER II

Discipline : Génie Informatique, Automatique et Traitement de Signal

Formation Doctorale : Systèmes Automatiques et Microélectronique

Ecole Doctorale : Information, Structures et Systèmes

par

Haythem GHAZOUANI

Titre :

Navigation visuelle de robots mobiles dans un environnement d'intérieur

Soutenue publiquement le 12 décembre 2012

JURY

Ouajdi KORBAA	Professeur, Université de Sousse	Président du Jury
Imed Riadh FARAH	Maître de conférences, Université la Manouba	Rapporteur
Youcef MEZOUAR	Professeur IFMA, Institut Pascal	Rapporteur
Michel DEVY	Directeur de recherche, LAAS-CNRS Toulouse	Examinateur
Moncef TAGINA	Professeur, Université la Manouba	Directeur de thèse
René ZAPATA	Professeur, Université de Montpellier 2	Directeur de thèse

... à la mémoire de mon père

REMERCIEMENTS

Je souhaiterais faire part de ma reconnaissance, de façon générale, au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM). Je tiens aussi à remercier M. Michel Robert et M. Jean-Claude König directeurs successifs du LIRMM, pour m'avoir accueilli au sein du laboratoire durant ma thèse.

Je tiens tout particulièrement à remercier mes directeurs de thèse, M. Moncef Tagina et M. René Zapata pour leurs conseils scientifiques très précieux et la confiance qu'il m'ont accordée tout au long de mon travail de thèse. Je vous remercie de m'avoir donné l'opportunité de m'exprimer sur un sujet aussi intéressant qu'enrichissant ainsi que pour la liberté d'action et la confiance qu'ils m'aviez concédées.

Je remercie sincèrement M. Imed Riadh Farah, maître de conférences de l'Université de la Manouba, et M. Youcef Mezouar, maître de conférences de l'Université de Blaise Pascal, Clermont-Ferrand II, qui m'ont fait l'honneur d'être les rapporteurs de mon travail, M. Ouajdi Korbaa, professeur de l'Université de Sousse qui a présidé le jury, ainsi que M. Michel Devy directeur de recherche du LAAS-CNRS Toulouse pour avoir examiné mon travail. Je les remercie pour leur regard critique et pertinent, et leurs conseils.

Un merci chaleureux au peloton de mes amis thésards et ex-thésards, merci pour votre amitié et bonne humeur que je n'oublierai jamais.

Je remercie aussi celle qui a accompagné mon coeur ces dernières années, celle qui a supporté la rédaction, la soutenance, les départs en France, qui sait me réconforter quand rien ne va plus. Mille mercis, Amira, pour tous ces merveilleux moments que tu m'as offerts pendant cette période parfois difficile.

Je remercie aussi ma puce Mayar, pour la belle ambiance qu'elle

nous offre et pour l'amour inconditionnel qu'elle m'accorde, juste en te regardant jouer et grandir, tu m'as donné de l'espoir qui m'a tant aidé dans les moments difficiles.

Je tiens aussi à remercier ma famille, très particulièrement mon frère Heni, mes deux soeurs Hajer et Hiba, ma chère mère Naïma qui, même si la vie n'a pas toujours été facile pour elle, a su parfois se sacrifier pour nous offrir une vie meilleure. Je passerai pas sans oublier la personne qui n'est plus dans ce monde mais qui restera à jamais dans mon coeur, la personne qui m'a toujours aimé sans attendre de retour, qui m'a initié à la vie, je ne vous oublierai jamais et tu resteras à jamais une flamme dans mon coeur qui poussera mes limites. Je souhaite que là où tu es, tu es fier de moi mon cher père.

Enfin, pour finir et pour être sûr de n'oublier personne, je remercie tout le monde qui de loin ou de près m'a aidé à finaliser ce travail.

RÉSUMÉ

Les travaux présentés dans cette thèse concernent le thème des fonctionnalités visuelles qu'il convient d'embarquer sur un robot mobile, afin qu'il puisse se déplacer dans son environnement. Plus précisément, ils ont trait aux méthodes de perception par vision stéréoscopique dense, de modélisation de l'environnement par grille d'occupation, et de suivi visuel d'objets, pour la navigation autonome d'un robot mobile dans un environnement d'intérieur.

Il nous semble important que les méthodes de perception visuelle soient à la fois robustes et rapide. Alors que dans les travaux réalisés, on trouve les méthodes globales de mise en correspondance qui sont connues pour leur robustesse mais moins pour être employées dans les applications temps réel et les méthodes locales qui sont les plus adaptées au temps réel tout en manquant de précision. Pour cela, ce travail essaye de trouver un compromis entre robustesse et temps réel en présentant une méthode semi-locale, qui repose sur la définition des distributions de possibilités basées sur une formalisation floue des contraintes stéréoscopiques.

Il nous semble aussi important qu'un robot puisse modéliser au mieux son environnement. Une modélisation fidèle à la réalité doit prendre en compte l'imprécision et l'incertitude. Ce travail présente une modélisation de l'environnement par grille d'occupation qui repose sur l'imprécision du capteur stéréoscopique. La mise à jour du modèle est basée aussi sur la définition de valeurs de crédibilité pour les mesures prises.

Enfin, la perception et la modélisation de l'environnement ne sont pas des buts en soi mais des outils pour le robot pour assurer des tâches de haut niveau. Ce travail traite du suivi visuel d'un objet mobile comme tâche de haut niveau.

Mots clefs : Robot mobile, Vision stéréoscopique, Distribution de possibilités, Logique floue, Grille d'occupation, Propagation d'erreur, Suivi d'objets, Segmentation.

ABSTRACT

This work concerns visual functionalities to be embedded in a mobile robot for navigation purposes. More specifically, it relates to methods of dense stereoscopic vision based perception, grid occupancy based environment modeling and object tracking for autonomous navigation of mobile robots in indoor environments.

We consider that is important for visual perception methods to be robust and fast. While in previous works, there are global stereo matching methods which are known for their robustness, but less likely to be employed in real-time applications. There are also local methods which are more suitable for real time but imprecise. To this aim, this work tries to find a compromise between robustness and real-time by proposing a semi-local method based on the definition of possibility distributions built around a fuzzy formalization of stereoscopic constraints.

We consider also important for a mobile robot to better model its environment. To better fit a model to the reality we have to take uncertainty and inaccuracy into account. This work presents an occupancy grid environment modeling based on stereoscopic sensor inaccuracy.. Model updating relies on the definition of credibility values for the measures taken.

Finally, perception and environment modeling are not goals but tools to provide robot high-level tasks. This work deals with visual tracking of a moving object such as high-level task.

Keywords : Mobile robot, Stereoscopic vision, Possibility distribution, Fuzzy Logic, Occupancy grid, Error propagation, Object tracking, Segmentation.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

Au cours des derniers siècles, les progrès épargnant les humains des travaux forcés se sont accélérés. Partout dans le monde, les animaux et les machines effectuent les activités les plus difficiles, fournissant ainsi une vie plus facile, avec plus de sécurité et plus d'indépendance. La volonté de trouver des substituts pour les humains dans des environnements dangereux et pour les activités fatigantes ou répétitives a été l'une des motivations principales pour la recherche de systèmes autonomes et de la robotique.

La robotique est une matière pluridisciplinaire au carrefour de la mécanique, de l'électronique, de l'informatique, des mathématiques mais aussi d'autres disciplines comme les neurosciences ou la psychologie, sans parler du domaine d'application qui s'étend de la microchirurgie à l'exploration spatiale. Les avancées technologiques dans toutes ces disciplines ouvrent aujourd'hui aux chercheurs de nouvelles perspectives et des centaines de voies de recherche à explorer. Cependant, la discipline qui a vraiment poussé les limites de la robotique plus que les autres, surtout dans le domaine applicatif, c'est la vision, ce processus si complexe et si intriguant.

1.1. Motivation

En tant qu'êtres humains, nous percevons la structure tridimensionnelle du monde qui nous entoure avec une facilité apparente. Pensez à quel point la perception tridimensionnelle est fascinante et au même temps intrigante quand vous regardez un vase de fleurs déposé sur la table près de vous. Vous pouvez déduire la forme et la translucidité de chaque pétale à travers les motifs subtils de la lumière et des ombres qui passent le long de sa surface, et segmenter sans effort chaque fleur du fond de la scène. Ou, en regardant un portrait de groupe encadré, vous pouvez facilement compter (même nommer) toutes les personnes sur la photo, et même deviner leurs émotions de leurs apparences faciales. Des psychologues perceptuels ont passé des décennies à essayer de comprendre comment le système visuel fonctionne, et même

s'ils ont su concevoir des illusions optiques et démêler certains de ses principes, une solution complète à cette énigme demeure insaisissable.

Les chercheurs dans la vision par ordinateur ont développé des techniques mathématiques pour la récupération de la forme tridimensionnelle et de l'apparence des objets dans l'imagerie. Nous avons maintenant des techniques fiables pour calculer avec précision un modèle 3D partiel d'un environnement à partir des milliers de photographies partiellement chevauchantes. Étant donné un ensemble assez large d'images prises à partir de différentes orientations d'un objet particulier ou d'une façade, nous pouvons créer des modèles denses exacts de surfaces 3D à l'aide de la stéréo correspondance. Nous pouvons suivre un objet ou une personne se déplaçant sur un fond complexe. Nous pouvons même, avec un peu de succès, essayer de trouver et de nommer toutes les personnes sur une photo en utilisant une combinaison de la détection et la reconnaissance de visage, des vêtements, des cheveux. Cependant, malgré tous ces progrès, le rêve d'avoir un ordinateur capable d'interpréter une image au même niveau qu'un cerveau de deux ans (disons capable de compter tous les animaux dans un tableau) reste insaisissable.

Pourquoi la vision est si difficile ? Dans une partie, c'est parce que la vision est un problème d'inversion, dans lequel nous cherchons à récupérer des inconnues, étant donné des informations insuffisantes pour spécifier complètement la solution. Il faut donc recourir à des modèles basés sur la physique et la probabilité pour lever l'ambiguïté entre des solutions potentielles. Cependant, la modélisation de l'univers visuel dans toute sa riche complexité est beaucoup plus difficile que, par exemple, la modélisation du conduit vocal qui produit des sons parlés. Il est donc plus judicieux de se focaliser sur des modules bien précis de la vision. C'est pourquoi les travaux qui concernent la robotique et la vision, ne tentent pas de produire la faculté de la vision humaine en totalité, mais plutôt essayent d'embarquer des fonctionnalités visuelles bien précises dans les robots. Pour le moment, aucun travail n'a tenté de reproduire le processus visuel complet. Mais en tentant d'optimiser les différents modules visuels, chacun à part, en réduisant les temps de réponse et en améliorant la précision on arrivera un jour, peut être, à approcher l'oeil humain.

1.2. Définition du problème

Les travaux présentés dans cette thèse concernent le thème des fonctionnalités visuelles qu'il convient d'embarquer sur un robot mobile, afin qu'il puisse se déplacer dans son environnement. Plus précisément, ils ont trait aux méthodes de

perception et de modélisation de l'environnement, et de suivi visuel d'objets, pour la navigation d'un robot mobile autonome.

Les hypothèses qui nous sont imposées sont : le robot ne disposent d'aucune information a priori sur son environnement, il est équipé d'un système d'évitement d'obstacle utilisant ses capteurs sonars et son environnement est un environnement d'intérieur structuré.

La problématique de cette thèse est d'opter pour les bonnes techniques à utiliser pour développer un système de perception visuel pour un robot mobile autonome lui permettant de modéliser au mieux son environnement sans avoir des informations a priori, tout en essayant de doter le robot de tâches visuelles de haut niveau tel que le suivi d'objet mobile.

1.3. Démarche générale

Si quelqu'un vient demander quel est le meilleur détecteur de contours. La première question qu'on lui pose c'est pourquoi ? Quel type de problème, il est en train de résoudre et pourquoi il croit que le détecteur de contours est un composant important. S'il est en train de localiser des visages dans une image, on va lui répondre que les détecteurs de visages les plus réussis utilisent une combinaison de la détection de la couleur de peau et des régions caractéristiques et ne reposent pas sur les détecteurs de contours. S'il cherche à correspondre les contours de portes et des fenêtres dans un bâtiment en vue de la reconstruction 3D, on va lui dire que c'est une bonne idée, mais il serait mieux de régler le détecteur pour détecter des longs contours...

Ainsi, Il est préférable de penser à l'envers, c'est à dire du problème à portée de main aux techniques appropriées, plutôt que de saisir la première technique dont on pourra avoir entendu parler. Ce genre de travail en arrière, des problèmes aux solutions, est typique d'une approche d'ingénierie plutôt qu'à l'étude de la vision. Tout d'abord, nous présentons une définition détaillée du problème et décidons des contraintes et/ou des spécifications pour ce problème ce que nous avons fait dans la section précédente. Ensuite, nous essayons d'en faire sortir les techniques qui ont fait leurs preuves, mettre en œuvre quelques-unes d'entre elles, enfin évaluer leurs performances et faire une sélection. Pour que ce processus fonctionne, ils faut disposer de données de test réalistes, à la fois synthétiques qui peuvent être utilisées pour vérifier l'exactitude et analyser la sensibilité au bruit, et des données du monde réel typiques à la façon dont le système finira par s'habituer. Cependant ce manuscrit n'est pas qu'un texte d'ingénierie (une source de recettes). Il suit

aussi l'approche scientifique abordée pour les problèmes de vision basiques. Là, nous essayons de modéliser au mieux le système à portée de main : comment la scène est créée, comment la lumière interagit avec la scène, et comment les capteurs fonctionnent, y compris les sources de bruit et d'incertitude. La tâche se résume alors dans la manière d'inverser le processus d'acquisition pour trouver la meilleure description possible de la scène. Nous utilisons souvent une approche statistique pour la formulation et la résolution des problèmes de vision par ordinateur. Le cas échéant, des distributions de probabilité sont utilisées pour modéliser la scène et le processus d'acquisition d'images bruitées.

Cette démarche que nous avons décrite et que nous suivons tout au long de cette thèse a des incidences sur la structure des chapitres qui forment le contenu de ce manuscrit. En effet, un chapitre serait dévoué à l'étude des approches reconnues pour faire face à ce genre de problèmes pour pouvoir fixer l'approche globale. Ensuite chaque chapitre est une implémentation d'un module visuel qui renferme, l'étude des techniques possibles, une description de l'approche proposée en précisant l'apport par rapport à l'approche classique ainsi que les résultats expérimentaux sur des données synthétiques et des données réelles si possible.

1.4. Plan et contributions de la thèse

La structure de ce manuscrit est la suivante :

Le deuxième chapitre se concentre sur la présentation des différentes approches de navigation visuelle en mettant l'accent sur celles qui sont spécifiques aux environnements d'intérieur. Nous présentons aussi les travaux précédents pour la modélisation des environnements d'intérieur. Enfin, suivant les contraintes et les spécifications de notre problématique, nous essayons de faire un choix pour les axes principaux de l'approche à adopter.

Le troisième chapitre s'intéresse à la perception de l'environnement par vision stéréoscopique. Dans ce chapitre, nous présentons une nouvelle approche de mise en correspondance. Cette approche se base sur des distributions de possibilités pour la mise en correspondance que nous proposons et présentons en détails dans ce chapitre.

Le quatrième chapitre est consacré à la description de notre méthode de cartographie de l'environnement. Cette méthode se base sur les données requises par la stéréovision pour la construction d'une grille d'occupation. La nouveauté dans la méthode proposée est la représentation incertaine de la propagation de l'erreur

stéréoscopique et l'emploi d'une méthode de mise à jour de la grille, basée sur la crédibilité des nouvelles mesures et des mesures précédentes.

Le cinquième chapitre est consacré à l'étude d'un module visuel permettant au robot de détecter et de suivre un objet de forme régulière coloré. L'originalité dans l'approche qui a été proposée est l'apprentissage hors ligne par segmentation couleur d'une image de la balle ainsi que l'estimation robuste des paramètres réels du cercle.

Enfin, nous conclurons sur l'ensemble de ces travaux dans le dernier chapitre.

Chapitre 2

Les approches visuelles pour la navigation en robotique

2.1. Introduction

La robotique expérimentale est un domaine de recherche qui renferme plusieurs disciplines : ingénierie, informatique, intelligence artificielle, etc... Le but de la robotique est de produire l'indépendance et l'automatisme efficace[Gha07]. De nos jours, les recherches en navigation robotique se penchent vers les approches visuelles. Le compromis entre la puissance de calcul des systèmes embarqués dans les robots et la complexité des algorithmes de vision a poussé les chercheurs à produire des algorithmes plus efficaces et plus intelligents. Le but de la vision artificielle est de donner à une machine des capacités visuelles de façon à pouvoir acquérir des informations sur l'environnement extérieur. La vision artificielle est un résultat direct de l'évolution du traitement d'images et la reconnaissance de motifs [Pra78], [Wat98], [Mez09], [Mez03].

Une ou plusieurs caméras sont nécessaires pour acquérir les images qui vont être utilisées par la suite pour la navigation. Le choix du modèle de la caméra est d'une grande importance pour la navigation, la qualité de l'image ainsi que les résultats du traitement en dépendent. Un bon modèle de la caméra peut réduire les problèmes de distorsion. Le système de traitement d'images doit être un système temps réel pour produire des résultats dans un temps toléré par les applications de navigation.

Deux principales approches de la vision co-existent ; l'approche naturelle et l'approche artificielle. La première approche essaye de reproduire un système de vision naturel inspiré du système humain de vision ou les systèmes de vision des insectes [Gha10b]. La deuxième approche extrait, à partir des images, des informations intelligibles par la machine qui va faire le traitement.

Dans ce chapitre, nous présentons une classification possible des systèmes de vision pour la navigation en robotique. Nous essayons, en même temps de faire nos choix pour l'approche que nous allons adopter pour cette thèse.

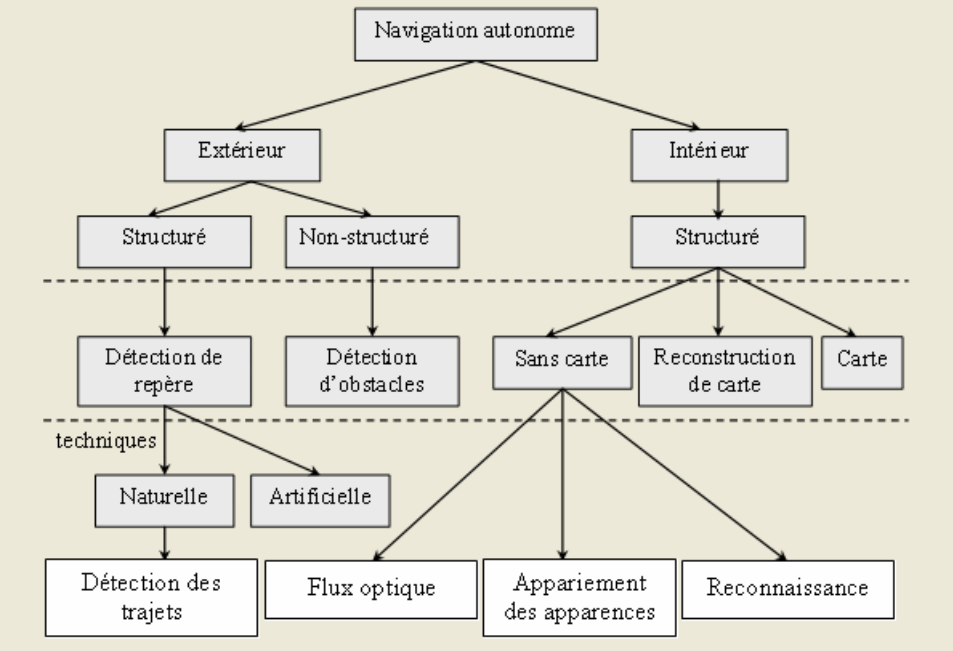

FIGURE 2.1. Schéma de classification des approches visuelles pour la navigation autonome

2.2. Classification des approches de navigation visuelle

Un schéma possible pour classifier les approches visuelles pour la navigation autonome est présenté dans la Figure 2.1. Les systèmes visuels pour la navigation autonome des robots peuvent être classifiés selon le type de l'environnement, les hypothèses et les buts de la navigation, et les techniques utilisées dans les approches de navigation.

L'environnement dans lequel le robot se déplace a une grande influence sur l'approche de navigation. Donc, nous avons une première classification entre la navigation en milieu intérieur et la navigation en milieu extérieur [Mez03]. Les techniques utilisées dans la navigation intérieure ne sont pas les mêmes utilisées dans la navigation extérieure. Pour décrire l'état de l'art dans le domaine de la navigation de robots mobiles, nous faisons une distinction basée sur le premier niveau de classification qui est la structuration de l'environnement :

- Les environnements structurés peuvent être représentés par des primitives géométriques simples (i.e., détection de lignes droites, surfaces planes, couloirs, portes...).
- Les environnements non structurés sont considérés pour des applications en milieu vraiment naturel (site planétaire, polaire, forestier...) ; en milieu terrestre, ce sont de riches sources d'information contextuelle, de couleur et de texture.

– Les environnements semi-structurés sont en essence des environnements naturels qui ont subi une modification partielle de l'homme, typiquement les sentiers ou chemins laissés par des passages fréquents de l'homme ou des animaux, par exemple dans le cadre d'activités agricoles.

Plus précisément, les contributions scientifiques de la navigation visuelle, sont classifiées en deux catégories : la première, la plus prolifique, pour robots d'intérieur et la seconde, en pleine croissance, pour robot d'extérieur.

2.2.1. Navigation visuelle en milieu intérieur

Dès les premiers projets en robotique mobile, les séquences d'images ont été proposées pour fournir des informations utiles à la navigation d'un robot [Gir79]. Elles sont généralement traitées par des méthodes de reconnaissance de formes pour détecter une cible connue par un modèle (apparence de la cible ou modèle géométrique d'un objet) dans les images successives. La plupart de ces approches utilisent des modèles structurés ; le processus de navigation est associé à l'une des modalités suivantes :

2.2.1.1. Utilisation de cartes connues de l'environnement

Plusieurs travaux ont adopté cette approche [Mor85, Far08, Kim94, Kim95, End12, Fal12, Ori95, Ohy98, Zin98, Horn95, Bor96]. Ce sont des systèmes basés sur des modèles géométriques crées par l'utilisateur ou des cartes topologiques de l'environnement. Ces modèles peuvent contenir des degrés différents de détails, variant d'un modèle CAO complet de l'environnement à un simple graphe d'interconnections ou inter-relations entre les éléments de l'environnement. Étant donné que l'idée centrale de la navigation basée sur les cartes est de fournir au robot, directement ou indirectement, une liste d'amers prévus à être rencontrés durant la navigation, la tâche du système de vision est alors de chercher et identifier ces repères dans l'image observée. Une fois identifiés, le robot peut utiliser la carte fournie pour estimer sa propre position (auto-localisation) en faisant correspondre la scène observée avec la description de l'amer dans la base de données. Le calcul impliqué dans la localisation basée sur la vision peut être divisé en les étapes suivantes :

– Acquisition des informations à partir des capteurs : cela revient à acquérir et numériser les images issues des caméras.
– Détecter les amers : généralement cela revient à faire l'extraction des contours des objets, le lissage, le filtrage, et la segmentation des régions en se basant sur les différences des niveaux de gris, couleurs profondeurs, ou mouvements.

– Établir la correspondance entre l'observation et les amers : dans cette étape, le système essaye d'identifier le motif observé dans la scène acquise avec les motifs de la base de données en utilisant des critères de similarité.
– Calcul de la position : une fois une correspondance (ou plusieurs) est identifiée, le système calcule sa position en fonction des positions des motifs identifiés à partir de la base de données.

2.2.1.2. Construction incrémentale d'une carte

Plusieurs travaux ont utilisé la reconstruction de la carte pour la navigation [Mor83, Smi95, Com97]. Ces systèmes utilisent leurs propres capteurs pour construire un modèle géométrique ou topologique de l'environnement, utilisent ensuite ces modèles pour la navigation. Le robot construit son propre modèle de l'environnement en utilisant l'information provenant des différents capteurs (ultrasons, caméras, lasers, etc.) pendant une phase d'exploration. Dans un premier temps, le robot doit acquérir un modèle adapté pour sa propre localisation ; ce sont souvent des cartes stochastiques éparses contenant les représentations paramétriques des amers détectés par le robot tandis qu'il se déplace dans l'environnement. Ces cartes sont construites de manière incrémentale grâce à des techniques d'estimation (filtrage de Kalman, filtrage particulaire...) proposées pour localiser le robot dans l'environnement et pour fusionner les données acquises depuis la position courante. Ces méthodes, très étudiées depuis une quinzaine d'années, traitent donc du problème connu sous le mnémonique SLAM, pour *Simultaneous Localization And Mapping* en anglais. Une fois qu'il a acquis une carte d'amers, le robot sait se localiser ; il peut alors acquérir d'autres représentations, typiquement un modèle de l'espace libre.

2.2.1.3. Navigation dépourvue de carte

Plusieurs travaux s'inscrivent dans cette classe d'approche [Pre92, Ros82, Vez05, San93]. Ce sont des systèmes qui n'utilisent aucune représentation explicite de l'environnement dans lequel la navigation aura lieu, mais plutôt essayent de reconnaître les objets rencontrés dans l'environnement ou de traquer ces objets en générant des mouvements en se basant sur les observations visuelles.

Il est vrai que, pour les approches qui font la construction des cartes automatiquement, il n'y a pas aussi de description a priori de l'environnement. Mais avant que la navigation commence le système doit construire une carte. Dans la catégorie de la navigation sans carte, on inclut la navigation basée sur le flux optique, la navigation basée sur l'appariement des apparences et la navigation basée sur la reconnaissance d'objet.

- Navigation basée sur le flux optique : Plusieurs travaux s'inscrivent dans cette catégorie [San93, Fra05, Sch10]. Vito et al. ont développé dans [San93] un système basé sur le flux optique qui imite le comportement des abeilles. Il est à croire que la position latérale des yeux des insectes favorise un mécanisme de navigation utilisant des caractéristiques dérivées du mouvement plutôt que des informations de profondeurs. Pour les insectes le champ binoculaire traité est extrêmement restreint, donc les informations extraites sur la profondeur sont minimales. Par contre, la parallaxe ou l'incidence du changement de position de l'observateur sur l'observation d'un objet, peut être plus utile spécialement quant l'insecte est en mouvement relatif par rapport à l'environnement. Par exemple, des caractéristiques comme le temps restant pour collision qui dépend de la vitesse est plus significatif que la distance quand l'objectif c'est : sauter à travers l'obstacle. Comme pour le robot *Robee* dans [San93], une approche stéréo divergente a été employée pour imiter le réflexe de centrage de l'abeille. Si l'abeille est au centre d'un couloir, la différence entre la vitesse de l'image vue par l'œil gauche et la vitesse de l'image vue par l'œil droit est approximativement zéro, et l'abeille reste au milieu du couloir. Par contre, si les vitesses étaient différentes, l'abeille se déplacerait vers le côté dont l'image change avec la vitesse la plus faible. Concernant l'implémentation robotique, l'idée basique est de mesurer la différence entre les vitesses des images latérales gauche et droite et utiliser ces informations pour le guidage du robot.
- Appariement basé sur les apparences : Comme exemple de travaux qui ont adopté cette approche on peut citer [Koh10, Oni09]. Une autre manière pour accomplir une tâche de navigation sans recours à une carte de l'environnement, est de mémoriser cet environnement. L'idée est de stocker des images ou des modèles et associer ces images avec des commandes de contrôle qui conduisent le robot à sa destination finale. Une base de données d'images est créée pour être utilisée par le système pour vérifier si le robot est dans une situation déjà rencontrée. Si oui la navigation doit se rappeler du résultat durant l'ancien traitement.
- Reconnaissance d'objets : Parmi les travaux qui ont abordé cette approche on site [Kim95, Lat10]. La reconnaissance d'objet est l'idée de base pour cette approche de navigation. Dans cette approche des commandes comme «se déplacer au bureau devant vous» sont données au robot. Dans ce cas, la commande est très importante pour le système comme elle tient des informations

en elle-même. Par exemple, «se déplacer au bureau devant vous» informe le robot que le repère c'est un bureau et qu'il est situé devant lui.

Tout bien considéré, il paraît que les méthodes géométriques sont les mieux adaptées aux environnements d'intérieur. Généralement, dans la littérature, des modèles mathématiques formels sont proposés dans ce cas. Ces modèles contiennent implicitement des actions d'évitement d'obstacles, de détection d'amers, de construction ou d'actualisation de cartes et d'estimation de la position du robot.

2.2.2. Navigation visuelle en milieu extérieur

Pour les milieux naturels d'extérieur, la construction d'une carte est nettement plus compliquée. D'une part, le niveau de structuration est faible ce qui entraine une extraction de primitives géométriques complexes. D'autre part, la modélisation est difficile quand les scènes changent dynamiquement (à cause de la météo, de la saison, des conditions d'illumination, etc.) ou quand d'autres agents dynamiques interfèrent dans le même environnement. Ce type de navigation peut être divisé en deux classes suivant le type de structuration, i.e., navigation en environnements structurés ou non structurés.

Les environnements non structurés manquent de primitives géométriques régulières (lignes blanches bien délimitées, largeur de la voie relativement constante, etc.) qui sont exploitées dans les approches décrites ci-dessus pour effectuer la navigation. Ce type de milieu contient plutôt des chemins de terre à la campagne, des terrains accidentés ou n'importe quelle zone traversable par un véhicule (robot) tout-terrain. C'est dans cette catégorie que se situent tous les projets dédiés à l'exploration planétaire [Wil92, Cha95] utilisant des ROVERs avec locomotion tout-terrain et avec un niveau élevé d'autonomie.

Dans les applications terrestres, le robot exécute généralement une tâche prédéfinie, comme le suivi d'un élément qu'il doit reconnaître dans l'environnement tout au long d'une région navigable [Lor97]. La détection des zones navigables [Bet96a] et la détection d'obstacles lors du déplacement du robot [Mal00] exploitent la construction et la fusion des cartes de traversabilité. Pour pouvoir exécuter une tâche de navigation, le robot requiert des fonctions additionnelles pour la détection et le suivi d'amers (discontinuité sur la ligne d'horizon, bâtiments, un grand arbre . . .) exploités pour se localiser ou pour exécuter des commandes asservies.

C'est ici que la vision intervient par des techniques de segmentation, de classification par texture et couleur dans l'image segmentée. Ces techniques sont les plus adaptées pour maintenir le véhicule sur la région navigable. Cependant, il est très rare de trouver des travaux qui rapportent l'utilisation de la texture dans le

contexte de l'évitement d'obstacles ou de la navigation [Bet96b]. Par exemple, Fernandez [Fer95] présente une approche pour la détection rapide et automatique de routes, en utilisant une segmentation de l'image par une analyse de texture sur une architecture de réseau de neurones.

En outre, les effets de l'instabilité colorimétrique sur les capteurs sont plus prononcés à l'extérieur car l'information visuelle est fort dépendante de la géométrie (direction et intensité de la source lumineuse) et de la couleur (distribution de la puissance spectrale) de la lumière. Il est clair que la constance de la couleur à l'extérieur est très compliquée à obtenir surtout avec des conditions atmosphériques imprévisibles. Certaines approches [Cel02] préfèrent ainsi contourner ce problème en exploitant l'opposition ou le rapport de couleurs ; par exemple, pour atténuer les effets causés par des variations d'illumination, les rapports R=G et B=G ou l'espace rgb normalisé sont souvent utilisés.

P. Lasserre a présenté dans sa thèse un des travaux les plus connus de la navigation visuelle en milieu naturel [Las96]. Elle utilise une caméra vidéo pour obtenir des informations nécessaires à la localisation et à la navigation du robot dans son environnement. Elle a travaillé sur deux approches : l'obtention de l'information tridimensionnelle à partir d'un système stéréoscopique d'une part, l'identification de la nature des objets contenus dans la scène d'autre part.

Al Haddad, dans sa thèse [Had98], considère la génération autonome de déplacements, à partir des informations fournies par une paire de caméras monochromes. Il a poursuivi les développements sur la stéréovision exploitée pour obtenir en ligne des données tridimensionnelles sur l'environnement, à partir desquelles une carte locale d'obstacles est déterminée : il a proposé un algorithme de stéréo-corrélation, technique qui est adaptée pour des scènes texturées. Il a proposé aussi deux méthodes de génération de mouvements : l'une réactive, et l'autre « pseudo-planifiée ». Ces déplacements sont enchaînés afin d'atteindre un but situé à quelques dizaines de mètres, dans un environnement essentiellement plan.

R. Murrietta Cid a continué les travaux sur l'interprétation des informations obtenues à partir de la vision monoculaire par une caméra vidéo couleur [Mur98]. En effet, la connaissance du terrain ou des objets situés dans un environnement d'extérieur peut être améliorée en ajoutant des informations telles que la couleur ou la texture. Parra et Murrietta ont présenté dans [Mur01] une approche de navigation qui combine les informations de profondeur issues de la stéréo, et les informations bidimensionnelles de la vision couleur, dans le but de repérer et suivre pendant le mouvement du robot des amers dont le type est connu. La méthode développée est fondée sur (1) la segmentation d'une image de profondeur en régions correspon-

dant au terrain (surface uniforme) et aux entités qui en émergent (roches, chemin, végétation, . . .), (2) sur l'extraction de caractéristiques sur ces régions, basées sur la couleur et la texture, (3) sur l'identification de la nature du terrain (terre, herbe) et des objets qui en émergent (rocher, arbres) et (4) sur le suivi des objets utiles pour le repérage du robot. Notons que ces travaux n'ont pas été intégrés sur un robot, ce qui en limite singulièrement la portée.

En bref, du fait de la richesse des images, l'utilisation de la vision artificielle en robotique revêt une importance toute particulière car elle permet de fournir à la machine les capacités nécessaires pour réagir avec son environnement. Dans ce cadre, on peut trouver des méthodologies d'extraction d'indices visuels dans les images permettant d'obtenir un guidage efficace et/ou une localisation précise d'un robot mobile par rapport à son univers.

Nous allons traiter dans cette thèse, des environnements intérieurs classiques, bien illustrés par nos laboratoires : un réseau de couloirs qui desservent des pièces. Pour naviguer dans ces types d'environnements, nous ne disposons d'aucune représentation a priori. Notre système doit, tout d'abord, construire un modèle de son environnement pendant une étape d'exploration. Les représentations de l'environnement les plus utilisées dans la littérature son décrites dans la section suivante.

2.3. Modélisation de l'environnement d'intérieur pour la navigation

Dans la robotique mobile, une multitude de représentations de l'environnement a été employée par les chercheurs. Plusieurs types de cartes ont été développés. Les travaux récents pour la modélisation des environnements d'intérieur utilisent les capteurs RGB-D [Hen12, Fal12, End12]. Il existe un consensus général sur le fait que ces différents types ont des avantages et des limitations et qu'ils sont plus ou moins adaptés selon la mission à accomplir. Par exemple, les cartes métriques sont difficiles à élaborer et à mettre à jour à cause de l'incohérence entre le mouvement du robot mobile et la perception. Elles sont moins adaptées pour les problèmes symboliques. En revanche, les cartes topologiques sont mieux adaptées pour représenter les grands environnements, pour rajouter un niveau symbolique ou pour communiquer avec l'homme. Mais ces cartes permettent seulement une localisation globale, et une planification de la trajectoire de façon sous optimale. Lors de la construction des cartes topologiques, la distinction des différentes composantes est difficile sans l'utilisation d'informations métriques.

Partant de cette analyse, ces représentations sont, la plupart des cas, complémentaires. De ce fait, vient la notion des approches hybrides telles que les approches métriques et topologiques. En particulier, l'utilisation conjointe de deux types d'approches est susceptible de favoriser l'exploitation de la carte résultante pour les besoins de navigation du robot. C'est pourquoi les approches hybrides, qui combinent différents types de modèles élémentaires, se généralisent.

2.3.1. Cartes métriques

L'approche métrique a pour but de générer une carte géométrique plus ou moins détaillée de l'environnement à partir des données perçues. Les informations de longueur, distance, position etc., sont explicites dans les cartes métriques et sont en général définies dans un référentiel unique. Ces cartes sont souvent plus faciles à interpréter par l'homme car elles offrent une relation bien définie avec le monde réel [Thr02].

L'élaboration des cartes métriques nécessite la gestion de la cohérence géométrique de l'ensemble de la représentation. En particulier lorsque le robot retourne sur un lieu déjà visité après avoir réalisé une boucle dans l'environnement. Ce phénomène qu'on appelle *la fermeture de la boucle* est l'un des défis majeurs de la construction des cartes métriques. Plusieurs raisons permettent de l'expliquer :

- Dans certains cas (par exemple dans le cas des grilles d'occupation classiques), le système ne peut corriger les positions antérieures du robot (et donc indirectement des éléments observés par le robot à ces positions antérieures), ce qui paraît pourtant indispensable pour assurer une bonne superposition des données présentes et passées ;
- Parfois, le robot ne maintient qu'une seule hypothèse sur sa position (par exemple dans le cas des filtres de Kalman où la position du robot est modélisée par une distribution gaussienne unimodale) : il ne peut pas gérer correctement certaines ambiguïtés de positionnement ;
- Si le robot ne tient pas vraiment compte de l'incertitude sur sa position, l'appariement entre l'observation et la carte courante est perturbé. En effet, cet appariement est souvent réalisé par des méthodes de descente de gradient qui nécessitent une bonne estimation initiale de la position du robot , alors que l'incertitude sur cette position peut croître sans borne lors du parcours du cycle.

Par conséquent, la gestion des cycles est l'un des enjeux majeurs de la construction de cartes métriques puisque en plus de la détection de la fermeture, elle nécessite la gestion de la cohérence géométrique de l'ensemble de la représentation.

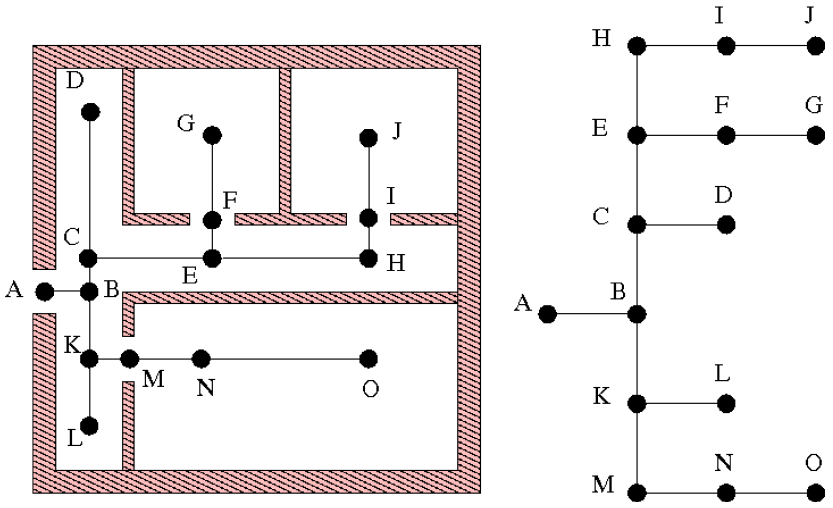

FIGURE 2.2. Carte métrique de l'environnement avec des lieux topologiques reliés entre eux par des relations d'adjacence et la carte topologique correspondante.

2.3.2. Cartes topologiques

L'approche topologique vise à construire une représentation plus abstraite illustrant les relations entre les éléments de l'environnement, sans utiliser un référentiel absolu [Fab02]. Une carte topologique est un graphe dont les sommets correspondent à des places, souvent associés à des informations perceptuelles reconnaissables par le robot, et dont les arêtes indiquent qu'il y'a un chemin entre les deux sommets qui est traversable par le robot (voir la Figure 2.2). Il n'existe pas de convention pour les cartes topologiques et la symbolisation des sommets et arêtes n'est pas semblable d'une approche à l'autre. Plus précisément, les principales différences entre représentations se situent ainsi dans la nature et la densité des sommets et la manière avec laquelle ces sommets sont ajoutés dans la carte ainsi que dans la nature des arêtes.

L'avantage principal des approches topologiques est l'absence d'incertitude sur le mouvement des robots : les incertitudes ne s'accumulent pas globalement car le robot effectue une navigation locale, entre endroits. Ces cartes peuvent présenter une séparation des éléments de l'espace qui facilite leurs compréhension par l'homme, notamment si les lieux symbolisent des pièces ou des couloirs : par exemple, on peut imaginer de donner l'ordre au robot d'aller à la cuisine plutôt qu'au point (x, y).

En revanche, la principale limitation des approches exclusivement topologiques est l'absence d'informations spatiales. Ce manque d'informations peut empêcher le robot de réaliser des raisonnements géométriques sur l'ensemble de son envi-

ronnement. Plus précisément, le robot peut avoir des difficultés à sélectionner la trajectoire optimale entre deux lieux [Thr02] : d'une part, le manque d'information sur les mesures l'empêche de choisir entre deux chemins du graphe qui mènent au même endroit, et d'autre part, il n'est pas possible de trouver un chemin plus direct dans l'espace métrique bidimensionnel que ceux qui sont implicitement codés dans les arêtes du graphe. Si les sommets sont impossibles à distinguer, la construction d'une carte purement topologique nécessite d'élaborer un processus très coûteux et peu efficace lors de l'ajout d'un nouveau sommet [Duf05].

2.3.3. Grilles d'occupation

Dans la représentation de l'environnement sous forme d'une grille d'occupation, l'espace est partitionné en un ensemble de cellules distinctes. Un vecteur d'attributs (éventuellement un seul nombre ou un seul bit dans le cas de cartes binaires) est attaché à chacune des cellules pour représenter ses propriétés : souvent, il s'agit du degré d'encombrement par un obstacle (indice indiquant que la cellule correspondante est occupée ou non par un obstacle).

Les grilles d'occupation constituent une représentation surfacique simple et populaire. Dans ce type de modèle, l'espace est discrétisé selon une grille régulière en cellules carrées ou rectangulaires de même taille. Chaque cellule contient un indice (probabilité, histogramme, etc.) indiquant si l'espace correspondant est plutôt libre ou occupé. Elles fournissent en outre une estimation statistique de la confiance dans les données, et certaines approches permettent même de détecter les zones conflictuelles ou les régions qui nécessitent des compléments d'observation. De plus, contrairement aux représentations composées de scans lasers bruts, elles génèrent des informations *de plein et de vide* puisqu'elles indiquent directement où sont placés les obstacles. Elles sont par ailleurs relativement aisées a interpréter par l'homme même si elles présentent parfois des zones floues et ambiguës : dans le cas bayésien notamment, il est difficile de déterminer si ces zones imprécises sont dues à un manque d'information ou a la présence d'informations contradictoires.

Concernant leurs méthodes de constructions, les grilles d'occupation sont plutôt économiques en ressources de calcul : leur mise a jour s'avère rapide et facile. Il est de plus possible de les construire en ligne, sans contrainte particulière sur la trajectoire à suivre pour le robot. Enfin, les grilles d'occupation permettent une intégration aisée de divers types de capteurs : l'algorithme de fusion est immédiat. Elles sont de plus bien adaptées a des capteurs bruités (vision stéréoscopique, sonars, radars...) où les primitives géométriques sont assez difficiles à extraire du fait de l'incertitude sur les données.

L'avantage principal des grilles d'occupation est leur capacité à représenter l'espace de manière très dense, en fonction du pas de discrétisation de la grille. Elles sont adaptées à des environnements de forme quelconque, et elles donnent une estimation statistique ou probabiliste de la confiance dans les données. De plus, elles fournissent des informations d'occupation et donc sur les positions des obstacles. Ainsi, elles sont souvent utilisées lorsque l'application visée repose sur la connaissance de l'espace libre, en particulier pour la planification de trajectoires (à partir de transformations en distance ou de champs de potentiels par exemple). Elles sont en général relativement faciles à interpréter par l'homme.

Le majeur inconvénient des grilles d'occupation se situe dans leur manque de compacité : elles sont mieux adaptées à la représentation d'environnements encombrés et inefficaces dans les grands espaces vides. De plus, la résolution de la discrétisation étant prédéfinie, elles sont incapables de s'adapter automatiquement à la densité de l'espace occupé. Par conséquent, si les grilles d'occupation se prêtent bien à certaines approches de planification, elles peuvent cependant se révéler inefficaces du fait du manque d'adaptation à l'échelle de l'espace de travail (plusieurs cellules dans des grands espaces libres).

2.3.4. Modèle de primitives

Guivant et al. [Gui04] définissent les primitives géométriques (*features* en anglais) pouvant servir de balises (on parle aussi d'amers suivant un jargon maritime couramment employé en robotique) comme des parties distinctives de l'environnement facilement reconnaissables via un type de capteur donné et qui admettent une description paramétrique : notons qu'on parle généralement de balises quand il s'agit d'amers artificiels.

Un amer géométrique pour la localisation doit avoir des caractéristiques spécifiques : pouvoir discriminant, domaine de visibilité important, stabilité, invariance aux changements métrologiques et lumineuses. Pour garantir la cohérence de la carte de l'environnement, les modèles de primitives doivent tenir compte des incertitudes sur les primitives géométriques qui sont généralement représentées par des distributions gaussiennes sur les paramètres géométriques de la primitive (positions cartésiennes ou polaires, longueurs, etc.).

La représentation par amers fournit un cadre de travail mieux adapté pour résoudre le problème de la Cartographie et Localisation Simultanées (SLAM), à travers les approches basées sur le filtre de Kalman étendu (EKF) : la fermeture de boucles est gérée de manière transparente via la matrice de covariance. Le nombre des amers étant limité, les mises en correspondance sont souvent moins coûteuses

que dans une grille d'occupation par exemple. En revanche, les représentations par amers sont généralement peu denses et permettent seulement la cartographie et la localisation sans planification et sans représentation des obstacles à l'inverse des grilles d'occupation. En effet, les balises ponctuelles informent peu sur la position des grands obstacles continus et les balises de type segment ne sont généralement pas jointifs : le contour des obstacles est rarement continu et fermé, ce qui génère des ambiguïtés (par exemple, on peut se demander si l'on a observé une ouverture dans l'interstice entre deux murs non jointifs).

2.3.5. Cartes Hybrides

L'idée des cartes hybrides est d'utiliser deux ou plusieurs types de représentation ensemble, ce qui va permettre de profiter des avantages de chaque représentation, et pourrait aider à surmonter les limites. Par exemple, plusieurs auteurs ont proposé des méthodes hybrides métriques-topologiques dans l'intention de combiner la précision des cartes métriques avec l'extensibilité des cartes topologiques [Cho97, Duc00, Bos03, Gas99, Sim98, Thr98, Tom01]. Un autre exemple, comme l'indique Bailey [Bai02], la topologie permet de subdiviser l'environnement de manière naturelle, elle induit un espace de stockage et des temps de calcul réduits et assure une connexité ainsi qu'une cohérence à grande échelle. En contrepartie, les cartes métriques induisent une précision locale élevée, une quantification de l'incertitude et une certaine généricité. Quant à la mise en correspondance des données observées avec la carte, elle se fait plutôt par reconnaissance de caractéristiques locales dans le cadre topologique, par opposition à une association de données contrainte par estimation de la pose du robot dans le cas métrique. Ainsi, on trouve dans la littérature un grand nombre de modèles hybrides qui proposent des combinaisons plus ou moins étroites entre informations métriques et topologiques. De plus, comme le souligne Thrun [Thr02], la distinction entre cartes métriques et cartes topologiques n'est pas nette car la grande majorité des systèmes topologiques robustes recourent également aux informations métriques.

2.4. Approche retenue pour la navigation

Notre approche de navigation est caractérisée par l'utilisation d'un capteur stéréoscopique pour construire un modèle 3D de l'environnement. A partir de ce modèle 3D, sera construite une carte d'occupation de l'environnement. La méthode de modélisation 3D de l'environnement est fondée sur la technique des grilles d'occupation. La carte 2D de navigation sera obtenue à partir de la projection du modèle

tridimensionnel de l'environnement. Nous expliquons dans ce qui suit les choix du capteur stéréoscopique et des grilles d'occupation pour construire la carte de l'environnement.

2.4.1. Choix de la stéréovision

La vision par ordinateur vise à représenter la structure tridimensionnelle de l'espace. La stéréoscopie binoculaire utilise deux images prises avec deux caméras dont les angles de vue sont légèrement différents. Un modèle géométrique dit sténopé (*Pin Hole*) est considéré pour le dispositif stéréoscopique. Une étape de calibrage permet de trouver les différents paramètres caractéristiques d'un banc stéréo. Ainsi, le calibrage permet de trouver le modèle de projection de chaque caméra et la relation spatiale entre elles. Cette connaissance permet de calculer les coordonnées 3D d'un point à partir de ses deux projections dans les deux images par une simple triangulation. Dans notre modèle de représentation 3D nous utilisons la stéréovision pour plusieurs raisons. Tout d'abord, parce qu'il s'agit d'un outil de perception peu cher et disponible. De plus, les informations stéréoscopiques sont riches et redondantes ce qui donne une possibilité de précision. Ensuite, la stéréovision a déjà fait ses preuves en temps réel. Enfin, avec les informations de profondeur données par la stéréovision et les informations photométriques (intensité, couleur, texture, etc.) données par l'image, la stéréovision est, en principe, un capteur idéal pour la modélisation de l'environnement.

2.4.2. Choix des grilles d'occupation

Un robot ne possédant a priori aucune information sur l'environnement, dans lequel il doit se déplacer et agir, doit être capable de construire une carte représentant son environnement grâce à l'ensemble de ses capteurs. Cette carte est indispensable pour sa navigation. Dans certaines applications robotiques, une carte de l'environnement peut être fournie par des sources extérieures, mais dans le plupart des cas, ces données sont insuffisantes ou inexistantes pour les applications qui ont besoin de perception précise dans une zone d'activité locale intérieure du robot. D'autre part, un robot autonome doit être capable de réagir à des changements inattendus dans son environnement. C'est le cas pour notre robot mobile qui n'a aucune information sur son environnement et qui requiert une carte pour la navigation. La construction d'une carte, qui répond à ces besoins, devra être incrémentale, en utilisant toutes les informations perceptuelles successivement acquises par le capteur stéréoscopique du robot au cours de son déplacement. L'approche par grille

d'occupation est la mieux adaptée à ce type de problème. C'est une modélisation de l'environnement qui décompose l'espace dans lequel un robot mobile évolue en plusieurs cellules de tailles égales. L'état d'occupation de chaque cellule est une valeur statistique calculée à partir des mesures fournies par les capteurs. Contrairement à beaucoup d'approches qui traitent de la navigation et qui supposent que les environnements d'un robot mobile sont statiques, hypothèse peu compatible avec des robots de services interagissant avec des humains et d'autres robots, la grille d'occupation est une représentation qui supporte la mise à jour de l'environnement, et permet de réviser facilement les états d'occupation, donc de suivre le changement de l'environnement autour du robot, ce qui permet une meilleure réactivité. En plus, une grille d'occupation est capable de représenter des espaces de travail de forme quelconque, et ne cherche pas une approximation des données par des modèles exactes et des formes géométriques qui peuvent être inadéquates. Elle est en général adoptée pour les applications qui reposent sur la détection de l'espace libre [Kwo97] ou pour l'évitement d'obstacles [Borg02]. Dans notre approche, une grille d'occupation floue est utilisée pour modéliser l'environnement à partir des informations de profondeurs issues du système de perception stéréoscopique. Une carte de l'environnement est ensuite générée à partir de ce modèle.

2.5. Conclusion

Dans ce chapitre nous avons d'abord présenté l'état des travaux les plus réputés dans le domaine de la navigation visuelle en faisant la distinction entre les travaux concernant les environnements d'intérieur et les environnements d'extérieur. Ensuite, nous avons mis l'accent sur l'état de l'art des différentes méthodes de représentation des environnements d'intérieur. Nos travaux de recherche se placent donc dans le thème de la navigation visuelle des robots dans des environnements d'intérieur. Dans ce cadre, nous étudions les méthodologies de perception des environnements nous permettant d'obtenir ensuite une reconstruction d'un modèle de l'environnement et une carte de navigation.

Dans le prochain chapitre, nous allons décrire la première étape de l'approche proposée dans cette thèse concernant la perception de l'environnement en utilisant la vision stéréoscopique.

Chapitre 3

Perception de l'environnement par vision stéréoscopique

3.1. Introduction

Le système de navigation humain est un système de navigation visuel tridimensionnel. L'image en deux dimensions n'est qu'une représentation du monde réel tridimensionnel. Le cerveau humain part d'une information d'intensité lumineuse fournie par deux images, une de chacun des deux yeux. Ensuite, il transforme cette information en une représentation sur laquelle on puisse raisonner. La première étape d'un processus d'analyse d'images va consister à structurer l'information contenue dans les pixels de l'image afin d'éliminer d'une part, l'information non utile à la tâche de vision et, d'autre part, d'extraire et de représenter l'information nécessaire à la poursuite du processus d'analyse. Dans le cas de la vision pour la navigation robotique, l'information utile serait de modéliser les surfaces planes ainsi que la forme, les dimensions et les positions des obstacles.

Plusieurs auteurs ont exprimé leur conviction que les systèmes de vision robotiques doivent reproduire le système de vision humain. Cette méthode, fidèle à la vision humaine, est la vision stéréoscopique ou stéréovision. La stéréovision, vision stéréoscopique ou stéréoscopie, est l'ensemble de tous les procédés qui permettent de déterminer les informations de profondeur d'une scène à partir de plusieurs images (deux images ou plus) vidéo de celle-ci, prises sous des angles de vue différents. Le système stéréoscopique le plus utilisé et le plus simple est le système stéréoscopique binoculaire qui n'utilise que deux images. Cependant, certains auteurs recommandent vivement l'utilisation de trois caméras pour s'abstraire de certaines ambiguïtés en cas d'occultations. Dans ce cas, on parle alors de la stéréovision trinoculaire.

Similaire à la vision chez l'Homme, la perception du relief par stéréoscopie binoculaire est assurée principalement par l'exploitation du décalage existant entre les paires d'images stéréoscopiques. Ce décalage entre les mêmes points physiques d'une paire d'images, appelé *disparité*, ne peut être déterminé qu'en effectuant des

correspondances entre les deux images. De ce contexte, surgit le problème central de la vision stéréoscopique qui est la mise en correspondance quoique dans la littérature, des méthodes de stéréovision sans mise en corréspondance existent [Eyn10]. Plusieurs méthodes de mise en correspondance ont été proposées soit pour satisfaire des besoins d'efficacité, soit pour améliorer la précision de la solution.

Ce chapitre est consacré dans sa première partie à décrire les principes fondamentaux de la vision stéréoscopique ainsi que les méthodes existantes. La deuxième partie sera dévouée à l'illustration de notre approche de mise en correspondance pixel à pixel.

3.2. Vision stéréoscopique

À cause de l'aspect projectif de la création des images, il est impossible de recréer la géométrie tridimensionnelle d'une scène à partir d'une seule image [Xie89]. La stéréovision, ou la perception d'une scène 3D à partir, de deux images 2D ou plusieurs, permet de reconstruire un relief apparent. Dans la plupart des cas, seulement deux images sont utilisées. Les travaux de Ito [Ito86] et Ayache [Aya89] ont porté sur l'utilisation de la stéréoscopie trinoculaire. La redondance des données apportée par la troisième caméra, est utilisée, pour contrôler les résultats des disparités calculée à partir de la première paire d'images, ainsi que pour lever quelques ambiguités. Des méthodes de stéréovision récentes donnent des résultats avec une exactitude de l'ordre du sub-pixel [Don11]. La stéréovision est appliquée dans des domaines variés : construction de carte et reconnaissance aérienne, navigation pour la robotique, reconstruction d'objets, modélisation d'environnements, reconstruction volumétrique des scènes, etc.

3.2.1. Principe de base

Soit le système stéréoscopique représenté dans la Figure 3.1. Soit p un point de la scène, p_1 et p_2 sont ces deux projections sur les images R_1 et R_2. La connaissance de la géométrie relative des deux capteurs permet de reconstruire la position tridimensionnelle du point p à partir de la connaissance de ses deux projections. Pour faire cette reconstruction, il est nécessaire d'être capable d'apparier les deux projections.

3.2.2. Mise en correspondance stéréoscopique

La stéréo correspondance ou la détermination des pixels correspondants est le problème principal de la stéréovision. C'est la tâche la plus importante et la

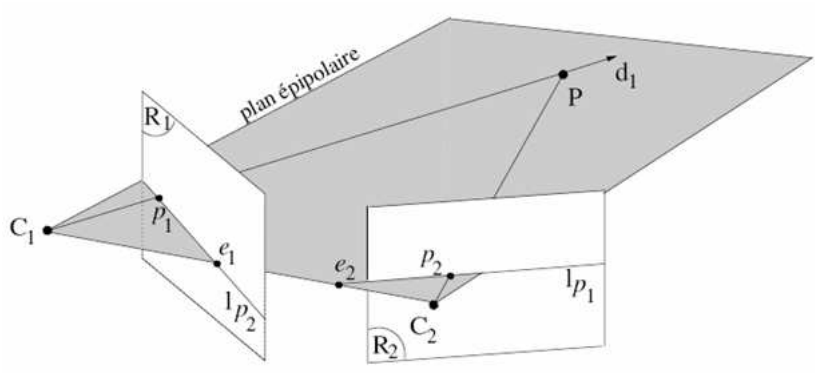

FIGURE 3.1. Géométrie épipolaire

plus coûteuse en terme de temps. Le problème se rapporte à calculer la différence entre les vecteurs formés respectivement par les origines des deux images gauche (l'image de référence) et droite et un pixel de l'image de référence et son correspondant dans l'autre image ($\overrightarrow{o1p1}$ dans l'image $R1$ et $\overrightarrow{o2p2}$ dans l'image $R2$ de la Figure 3.1). Cette différence entre les deux pixels correspondants est appelée disparité. Ce qui ramène le problème de stéréo correspondance à un calcul des disparités pour des pixels de l'image de référence. Le résultat est une carte de disparités dense ou éparse suivant que tous les pixels ou seulement quelques uns (pixels caractéristiques) ont été appariés.

Dans le calcul des correspondances plusieurs problèmes peuvent mener à des faux appariements. Dans ce qui suit nous donnons un court sommaire de ces problèmes :

- Les occultations : Les images sont prises à partir de différents angles de vue. Des pixels de l'image de référence ne vont pas avoir de correspondants dans l'image de gauche car ils seront cachés à partir du deuxième angle de vue. Ces pixels seront appelés des pixels occultés et aucune disparité ne devrait être calculée pour eux.
- Les distorsions photométriques : Les surfaces ne sont pas parfaitement lambertiennes, leurs luminosités changent suivant l'angle à partir duquel elles sont vues. Donc les caméras vont acquérir des valeurs différentes du même point de la scène.
- Les distorsions projectives : En raison de la différence des projections perspectives des caméras, un objet est projeté de manières différentes dans les deux images.

Pour réduire les faux appariements et réduire le domaine de recherche des correspondants, des contraintes stéréoscopiques ont été proposées :

- Contrainte de similarité chromatique : La contrainte de similarité chromatique est la contrainte de base dans la mise en correspondance stéréoscopique. Elle spécifie que, si les conditions de prises d'images n'ont pas beaucoup changé et si les angles des prises d'images ne sont pas éloignés, les pixels correspondants doivent avoir des intensités similaires et leurs voisinages doivent être fortement corrélés.
- Contrainte d'ordre [Bak81] : La contrainte d'ordre spécifie que les projections de deux points physiques sur deux images apparaissent dans le même ordre. Dans leur livre [Hor95], Horaud et Monga ont montré que cette contrainte est violée si la scène observée contient des objets ayant des surfaces transparentes fortement inclinées par rapport au plan des images. Dans ce cas l'ordre des projections est inversé.
- Contrainte d'unicité[Marr79] : Si les objets observés sont opaques et si la disparité n'est pas très forte alors un pixel de la première image a au plus un seul pixel correspondant dans la deuxième image.
- Contrainte de continuité [Marr79] : Cette contrainte stipule que la fonction de disparité qui associe à un pixel de l'image de référence la différence avec son correspondant dans la deuxième image est continue dans un voisinage local. Cette contrainte est violée dans les zones de discontinuité qui causent un changement brusque de la disparité.
- Contrainte épipolaire : Les deux projections p_1 et p_2 du point p respectivement sur les deux plans images R_1 et R_2 sont liés par la relation d'épipolarité. Pour comprendre cette relation, plaçons nous dans le cas représenté par la Figure 3.1. Si P a comme projection p_1 dans la première image, il appartient à la demi droite $[c_1 p_1)$. La projection de P sur la deuxième image parcourt aussi une demi droite qui est la projection perspective de $[c_1 p_1)$ sur le plan image R_2 ($l p_1$ sur la Figure 3.1). Cette droite porte le nom de la droite épipolaire. La contrainte d'épipolarité stipule que les correspondants potentiels dans l'image R_2 sont sur la droite épipolaire $l p_1$. Les intersections de la droite $(C_1 C_2)$ avec les plans rétiniens R_1 et R_2 définissent les épipoles e_1 et e_2 des caméras 1 et 2. Les droites épipolaires dans une image s'intersectent à l'épipole. Cette contrainte réduit l'espace de recherche d'un espace bidimensionnel à un espace unidimensionnel. Pour simplifier le problème de stéréo correspondance et pour plus d'efficacité, on utilise la contrainte épipolaire pour la rectification des pairs d'images stéréo. Les images d'entrées sont rectifiées de sorte que les droites épipolaires correspondants soient des lignes horizontales. On aura que des disparités horizontales. La rectification des images stéréosco-

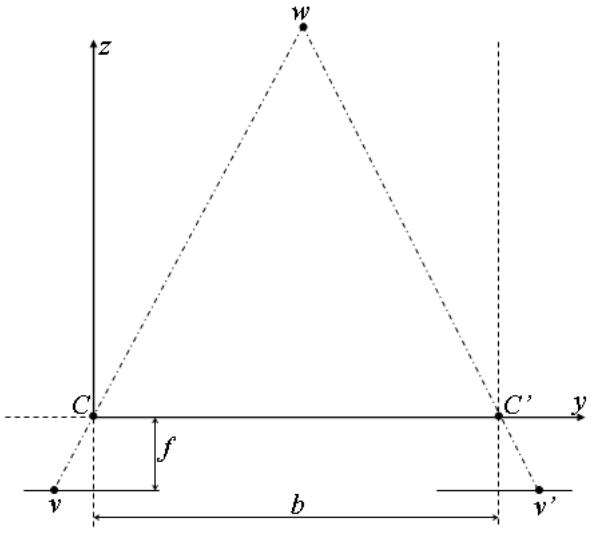

FIGURE 3.2. Reconstruction stéréoscopique

piques consiste à recalculer, pour deux images en positions générales, deux nouvelles images telles que les droites épipolaires sont horizontales, ce qui implique que les deux nouveaux épipoles soient à l'infini. Plusieurs rectifications sont possibles mais la plus simple consiste à garder les deux centres de projection et utiliser comme nouveaux plans rétiniens un seul et même plan contenant la direction de la droite passant par les deux centres de projection. Un plan contenant une droite n'étant pas défini de manière unique, il faut choisir une orientation. Pour l'unicité, on peut mettre la convention que le nouveau plan rétinien contient la direction de la droite intersection des plans rétiniens des images originales. Pour plus de détails sur la rectification d'images le lecteur peut se référer au chapitre 2 de la thèse de Devernay [Deve97].

3.2.3. Reconstruction tridimensionnelle

A l'issue de la phase de mise en correspondance, les couples appariés sont utilisés pour calculer les positions 3D. Ce calcul, appelé triangulation géométrique, est possible grâce aux transformations qui sont établies lors de la phase de calibrage.

Une fois les paires des pixels correspondants sont trouvés, il serait possible de déterminer la distance réelle au point physique si on connaît : les positions mutuelles des deux caméras (paramètres extrinsèques) et les paramètres des capteurs (paramètres intrinsèques). Dans la Figure 3.2 nous avons deux caméras parallèles entre elles, avec des plans rétiniens qui coïncident.

On considère que la disparité est le long des axes des y uniquement. C'est la raison pour laquelle la Figure 3.2 est en $2D$. Si on considère que l'axe de référence est la caméra de gauche on peut déduire les équations 3.1 et 3.2.

$$\begin{cases} \dfrac{-f}{z} = \dfrac{v}{y} \\ \dfrac{-f}{z} = \dfrac{v'}{b-y} \end{cases} \tag{3.1}$$

$$z = \frac{b \times f}{v - v'} \tag{3.2}$$

Donc si on connaît la géométrie du système (b et f dans cet exemple) et la disparité $d = v - v'$, on peut connaître la distance à l'objet en utilisant la dernière équation. Une fois on a la coordonnée z on peut déduire les cordonnées réels x et y en utilisant l'équation 3.3.

$$\begin{cases} x = \dfrac{x_1 \times z}{f} \\ y = \dfrac{y_1 \times z}{f} \end{cases} \tag{3.3}$$

Avec x_1 et y_1 sont les coordonnées correspondants dans le plan de projection. Ces équations ne prennent pas en compte l'incertitude. Dans des conditions réelles ces équations doivent être réécrites pour tenir compte de l'incertitude.

3.3. Techniques de mise en correspondance

Dans cette section nous donnons une brève description des méthodes les plus importantes pour la stéréo correspondance. Nous abordons tout d'abord les éléments de mise en correspondance pour la stéréo correspondance classique basée sur les caractéristiques et la stéréo correspondance surfacique. Ensuite nous illustrons ces méthodes suivant une classification en deux catégories principales : Les méthodes de mise en correspondance locales et les méthodes de mise en correspondance globales.

3.3.1. Éléments de mise en correspondance

En regardant les travaux qui composent la bibliographie de notre état de l'art sur les techniques de stéréo correspondance, on peut distinguer deux approches d'appariements : les méthodes qui apparient des pixels et qui sont donc très proches de la structure échantillonnée des images, et les méthodes qui mettent en corres-

pondance des indices caractéristiques. Un indice caractéristique est un élément de structure de l'image facilement reconnaissable à travers sa signature d'illumination.

3.3.1.1. Mise en correspondance basée sur les indices caractéristiques

C'est la technique classique de stéréo correspondance [Bak81, Oht85, Hsi92, Bol93]. Ces méthodes commencent par extraire des indices caractéristiques des deux images. La recherche des correspondances s'effectue entre ces indices caractéristiques. La première phase de cette technique est l'extraction des caractéristiques des images. Un indice caractéristique est un élément de structure de l'image dont la signature lumineuse présente est peu complexe pour l'appariement ainsi que pour la localisation. D'une part, ce sont des formes faciles à interpréter pour l'œil humain et d'autre part, elles représentent une description physique réelle de la scène contenue dans l'image. Les indices sont généralement classés en deux groupes :

Le premier groupe est composé des indices du type contours. Les méthodes les plus utilisées pour les détecter sont les méthodes dérivatives [Hor95]. Elles sont divisées en trois catégories :

- Les méthodes utilisant le calcul du gradient. Dans un premier temps, ces méthodes déterminent les gradients directionnels de l'image en chaque point. Ensuite, elles calculent la norme du gradient. Enfin, le calcul du maximum local de la norme du gradient permet de définir les points du contour.
- Les méthodes basées sur le calcul des dérivées secondes. Ces méthodes utilisent la norme du gradient pour calculer la dérivée seconde dans la direction du gradient. Les points du contour sont obtenus en cherchant les passages par zéro de la dérivée seconde.
- Les méthodes reposant sur le calcul du Laplacien. Ces méthodes sont basées sur le calcul du Laplacien de l'image lissée et sur la recherche de ses passages par zéro.

Dans les premiers chapitres de [Hor95], les auteurs ont fait un inventaire détaillé de ces méthodes et ont mis l'accent sur des problèmes qui y sont traités comme le calcul des dérivées de l'image ou encore le chaînage des contours. Les méthodes de mise en correspondance basées sur les indices ont montré des limites si les textures présentes dans l'image ne sont pas tout à fait différentiables, et sont vouées à l'échec à cause de la qualité de la segmentation et la complexité de calcul très élevée. D'autre part, ces méthodes remplacent l'image originale par une image binaire contenant uniquement les contours extraits. Puisque les images ne contiennent que des contours, l'identification de droites ou de segments est simplifiée. L'inconvénient des contours est leur imprécision de localisation.

Le deuxième groupe d'indices caractéristiques concerne les points d'intérêts. Les points d'intérêts sont des signes distinctifs bidimensionnels facilement détectables tels que la plus forte courbure, les coins, les jonctions entre les segments, etc. Ces points sont beaucoup moins nombreux que les points de contours mais ils sont plus fiables. Les principaux détecteurs de points d'intérêts sont ceux de Canny [Can86], de Marr et Hildreth [Marr80] ou de Harris [Har88]. Le lecteur pourra trouver d'autres détecteurs dans les documents suivants : [Hor95, Sch96, Der90].

La deuxième phase de cette technique est la mise en correspondance. Une fois les indices caractéristiques obtenus, il faut les mettre en correspondance entre les différentes images. À chaque indice caractéristique on associe un vecteur d'attributs à partir du quel sera calculé l'appariement. Cette technique génère des cartes de disparité éparses qui peuvent être transformées en des cartes denses après une étape d'interpolation.

3.3.1.2. Mise en correspondance basée sur les pixels

Ce sont des méthodes qui cherchent à produire des disparités pour tous les pixels non occultés de l'image de référence[Cham05]. Le critère d'appariement est basé sur une mesure de similarité de la fonction d'illumination locale. Dans les méthodes basées sur les pixels, on tente d'apparier tous les pixels de l'image de référence avec les pixels de la deuxième image pour reproduire fidèlement la structure échantillonnée.

La méthode la plus simple consiste à comparer les intensités des pixels [Cox96]. Mais cette mesure est affectée par les changements ainsi que par les distorsions de l'image et présente des difficultés pour la mise en correspondance. Des techniques plus éfficaces utilisent des zones de comparaison contenant le voisinage du pixel considéré appelées fenêtres de corrélation ou zones de support. Le coût de similarité entre les deux zones de support est obtenue par une mesure statistique, de la distance entre les fonctions d'illuminations sur l'ensemble des pixels de la zone. Les mesures les plus couramment utilisées sont la somme des écarts quadratiques (Sum of Squared Differences ou S.S.D.) [Cox96, Oku01], la somme des écarts absolus (Sum of Absolute Differences ou S.A.D.) [Hir01], l'intercorrélation normalisée (Normalized Cross-Correlation ou N.C.C.) [Che99, Sar02] ou les méthodes de rangs et de recensements [Bha98]. Les méthodes de mise en correspondance pixel par pixel sont très affectées par les occultations, les défauts de réflexions et les changement dans la source de lumière entres les différentes vues. De plus, l'apparition d'artefacts causés par l'échantillonnage de l'image peut affecter la mesure de similarité. L'évaluation d'une mesure de mise en correspondance définie sur une zone de support peut

alors donner une fausse valeur d'appariement à deux pixels correspondants, même dans des images ne contenant pas d'effets perturbants tels que des occultations ou des surfaces non uniformes. Peu d'articles ont portés sur ces problèmes. Dans [Bir98], Birchfield et Tomasi ont proposé une méthode insensible à l'échantillonnage de l'image. La mesure de similarité utilise des fonctions statistiques linéaires basées sur la différence d'intensité entre les zones de support dans les deux images. Les mesures de similarité sont calculées sur des positions entières de pixels et sur des demi-pixels (là où la fonction de l'image est interpolée de façon symétrique dans chaque image). Szeliski et Scharstein [Sze02] ont proposé une méthode se basant sur les travaux de Birchfield et Tomasi [Bir98] mais utilisant une interpolation sur les images en vue d'obtenir des fonctions continues (et ainsi diminuer la sensibilité a l'échantillonnage). A l'inverse de [Bir98], la mesure de mise en correspondance est évaluée sur un espace de disparité appartenant à un seul objet physique où la disparité est contenue. Clerc [Cle98] a proposé de représenter la fonction d'illumination continue de l'image par des valeurs d'ondelettes attribuées à chaque pixel de l'image. De ce fait, les coûts de similarité sont calculés sur ces valeurs. Cette méthode permet de corriger les faux correspondants même dans le cas où le mêmes structures se répètent plusieurs fois avec des taille différentes dans les deux images.

3.3.2. Méthodes globales de mise en correspondance

Une méthode d'appariement est dite globale lorsque la fonction coût est évaluée sur l'ensemble de l'image. L'utilisation de ce type de méthode doit passer par la définition d'un modèle de l'espace apperçu pour régulariser (contraindre) l'ensemble des appariements. Certaines méthodes utilisent la contrainte épipolaire pour calibrer les deux images pour ramener ce problème 2D à un problème 1D [Bel96, Cox92]. Tandis que d'autres méthodes traitent directement du problème 2D[Ish98, Boy98]. La régularisation globale a pour but de réduire la sensibilité des algorithmes aux ambiguïtés provoquées par des occultations, une faible texture locale ou des défauts d'illumination. Le prix à payer pour cette amélioration est un accroissement de la complexité des calculs ainsi qu'une sensibilité du modèle de régularisation.

3.3.2.1. La programmation dynamique

La programmation dynamique est une technique métaheuristique mathématique qui diminue la complexité de calcul d'un problème d'optimisation en le décomposant en plusieurs sous problèmes plus simples. Appliquée au problème de la mise en correspondance de la stéréovision, cette technique cherche à optimiser la

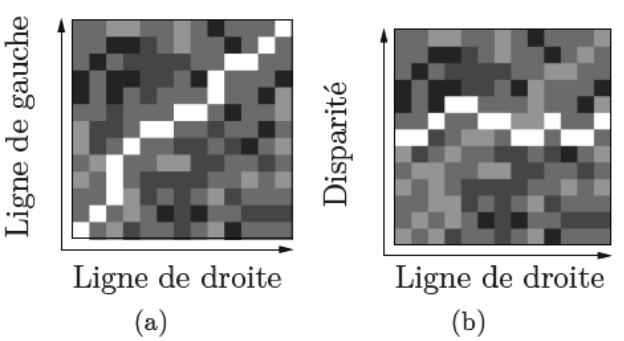

FIGURE 3.3. Représentation de l'espace des disparités en : (a) lignes droite-gauche, (b) ligne droite-disparité. L'intensité représente le coût de la mise en correspondance potentielle le long de la ligne de recherche, un pixel blanc représente un faible coût.

fonction coût d'un chemin dans une matrice dont les éléments sont de tous les appariements possibles. Dans la plupart des travaux, cette approche est utilisée pour une mise en correspondance monodimensionnelle en utilisant la restriction de l'espace de recherche par la contrainte épipolaire. Le problème de la mise en correspondance se réduit à un problème de mise en correspondance des pixels le long d'une ligne de l'image de référence avec ceux de la ligne de l'image de correspondance qui sont épipolaires.

Il est est envisageable d'appliquer les techniques par programmation dynamique uniquement si on suppose que le coût du chemin global est la somme des coûts des chemins partiels obtenus récursivement. Le coût pour chaque point dans l'espace de recherche est défini en utilisant une mesure statistique de mise en correspondance locale (SAD, SSD, etc.). Les occultations dans la technique par programmation dynamique sont détectées si un groupe de pixels dans une image sont assignés à un seul pixel dans l'autre image. Un coût d'occultation est appliqué au coût global du chemin pour modéliser ce type de solutions. On peut voir sur la Figure 3.3 une représentation de la matrice de recherche ainsi qu'une modélisation de l'espace monodimensionnel des disparités. Les axes sont définis par les lignes de recherche des images droite et gauche comme proposé par Ohta et Kanade [Oht85] ou Cox et al. [Cox96].

La formulation de la mise en correspondance par programmation dynamique pose divers problèmes : sont le choix du coût d'une occultation, la difficulté de garder une consistance interligne de recherche [Oht85, Bob99] et le respect des contraintes d'ordre et de continuité. Si on considère une ligne de recherche composée de N pixels, la complexité de calcul en utilisant la programmation dynamique est en $O(N^4)$ à laquelle il faut ajouter le temps requis pour les fonctions

des coûts locaux. Il existe un certain nombre de variations en vue de diminuer la complexité du calcul et de réduire les ambiguïtés de mise en correspondance. Par exemple, Baker et Binford [Bak81] ont proposé de déterminer la disparité ligne par ligne de façon indépendante pour chaque ligne d'une paire stéréoscopiques rectifiées [Fau93a]. Ils cherchent les appariements puis corrigent ceux qui ne vérifient pas la contrainte d'ordre. Ohta et Kanade [Oht85] ont proposé d'introduire à cette méthode des contraintes de continuités verticales entre les différentes lignes épipolaires.

Dans [Cox92], [Cox96, Smi95, Cox92], Cox propose un algorithme de vraisemblance maximum qui tient compte des contraintes stréoscopiques d'ordre et d'unicité. Le coût local dépend de la mesure de correspondance des pixels ainsi que d'une pénalité fixe pour les pixels sans correspondants. Les résultats obtenus dans [Cox92] montrent qu'il existe plusieurs minima globaux de la fonction de coût résultant de l'accumulation de plusieurs hypothèses. Dans [Cox96], Cox présente une version améliorée qui vérifie la continuité dans deux directions différentes de l'espace des disparités, respectant ainsi le principe de continuité physique. Geiger et al. [Gei01] ont mis à profit la constatation qu'une discontinuité de la disparité le long d'une ligne épipolaire est la signature d'une occultation pour définir un nouveau modèle d'occultation. Le coût d'un appariement est évalué sur des zones de support paramétrables capable de s'adapter aux variations d'intensité et de localisation. Belhumeur [Bel96] a développé un modèle basé sur un réseau Bayésien ou les régions occultées sont explicitement représentées et calculées. Dans ce papier, Belhumeur fait la description d'un certain nombre de modèles de scène du plus simple au plus complexe. Il discute sur le fait que les modèles fiables ne contiennent pas seulement des profondeurs simples mais aussi des discontinuités de profondeur localisées sur les contours des objets ainsi que des surfaces bosselées différemment inclinées. Ce modèle est utilisé pour calculer la fonction de disparité optimale par l'estimation du maximum à posteriori. Belhumeur présente aussi une nouvelle stratégie de programmation dynamique qui calcule simultanément la disparité, les régions occultées et les orientations des surfaces. Dans [Bob99], Bobick et al. ont présenté un algorithme qui recherchent les appariements et les régions occultées simultanément, mais sans que les contraintes de surface lisse ou de continuité soient utilisées. Ils ont introduit une structure de donnée appelée "image de l'espace des disparités" (Disparity Space Image ou D.S.I.), dans laquelle on recherche le meilleur chemin par programmation dynamique. La sensibilité aux coûts d'occultation ainsi que la complexité de calcul sont réduites grâce à l'utilisation d'éléments de mise en correspondance fiables qui orientent la recherche de la solution. Cette méthode exploite

la relation entre la largeur de la zone interdite d'une occultation dans le D.S.I. avec l'intensité des bords des objets. Cette méthode permet d'obtenir des résultats relativement fiables malgré la présence de très grandes régions d'occultation. L'un des principaux avantages de la mise en correspondance en utilisant la programmation dynamique est de fournir un support global pour des régions qui sont localement faiblement texturées et qui autrement seraient appariées de façon incorrecte. L'utilisation de cette méthode permet aussi de résoudre les problèmes d'appariement liés aux occultations. Par contre, les pixels, au voisinage d'une occultation, posent problème. En effet, le coût d'une mise en correspondance pour des pixels proches d'une occultation est élevé. Des méthodes pour pallier ces difficultés ont été proposées dans [Bir98]. Ces méthodes remplacent le coût d'une mise en correspondance dans le voisinage d'une discontinuité par un coût d'occultation fixe. Le principal inconvénient des méthodes par programmation dynamique, en plus de la complexité du calcul, est qu'une erreur locale peut être propagée tout le long de la ligne de recherche corrompant ainsi d'autre mises en correspondance potentiellement correctes.

3.3.2.2. Théorie des graphes

Si certaines méthodes utilisant la programmation dynamique tentent d'imposer une certaine régularité dans la fonction de disparité dans toutes les directions, la plupart n'exploitent pas totalement la cohérence bidimensionnelle. Ce défaut vient du fait que la majorité des méthodes utilisant la programmation dynamique apparient les pixels appartenant à la même ligne épipolaire sur les deux images sans prendre en compte une éventuelle continuité de l'image tridimensionnelle à reconstruire. La théorie des graphes permet de généraliser cette technique en deux dimensions. Ce type de méthode a été proposé par Roy et Cox [Roy98] puis a été formalisé par Veksler dans [Vek99], Kolmogorov et Zabih dans [Kol02a, Kol02b, Kol01]. Plusieurs autres travaux ensuite ont utilisé cette méthode [Par02, Roy99]

Le principe de flot dans un graphe peut être vu comme un problème d'écoulement d'eau dans un réseau de tuyaux. Considérons une source d'eau s de débit infini, un puits t de contenance infinie et un réseau de tuyaux reliant la source au puits. Le flot maximum que l'on peut faire passer dans ce réseau de tuyaux est contraint par le réseau. Certains de ces tuyaux séparent la source du puits et se comportent comme un goulot d'étranglement limitant à eux seuls l'ensemble du flot. C'est l'ensemble de ces tuyaux que l'on nomme "goulot d'étranglement" ou "coupure minimale" et la somme de leur capacité est appelée "capacité minimale". Si la capacité d'un de ces tuyaux augmente, la capacité minimale augmente et donc

le flot maximum augmente. La valeur du flot maximum est égale à la coupure minimale [For62]. Donc trouver l'ensemble des tuyaux réalisant le goulot d'étranglement minimum est un problème analogue à celui consistant à trouver la valeur du flot maximum.

Lorsqu'il s'agit d'utiliser ce type de méthode pour réaliser une mise en correspondance de pixels, le graphe représente un réseau reliant une source et un puits par l'intermédiaire de tous les pixels d'une des images stéréoscopiques. Chaque "tuyau" est segmenté en autant de tronçons que de disparités envisageables. La capacité de chaque tronçon représente une fonction de coût d'attribution de la disparité envisagée au pixel auquel le tronçon est relié. Chacun des tronçons est relié aux tronçons voisins par des tuyaux dont la capacité d'écoulement permet de contraindre l'ensemble du réseau.

Boykov et al. [Boy98, Boy99], dans leur première approche, ont basé la fonction d'énergie sur un champ aléatoire de Markov (M.R.F.) qui est capable de préserver les discontinuités. Le coût associé à chaque arc correspond aux termes de la fonction d'énergie. Dans le cas général de la théorie des graphes, le but est de trouver une coupe minimale. Due a la grande complexité de cette recherche, Boykov et al. ont proposé un algorithme qui cherche seulement une approximation de cette solution. Kolmogorov [Kol01] a poursuivi les travaux de Boykov en améliorant la fonction d'énergie dans laquelle les occultations étaient explicitement représentées. Buehler et al. [Bue02] ont repris la même approche basée sur la théorie des graphes pour traiter d'un cas de mise en correspondance avec un système trinoculaire.

Les méthodes basées sur la théorie des graphes ont deux limitations. La première est dûe à la contrainte de continuité, c'est qu'elles ont tendance à construire des cartes de disparité qui ont toujours le défaut d'aplatir les objets. La deuxième est que, comme la fonction de pénalité entre deux pixels voisins de disparités différentes n'est pas nécessairement convexe, la minimisation de l'énergie est un problème NP-complet [Kol01] dont on obtient finalement qu'une approximation. Ishikawa [Ish98] propose l'étude du cas ou cette fonction est convexe et décrit un graphe qui permet d'obtenir le résultat exact.

3.3.3. Les méthodes locales de mise en correspondance

Les méthodes locales ont été supplantées par les méthodes globales au début des années 1990. Moins de dix ans plus tard, les recherches se sont de nouveau concentrées sur le concept local en raison principalement des problèmes liés aux méthodes globales qui sont la complexité de calcul élevée ainsi que la définition d'un modèle précis de la scène observée et l'intégration des contraintes stéréoscopiques. Il est ap-

paru qu'il n'est pas nécessaire d'effectuer une optimisation globale pour formuler correctement le problème de la mise en correspondance et ainsi obtenir l'information de disparité mais qu'au contraire les informations locales de l'image étaient suffisantes.

Parmi les méthodes locales on trouve les méthodes basées sur le gradient ou les approches différentielles réalisant une estimation de mouvement en mettant en relation les dérivés spatiales et temporelles des images. Cette méthode qui a été initiée par Horn et Shunck [Big96] repose sur une propriété de conservation de l'illumination des points entre les différentes images. Dans le cadre de la mise en correspondance stéréoscopique, les méthodes basées sur le gradient ou flot optique [Luc81], [Glu92] cherchent à déterminer de petites disparités locales entre deux images en formulant une équation différentielle reliant le mouvement entre les deux images et l'intensité lumineuse. L'un des principaux inconvénients de l'approche différentielle est qu'elle nécessite une image suffisamment texturée sinon les gradients spatiaux et temporels tendent vers zéro. Cependant, l'estimation des gradients s'effectue au travers de filtres lisseurs (passe-bas) qui requièrent une faible texture de l'image pour être valide. Si la texture est trop importante, on aura de fortes variations d'intensité et la valeur estimée de la dérivée sera fausse du fait du lissage. Notons aussi que l'hypothèse d'illumination globale constante entre les deux images est fréquemment violée. De plus, cette approche utilise l'hypothèse que l'ensemble des points, utilisés pour lisser le champ de vitesse, appartiennent à un même objet. Cette hypothèse est fausse dans le cas de discontinuités du mouvement, c'est à dire à la frontière entre les différents objets. Enfin, cette méthode estime de faibles mouvements et donc de faibles disparités. Cet algorithme est donc mis en défaut pour des scènes comportant de fortes disparités. Une autre méthode locale est la méthode de mise en correspondance des indices. Cette méthode cherche à mettre en correspondance des éléments remarquables dans les images (coins [Big96], lignes, courbes [Sch98], etc.). Dhond et Aggarwal [Dho89] ont recensé un grand nombre de méthodes basées sur la mise en correspondance d'indices. Dans [Vin01], Vincent et Laganiere ont comparé les performances de différents algorithmes de mise en correspondance d'indices. Les cartes de disparités obtenues par la méthodes de mise en correspondance des indices sont éparses et ne sont qu'une approximation simpliste de la fonction de disparité réelle en raison principalement du nombre limité de régions segmentées et du faible nombre de surfaces identifiées. Néanmoins, un certain nombre de méthodes [Tom96, Lhu02] se base sur le principe des méthodes de mise en correspondance d'indices comme étape initiale à la mise en correspondance dense.

Enfin on trouve les techniques corrélatives. Dans ces méthodes, la mise en correspondance est basée sur l'appariement d'éléments les plus semblables. A chaque pixel à mettre en correspondance est assignée une liste de mesures caractéristiques. Une distance statistique, basée sur ces caractéristiques appelée aussi coût de correspondance, évalue leur similarité en vue de sélectionner des éléments appariés. Le coût algorithmique de la caractérisation d'un élément dépend uniquement des propriétés locales de l'image (avec un voisinage prédéfini). Le principe de ces méthodes repose alors sur la définition d'une fenêtre appropriée appelée zone de support contenant l'élément à mettre en correspondance ainsi que son voisinage, sur le choix des caractéristiques discriminantes et une mesure de similarité calculée entre ces éléments. Chaque position de la fenêtre génère une signature particulière et on retient au final la position de la fenêtre obtenant le meilleur score d'appariement [Bob99, Gei01][Oku01]. Dans la section suivante, nous nous intéressons à la technique de mise en correspondance corrélative.

3.4. Méthodes locales de mise en correspondance : mise en correspondance corrélative

Les techniques corrélatives sont employées généralement pour un appariement pixel à pixel. Nous avons choisi d'étudier ce type de technique car il produit une carte de disparité dense qui permet une reconstruction quasi totale de la scène observée par le robot. Dans cette technique, la recherche des appariements est basée sur un critère de ressemblance photométrique. On considère une fenêtre, appelée aussi *zone de support*, autour du point que l'on veut apparier, puis on cherche dans l'autre image le point dont la fenêtre, de même taille que la précédente, est la mieux corrélée (voir Figure 3.4). Le choix de la taille et la forme de la zone de support est très important dans l'algorithme de mise en correspondance. Un grand nombre de distances statistiques ont été proposées pour calculer la corrélation entre deux zones de support. La contrainte épipolaire est d'abord utilisée afin de réduire le nombre de pixels candidats à l'appariement. Certaines contraintes globales (unicité, ordre, continuité de la disparité) sont ensuite appliquées pour lever certaines ambiguïtés. Les méthodes corrélatives sont simples à mettre en œuvre. Cependant, elles présentent deux limitations principales :

– Elles sont sensibles aux distorsions de perspective. Un effet de perspective peut influencer les coefficients de corrélation soit par la différence de luminosité apparente des surfaces suivant l'angle de prise de vue, soit par le fait

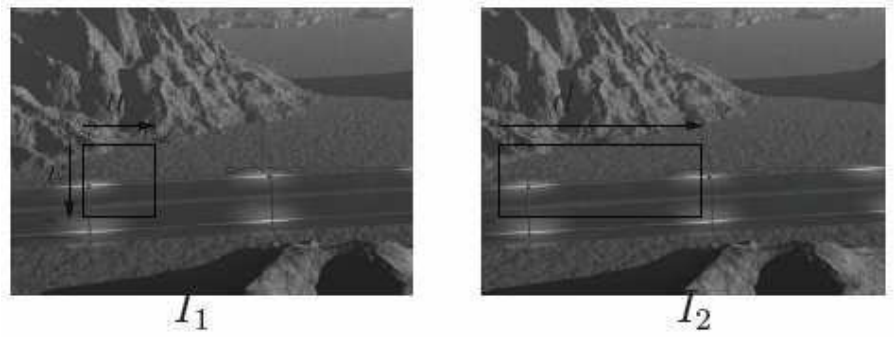

FIGURE 3.4. Mise en correspondance par corrélation

qu'on corrèle des zones de support de même taille alors qu'en réalité deux zones qui se correspondent n'ont pas les mêmes tailles.

– Elles ne permettent pas de détecter les occultations. Une méthode corrélative fournit toujours, pour chaque point, un correspondant qui présente la meilleure corrélation, même si elle est faible. Or, lors d'une occultation, un point d'une image peut ne pas avoir de correspondant dans l'autre image. L'utilisation d'un seuil pour ne retenir que les coefficients de corrélation suffisamment élevés peut conduire à éliminer en même temps certains bons appariements.

3.4.1. Le choix de la zone de support

Le choix de la taille et de la forme de la zone de support est très important pour produire des bons appariements. La zone doit être assez large pour contenir suffisamment d'éléments afin de rendre l'appariement fiable, la fiabilité étant liée au nombre de pixels utilisés pour évaluer l'indice d'appariement. A contrario, la zone doit être suffisamment petite pour éviter les problèmes liés aux occultations et aux fortes variations de disparité. Enfin, la zone doit fournir des caractéristiques suffisamment discriminantes pour garantir un appariement unique. Dans une démarche consistant à mettre en correspondance des pixels, l'approche classique consiste à définir un voisinage rectangulaire de taille fixe. Ce choix de taille fixe présente de nombreux inconvénients particulièrement s'il existe de grandes variations de profondeur dans la scène analysée : une taille de fenêtre réalisant un bon compromis entre fiabilité et risque sur une zone de l'image correspondant à des objets proches ne conviendra pas dans une zone où la scène perçue est très éloignée. La différence des points de vue entre les caméras peut engendrer des motifs différents même pour une zone correspondante.

Pour pallier ces inconvénients, il a été envisagé d'adapter la taille de la fenêtre aux caractéristiques détectées dans la fenêtre. Dans [Kan94], Kanade et Okutomi augmentent la taille des fenêtres dans les zones de l'image où la surface est peu texturée afin d'augmenter le pouvoir discriminant de la zone de recherche. Certains auteurs [Tao01, Zha02] ont proposé de segmenter l'image afin d'être sûr que la fenêtre de recherche ne contienne pas d'élément extérieur a l'objet étudié et donc éviter ainsi les problèmes de variation de disparité liés aux bords d'objets. Dans [Sze02], Szeliski et Scharstein modélisent les fenêtres de recherche dans l'espace des disparités afin de rendre leur algorithme insensible aux problèmes géométriques liés aux positions des différentes caméras.

3.4.2. Mesure de similarité

Le choix de la mesure de similarité, qui renseigne sur le degré de ressemblance entre les deux zones de support des deux images, influe sur la qualité du résultat. Une bonne mesure permet de discriminer nettement le bon appariement parmi tous les pixels candidats et doit être robuste aux différentes formes de bruit. Certaines mesures sont basées directement sur la corrélation chromatique entre les pixels correspondants, d'autres utilisent des statistiques de moment ou d'ordre. Dans [Asc93], Aschwanden et Guggenbuhl ont comparé ces différentes mesures.

Le principe des distances basées sur les moments est le suivant : il s'agit de faire la somme des différences (quadratique ou absolue) pixel a pixel des deux fenêtres (S.A.D. ou S.S.D.)[Gha10]. L'inconvénient de ces deux distances est qu'elles sont sensibles aux variations de l'illumination globale entre les deux images. C'est pourquoi on leur préfère généralement leurs versions centrées (Z.S.A.D. ou Z.S.S.D.) qui sont des distances invariantes aux translations uniformes de l'illumination globale. La mesure de distance statistique normalisée (N.Z.S.S.D.) permet de régulariser toutes les différences pixel a pixel au sein d'une même zone. L'intercorrélation normalisée (N.C.C.) est basée sur le même principe que les méthodes basées sur les moments mais effectue un calcul de corrélation entre les deux zones de support. Cette méthode statistique est celle qui est la plus couramment employée pour déterminer une ressemblance.

Zabih et Woodfill [Zab94] ont proposé une méthode alternative utilisant les statistiques de rang pour assurer la mise en correspondance. Les statistiques de rang font partie du groupe des statistiques robustes. Les auteurs attendent donc de ce type de méthode une réduction de la sensibilité de la mise en correspondance vis à vis du modèle de bruit utilisé. Le calcul de la distance statistique qu'ils proposent se déroule en deux temps : tout d'abord, ils appliquent des transformations non

paramétriques locales aux images. Cette transformation de rang est appliquée aux régions (le motif de référence et la zone de recherche) dans les deux images. La transformation de rang pour une petite région autour d'un pixel est définie comme le nombre de pixels composant la région pour lesquels l'intensité est plus petite que le pixel central. Les valeurs résultantes sont basées sur le nombre de pixels ayant un niveau de gris inférieur à un seuil plutôt que sur les niveaux de gris eux-mêmes. Ils ne retiennent au final que le nombre de pixels dont l'intensité est inférieure sans retenir leur localisation. Ensuite, après que la transformation de rang ait été appliquée, ils calculent la mise en correspondance de motif sur ces nouvelles valeurs en utilisant les méthodes classiques telles que la somme des écarts absolus.

Les statistiques de rang, si elles sont plus robustes, sont moins précises que les statistiques de moment. Dans le cas de la méthode de Zabih et Woodfill, la transformation non-linéaire permettant de passer d'un motif en niveaux de gris à deux proportions fait perdre beaucoup d'information. Pour pallier ce problème, les auteurs ont proposé une autre méthode sous le nom de "recensement" [Zab94] permettant de conserver l'information d'ordre relatif des pixels. Le principe de cette méthode est le suivant : ils comparent toujours le niveau de gris des pixels entourant le pixel central mais ils codent le résultat dans une chaîne booléenne permettant ainsi de conserver la distribution spatiale des rangs.

La mise en correspondance est alors calculée en utilisant la distance de Hamming entre les chaînes booléennes. Cette transformation augmente la dimension des données image par un facteur lié à la taille de la région locale, ce qui augmente le temps de calcul d'autant. Dans leur article de référence, Banks et Corke [Ban01] comparent les performances des méthodes basées sur les transformations de rang et de recensement par rapport aux méthodes basées sur les mesures de corrélation et de différences. Leurs conclusions indiquent que les méthodes basées sur des statistiques de rang offrent des performances comparables et sont plus robustes aux variations de luminosité ainsi qu'aux occultations. Les méthodes de mise en correspondance de motifs présentent l'avantage d'une grande simplicité d'implémentation logicielle ou matérielle [Fau93b, Lep99]. De plus, elles ont fait l'objet de nombreuses études [Bha98, Zab94] et leurs performances sont bien connues [Asc93].

Le principal inconvénient est que ces méthodes sont employées dans le cas de mise en correspondance d'images stéréoscopiques redressées. En effet, la discrétisation de l'espace des mouvements entre les deux images est implicite lorsqu'il s'agit de translation entre les deux images.

En revanche, elle ne l'est pas lorsque la différence de point de vue entre les images comporte en plus des rotations, et donc des déplacements non entiers des

pixels. Comment alors faire la différence pixel a pixel entre deux motifs ? L'autre inconvénient est que si le motif de référence disparaît complètement ou partiellement dans la zone de recherche suite à une occultation, le processus de mise en correspondance fournit quand même un résultat d'appariement. La mise en correspondance retenue sera celle la plus probable au sens de la distance utilisée mais ne représentera pas obligatoirement l'appariement correct. Il est donc nécessaire de fixer arbitrairement un seuil au-delà duquel la mise en correspondance est rejetée car trop peu fiable.

3.5. Proposition de distributions de possibilités pour la mise en correspondance pixel à pixel

La majeure partie des méthodes classiques de mise en correspondance utilisent une représentation statistique de l'erreur d'appariement. Les techniques de régularisation s'appuient sur l'hypothèse d'ergodicité du bruit de mise en correspondance et utilisent le voisinage de chaque pixel pour évaluer une représentation de la statistique des variations de niveaux de gris. Plus le voisinage utilisé est étendu, plus la statistique sera précise, mais moins elle sera fiable. Le manque de fiabilité étant lié aux régions ambiguës (discontinuité, occultations, ...). Dans cette section, nous définissons des distributions de possibilités qui utilisent une formulation sémantique de la contrainte chromatique et des contraintes géométriques d'unicité et d'ordre afin d'augmenter la robustesse de la mise en correspondance vis à vis des ambiguïtés d'appariements.

Nous proposons deux mesures de possibilité qui modélisent les contraintes stéréoscopiques en utilisant des sous ensembles flous. Pour chaque couple de pixels des deux images à apparier, une valeur de confiance est définie à partir des deux mesures de possibilité. La valeur de confiance exprime à quelle mesure l'appariement des deux pixels en question est fiable.

3.5.1. Espace 3D de disparité

Sans perdre de généralité, nous supposons que les paires stéréoscopiques ont été redressées pour avoir des disparités, seulement, le long de l'axe des Y. On définit un espace de disparité $3D$ dont les dimensions sont r, c et d pour désigner respectivement les lignes les colonnes et les disparités. Chaque élément (r, c, d) de l'espace de disparité est projeté au pixel (r, c) sur l'image de référence et au pixel $(r, c + d)$ sur l'image de droite. L'élément (r, c, d) fait référence à l'appariement du

pixel (r, c) de l'image de référence et du pixel $(r, c + d)$ de l'image de droite. On définit dans l'espace de disparité, une fonction L à valeurs réelles. $L(r, c, d)$ est le coût de l'appariement (r, c, d).

3.5.2. Possibilité de mise en correspondance

La contrainte chromatique stipule que les projections d'un même point physique ont des intensités d'illumination comparables. La majeure partie des approches de mise en correspondance stéréoscopique utilise cette contrainte dans une mesure de similarité statistique qui calcule la différence d'illumination entre deux zones de la paire stéréoscopique. Cette mesure, basée uniquement sur la distance numérique entres les intensités, est particulièrement perturbée par les changements causés par des phénomènes non-ergodiques comme le changement de point de vue, les occultations partielles, l'échantillonnage ou la numérisation. Ces changements ne peuvent pas être modélisés par des simples lois normales. Dans notre approche, nous proposons de modéliser la contrainte de similarité par une mesure floue plus robuste vis à vis du bruit et des changements. Nous avons défini une classification des pixels suivant leur niveau de gris. Trois classes ont été définies ; "*black*" pour la classe des pixels noirs, "*white*" pour la classe des pixels blancs et "*average*" pour les pixels dont le niveau de gris se situe entre le blanc et le noir. Les fonctions d'appartenance à ces classes sont des Gaussiennes centrées en 0, 127,5 et 255 (3.4).

$$\mu_{class}(m) = e^{-\frac{(I(m)-c_{class})^2}{2\sigma^2_{class}}} \tag{3.4}$$

$I(m)$ est l'intensité du pixel m, c_{class} et σ_{class} sont respectivement le centre et la déviation standard de la classe en considération. En se basant sur cette classification nous établissons la proposition suivante.

Proposition. *L'appariement entre le pixel m_1 et m_2, appartenant chacun à l'une des images stéréoscopiques, est "possible"si les deux pixels appartiennent à la même classe de gris. Autrement dit : (m_1 est black ET m_2 est black) OU (m_1 est white ET m_2 est white) OU (m_1 est average ET m_2 est average).*

Definition. Considérant deux pixels m_1 et m_2, appartenant chacun à l'une des images stéréoscopiques, on définit une possibilité de correspondance $\Pi(m_1, m_2)$ entre les deux pixels m_1 et m_2 en exprimant la proposition précédente avec les opérateurs logiques classiques. $\Pi(m_1, m_2)$ est une mesure de co-appartenance à une même classe de niveau de gris. Elle reflète à quel point il est possible d'avoir m_1 et m_2 comme pixels correspondant. l'expression de $\Pi(m_1, m_2)$ est donnée par (3.5).

$$\Pi(m_1, m_2) = \max \begin{pmatrix} \min(\mu_{black}(m_1), \mu_{black}(m_2)), \\ \min(\mu_{average}(m_1), \mu_{average}(m_2)), \\ \min(\mu_{white}(m_1), \mu_{white}(m_2)) \end{pmatrix} \tag{3.5}$$

$\mu_{class}(m)$ est le degré d'appartenance du pixel m à la classe en question. La possibilité de mise en correspondance entre deux pixels est comprise entre 0 et 1. Les valeurs proches de 1 indique une forte possibilité de mise en correspondance.

Notation. Dans tout ce qui suit, nous utilisons la notation $\Pi(r, c, d) = \Pi(m_1, m_2)$ avec $m_1 = (r, c)$ et $m_2 = (r, c + d)$.

3.5.3. Possibilité de non-mise en correspondance

Si on suppose que les objets sont opaques et que la disparité entre les deux images stéréoscopiques n'est pas très significative, la contrainte d'unicité stipule qu'un objet dont la projection est un point dans l'image de gauche, a au plus un seul pixel comme projection sur l'image de droite. L'utilisation de la contrainte d'unicité réduit le nombre de correspondants potentiels d'un pixels de l'image de référence. Elle permet donc de modifier une distribution initiale de correspondance pour avoir une nouvelle distribution qui viole moins la contrainte d'unicité. Dans notre approche, on utilise la contrainte d'unicité pour calculer une distribution de possibilités de non-mise en correspondance en se basant sur la distribution des possibilités de correspondance définie dans le paragraphe précédent.

En se référant à la distribution de possibilité de correspondance, un appariement (r, c, d) viole la contrainte d'unicité s'il y a un appariement (r, c, d') avec $d \neq d'$ et $\Pi(r, c, d) < \Pi(r, c, d')$

Definition. Considérant deux pixels $m_1 = (r, c)$ et $m_2 = (r, c + d)$, appartenant chacun à l'une des images stéréoscopiques, on définit *une possibilité de non-mise en correspondance* $\Pi_U\overline{(r, c, d)}$ relativement à la contrainte d'unicité. $\Pi_U\overline{(r, c, d)}$ reflète à quel point la mise en correspondance des deux pixels $m_1 = (r, c)$ et $m_2 = (r, c + d)$ viole la contrainte d'unicité. L'expression de cette possibilité est donnée par (3.6).

$$\Pi_U\overline{(r, c, d)} = \sup_{d' \neq d} \{\Pi(r, c, d') > \Pi(r, c, d)\} \tag{3.6}$$

Sous certaines conditions définies dans [Fau93a], l'ordre des pixels est préservé à travers les images. Cette contrainte peut être formulée par la proposition suivante :

Proposition. *Considérant deux pixels m_1 et m_2, appartenant respectivement à l'image de référence et l'image de correspondance, Si m_1 et m_2 sont les projections du même point physique M alors tous les pixels à droite (respectivement à gauche) du pixel m_1 ont leurs correspondants à droite (respectivement à gauche) du pixel m_2.*

De la même façon avec laquelle on a utilisé la contrainte d'unicité. Notre but est de déduire une distribution de possibilités de non-mise en correspondance relative à la contrainte d'ordre en partant de la distribution des possibilités de correspondance. Pour cette raison nous exprimons la proposition négative duale de la proposition précédente :

Proposition. *Considérant deux pixels m_1 et m_2, appartenant respectivement à l'image de référence et l'image de correspondance, Si m_1 et m_2 sont les projections du même point physique M alors tous les pixels à droite (respectivement à gauche) du pixel m_1 ne peuvent pas avoir leurs correspondants à gauche (respectivement à droite) du pixel m_2.*

En d'autre termes, un appariement (r, c, d) viole la contrainte d'ordre s'il y'a un appariement (r, c', d') qui vérifie la condition suivante : $(c < c' \; ET \; c + d > c' + d')$ OU $(c > c' \; ET \; c + d < c' + d') \; ET \; (\Pi(r, c, d) < \Pi(r, c', d'))$. En se basant sur cette analyse, nous donnons la définition suivante :

Definition. Considérant deux pixels $m_1 = (r, c)$ et $m_2 = (r, c + d)$, appartenant chacun à l'une des images stéréoscopiques, on définit *une possibilité de non-mise en correspondance* $\Pi_O\overline{(r, c, d)}$ relativement à la contrainte d'ordre. $\Pi_O\overline{(r, c, d)}$ reflète à quel point la mise en correspondance des deux pixels $m_1 = (r, c)$ et $m_2 = (r, c + d)$ viole la contrainte d'ordre. L'expression de cette possibilité est donnée par (3.7).

$$\Pi_O\overline{(r, c, d)} = \max \left(\begin{array}{ll} \sup\limits_{\substack{c'>c \\ d'<d-(c'-c)}} & \{\Pi(r,c',d')>\Pi(r,c,d)\}, \\ \sup\limits_{\substack{c'<c \\ d'>d+(c-c')}} & \{\Pi(r,c',d')>\Pi(r,c,d)\} \end{array} \right) \tag{3.7}$$

Definition. En utilisant les deux distributions des possibilités de non-mise en correspondance nous définissons une distribution de possibilités de non-mise en correspondance globale. La possibilité de non-mise en correspondance globale $\Pi\overline{(r,c,d)}$, dont l'expression est donnée par (3.8), exprime à quel point l'appariement (r,c,d) viole les contraintes stéréoscopiques d'unicité et d'ordre.

$$\Pi\overline{(r,c,d)} = \max(\Pi_U\overline{(r,c,d)}, \Pi_O\overline{(r,c,d)}) \tag{3.8}$$

3.5.4. Confiance de correspondance

En utilisant les distributions des possibilités de mise correspondance et de non-mise en correspondance, nous attribuons à chaque appariement (r,c,d) une *valeur de confiance de correspondance* $\tau(r,c,d)$ dont l'expression est définie en (3.9).

$$\tau(r,c,d) = \frac{\Pi(r,c,d)}{1+\lambda\times\Pi\overline{(r,c,d)}} \tag{3.9}$$

λ est la constante de confiance. La valeur de confiance $\tau(r,c,d)$ exprime dans quelle mesure nous pouvons nous fier à l'appariement des pixels (r,c) et $(r,c+d)$ en considérant les valeurs chromatiques des deux pixels et et les contraintes stéréoscopiques de leur appariement.

3.6. Proposition d'un nouvel algorithme de mise en correspondance

Le but ultime de cette section est d'obtenir une mise en correspondance dense de deux images stéréoscopiques, c'est à dire d'être capable de trouver, pour tout pixel de l'image 1, un pixel correspondant dans l'image 2. La mise en correspondance est l'étape la plus importante et la plus coûteuse de la stéréovision. Les méthodes de mise en correspondance classiques souffrent de faiblesse à cause de deux problèmes principaux. Le premier problème est le choix de la taille de la zone de support. D'une part, une petite taille de la zone de support augmente l'influence du bruit. D'autre part, une grande taille fait de sorte que la zone de support couvre des pixels avec des profondeurs différentes et leur support n'est pas fiable. Pour ces raisons, dans la méthode que nous proposons [Gha10a, Gha11], contrairement aux méthodes qui cherchent à optimiser la taille de la fenêtre de support ou utiliser une

taille adaptative [Hof89, Pan78, Luo08, Boy98], nous essayons d'attribuer des poids de support aux pixels dans la fenêtre de corrélation. La quantité de support, que reçoit le pixel central de la part d'un autre pixel de la fenêtre, doit être importante si les deux pixels sont situés sur le même objet physique (même profondeur). La difficulté dans une approche locale qui adapte les poids de support réside dans l'évaluation de la variance des profondeurs puisque c'est ce que nous avons l'intention de calculer [Lev73]. Le second problème est que les méthodes locales sont uniquement fiables si certains critères sont satisfaits : la source d'illumination doit être un point à l'infini, les surfaces de la scène doivent être parfaitement Lambertienne, la dissimilarité et la distorsion entre les vues sont faibles. Dans plusieurs situations où la stéréovision est appliquée, l'existence des environnements et des sources de lumières idéales n'est pas assumée. Dans ces situations, les métriques de corrélation classiques ne suffisent pas pour avoir des disparités exactes. Dans cette partie, nous utilisons la distribution de confiance de correspondance définie dans la section précédente pour estimer des disparités initiales. Nous utilisons ces disparités pour calculer les poids de support des pixels dans la fenêtre de support. Nous utilisons, ensuite, une fonction d'agrégation S.A.D. sur une zone de support de taille fixe, pondérées avec les poids de support calculés, pour déterminer les disparités finales.

3.6.1. Estimation des disparités initiales

Dans la section précédente, nous avons proposé une distribution de confiance qui définit pour chaque couple de pixels des deux images à apparier une mesure de confiance de correspondance rigide vis-à-vis des imprécisions. Nous utilisons cette distribution des valeurs de confiance comme coûts de correspondance pour estimer les disparités initiales. La disparité avec la valeur de confiance de correspondance maximum, est définie comme la meilleure disparité et est retenue comme disparité initiale.

Notation. Dans ce qui suit, nous utiliserons la notation $d_0(r,c)$ pour la disparité initiale du pixel (r,c), déterminée en se basant sur la distribution des valeurs de confiance des correspondances. La fonction de disparité initiale associe à chaque pixel (r,c) de l'image de référence son correspondant $(r, c + d_0(r,c))$ dans la deuxième image.

3.6.2. Calcul des poids de support

L'ajustement de la taille de la zone de support est crucial. En effet, il faut essayer de trouver le meilleur compromis entre les contraintes de temps de calcul et la qualité des résultats, sachant que la présence d'occultations conduit souvent à de faux appariements. Certains auteurs proposent alors d'utiliser des fenêtres adaptatives [Yoo06, Vek02, Gu08, Yan09, Zha09, Mat10], des fenêtres multiples ou encore des fenêtres prédites.

Notre approche consiste à fixer la taille et la forme de la zone de support avec une modulation de l'importance (le poids) accordée à chaque point de la fenêtre. Le poids d'un pixel de la fenêtre de corrélation dépend de la quantité de support qu'il fournit au pixel central. Nous calculons le poids de support de chaque pixel de la zone de support en se basant sur le principe de cohérence définie par Pradzny [Pra85]. Nous examinons les deux pixels candidats à la correspondance en calculant la quantité de support qu'ils reçoivent de leurs voisinages locaux. Le principe de cohérence stipule, que les disparités de deux pixels, projections de deux points physiques appartenant à un seul objet physique, sont similaires. L'idée est que deux pixels voisins avec des disparités similaires doivent se soutenir, tandis que deux pixels avec des disparités différentes ne doivent pas interagir entre eux. Pour incorporer cette réflexion dans un algorithme de mise en correspondance stéréoscopique, nous avons défini une fonction de poids de support. Cette fonction est donnée par (3.10).

$$w(i,j) = \frac{1}{\mu\sqrt{2\pi}\sqrt{i^2+j^2}} \exp\left(\frac{-\left|d_0(r,c) - d_0(r+i,c+j)\right|}{2\mu^2(i^2+j^2)}\right) \qquad (3.10)$$

$w(i,j)$ est le poids de support que le pixel central de la zone de support (r,c) reçoit du pixel $(r+i,c+j)$. Le poids de support dépend de la distance entre les deux pixels ($\sqrt{i^2+j^2}$) et de la différence de leurs disparités initiales ($d_0(r,c) - d_0(r+i,c+j)$). Les pixels les plus loins du centre dans la région de support et dont les disparité diffèrent significativement de celle du pixel central exercent moins d'influence dans le calcul des disparités finales.

3.6.3. Calcul des disparités finales

La différence absolue des intensités, dont l'expression est donnée par (3.11), est utilisée pour le calcul du coût de correspondance. $I_r(r,c)$ est l'intensité du pixel (r,c) dans l'image de référence et $I_m(r,c+d)$ est l'intensité du pixel décalé d'une disparité d du pixel (r,c) dans l'image de correspondance. Une fenêtre de support de taille $2w$ avec des poids de support tels que décrits dans le paragraphe précédent est utilisée pour calculer le coût d'agrégation $C(r,c,d)$ comme décrit dans (3.12).

$$C_0(r,c,d) = |I_r(r,c) - I_m(r,c+d)| \tag{3.11}$$

$$C(r,c,d) = \frac{1}{4w^2 I_{max}} \sum_{i=-w}^{i=w} \sum_{j=-w}^{j=w} w(i,j) C_0(r+i, c+j+d) \tag{3.12}$$

Le coût final de correspondance est donné par (3.13). Pour l'étape de calcul des disparités finales, la meilleure disparité est sélectionnée en comparant les coûts de toutes les disparités : *le-gagnant-prend-tout* (winner-takes-all) [Bek07].

$$L(r,c,d) = \frac{[\tau(r,c,d)]^{\alpha}}{C(r,c,d)} \tag{3.13}$$

3.6.4. Détection des zones occultées

Dans notre approche, nous détectons les zones occultées explicitement en examinant le coût de correspondance final. La valeur de confiance proposée, utilisée dans le calcul du coût final, permet de discriminer les faux appariements des bons appariements. Nous déterminons si un pixel est occulté ou non en comparant le coût de la meilleure disparité choisie pour ce pixel à un seuil. Si le coût est inférieur au seuil, le pixel en question est considéré comme occulté dans l'image de correspondance.

3.7. Validation expérimentale de l'approche proposée

Nous proposons, dans ce paragraphe, de comparer les résultats obtenus par notre approche avec les résultats obtenus en utilisant les méthodes classiques d'ap-

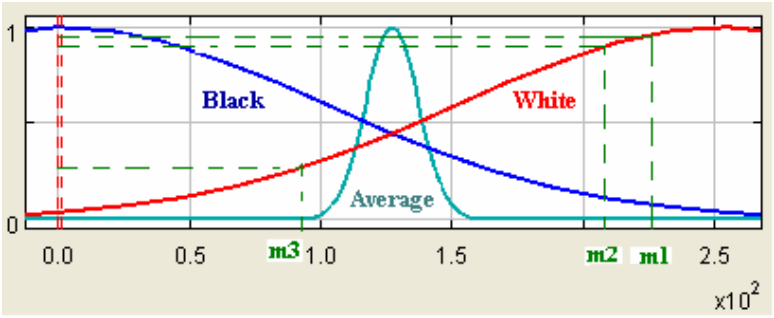

Figure 3.5. rajustement empirique des déviations standards des fonctions d'appartenance.

Figure 3.6. Cartes de disparité initiales trouvées avec (de la gauche vers la droite) $\lambda = 0.5$, $\lambda = 1$ et $\lambda = 1.5$

pariement. Nous avons décidé d'évaluer les appariements en construisant la carte de disparité par ces méthodes.

3.7.1. Fixation des paramètres empiriques de l'algorithme

Pour toutes les expérimentations, nous avons fixé $\sigma_{black} = \sigma_{white} = 7.071, \sigma_{average} = 2.236$ et $\alpha = 0.6$. Ces valeurs ont été déterminées empiriquement. La Figure 3.5 montre les fonctions d'appartenance pour les classes des pixels définies. En utilisant la classification des pixels définie suivant le niveau de gris, nous obtenons des valeurs significatives pour les possibilités de correspondance. La Figure 2 montre que, pour deux pixels m_1 et m_2 qui sont dans le même plage de niveau de gris, la possibilité de correspondance $\Pi(m_1, m_2) = 0.9$ alors que pour m_1 et m_3, $\Pi(m_1, m_2) = 0.26$

La Figure 3.6 montre trois cartes de profondeur pour la paire stéréoscopique de Tsukuba déterminée en utilisant la distribution de confiance de correspondance pour différentes valeurs de la constante de confiance λ. La meilleure carte de profondeur est obtenue pour une valeur de $\lambda = 1$.

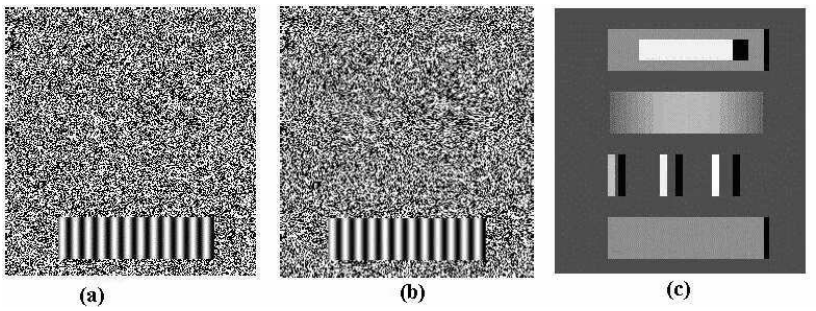

Figure 3.7. Images de synthèse, 50% densité (a) image de référence, (b) image de droite, (c) carte de disparité réelle, Les zones noires sont occultées

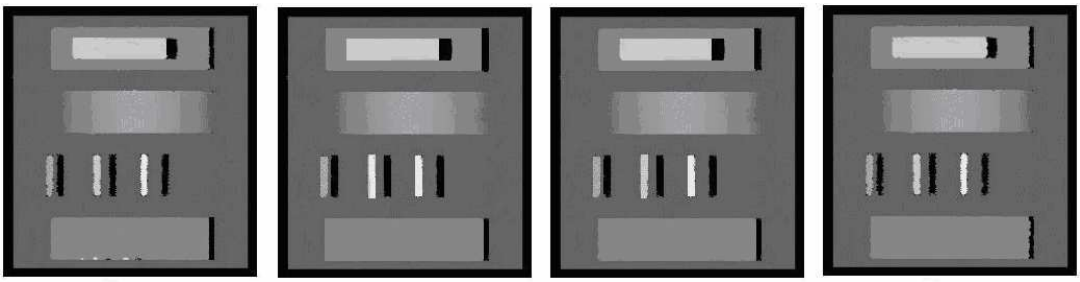

Figure 3.8. Cartes de disparité touvées en utilisant une taille (de la gauche vers la droite) 4×4, 12×12, 18×18, et 24×24 de la fenêtre de corrélation, les zones en noir sont les occultations détectées.

La Figure 3.7 (a) et (b) montrent une paire stéréoscopique d'images de synthèse avec un bruit aléatoire. C'est à dire que les variations d'illumination pour deux pixels correspondant sont faibles et proviennent uniquement de la variation de point de vue. La différence de position entre les deux vues est très faible et la scène contient peu d'occultations. La Figure 3.7 (c) est la carte des disparités théorique. La Figure 3.8 montrent des cartes de disparité générées par notre algorithme en utilisant des zones de support de tailles 4×4, 12×12, 18×18 et 24×24. Les meilleurs cartes sont obtenues avec une taille de 12×12 de la zone de support ($\omega = 6$).

3.7.2. Evaluation et comparaison

Pour évaluer la performance de notre algorithme de mise en correspondance nous utilisons un banc d'essai proposé par Shcarstein et Szeliski dans [Sze02] et mis à jour par Yoon et al. dans [Yoo05]. Nous avons évalué notre approche en utilisant les paires stéréoscopiques de test suivantes : Tsukuba, Venus, Sawtooth et Map. Le taux d'erreur utilisé dans le banc d'essai proposé par Shcarstein et Szeliski mesure

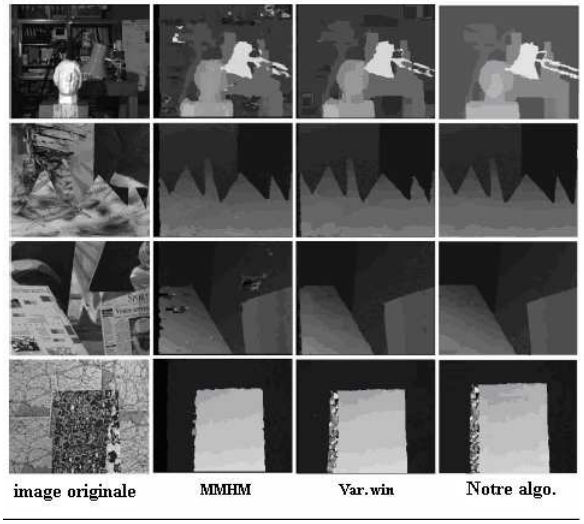

Figure 3.9. Résultats pour les paires stéréo (du haut vers le bas) Tsukuba, Swatooth, and Venus et Map.

le pourcentage des pixels pour lequel la difference entre la disparité mesurée et la disparité réelle est supérieure à un pixel. Ce taux d'erreur est une bonne mesure qualitative pour la comparaison des performances des différents algorithmes.

La Figure 3.9 illustre les résultats de notre algorithme ainsi que les résultats de quelques approches corrélatives en utilisant les cinq paires stéréoscopiques standards. Les résultats montrent que notre algorithme accomplit une bonne performance, même dans les zones peu texturées et dans les zones de discontinuité. Cette performance est due à l'utilisation de l'estimation des profondeurs dans les poids de support.

La Table 3.1 illustre une comparaison quantitative de notre approche avec différents algorithmes en terme de taux d'erreurs global. Notre approche donne le meilleur résultat pour les paires stéréoscopiques de *Tsukuba* et *Venus* et le deuxième meilleur résultat pour la paire *Sawtooth*. En fait, il s'agit de couples d'images naturelles avec différents niveaux de profondeur. Les différents plans de disparité sont en nombre limité et la différence de point de vue est moyenne. Les principales difficultés de mise en correspondance pour ce couple d'images sont un arrière plan fortement texturé et un grand nombre d'occultations ce qui a engendré de nombreux faux appariements pour les méthodes de mise en correspondance. Notre approche donne le troisième bon résultat pour la paire stéréoscopique *Map*.

Table 3.1. Comparaison des performances des algorithmes

	Tsukuba	sawtooth	Venus	Map
Algorithme proposé	1.38	1.33	1.11	0.3
Adapt. weights. [Yoo05]	1.51	1.14	1.14	1.47
Var. win. [Vek03]	2.35	1.28	1.23	0.24
Graph cut [Boy01]	1.94	1.30	1.79	0.31
Tree DP [Vek05]	1.77	1.44	1.21	1.45
Comp. win. [Vek01]	3.36	1.61	1.67	0.33
MMHM[Mul02]	9.76	4.76	6.48	8.42

Table 3.2. Comparaison des performances dans les régions dicontinues et les régions non texturées

	Tsukuba		Swatooth		Venus	
	disc	untex	disc	untex	disc	untex
Algorithme proposé	6.85	0.73	6.78	0.82	5.95	0.12
Adpt. Wgt[Yoo05]	7.24	0.65	5.48	0.27	4.49	0.61
Var. win[Vek03]	12.17	1.65	7.09	0.23	13.35	1.16
Graph cut[Boy01]	9.49	1.09	6.34	0.06	6.91	2.61
Tree DP[Vek05]	9.48	0.38	6.87	0.84	5.04	1.41
Comp. win[Vek01]	12.91	3.54	7.87	0.45	13.24	2.18
MMHM[Mul02]	24.39	13.85	22.49	1.87	31.29	10.36

Le Tableau 3.2 montre les résultats quantitatifs en terme de taux d'erreurs pour les régions peu texturées et les zones de discontinuité. Les meilleurs résultats de notre approche sont obtenus dans les zones de discontinuité. Les résultats sont moins bons en comptant que les zone peu-texturées. La discontinuité a tendance à être le problème des approches classiques de mise en correspondance qui sont plus perfomantes dans les grandes régions de disparité uniforme.

3.7.3. Résultats pour la détection des occultations

Le seuil utilisé pour la détection des occultations est fixé à 0.65. La carte de profondeur théorique de Tsukuba compte 84003 pixels étiquetés comme non-occultés et 1902 pixels étiquetés comme occultés. Comme illustré par la Table 3.3, notre algorithme a apparié correctement 82842 pixels et incorrectement 988 pixels et a étiqueté 181 pixels comme occultés. Parmis les 1902 pixels occultés, notre algorithme a correctement étiqueté 1798 pixels et incorctement étiqueté 104 pixels comme non occultés. Le pourcentage des les pixels occultés correctement trouvés est 94.53% ce qui représente un trés bon taux pour une application robotique de navigation temps réel.

Table 3.3. Comparaison de nos résultats avec la carte théorique Tsukuba en terme du nombre des pixels occultés et non occultés.

	occulté	non occulté	Total
occulté	1778	181	1979
non occulté	194	Correct : 82842 Incorrect: 988	83926
Total	1902	84003	85905

3.8. Conclusion

Nous avons présenté dans ce chapitre une nouvelle approche de mise en correspondance corrélative dense. La nouveauté dans cette approche est la modulation des poids de support dans la fenêtre de corrélation. Pour cela, nous avons proposé des distributions de possibilités. Le principal objectif de ces distributions de possibilités était de donner une estimation initiale rapide des disparités tout en s'affranchissant des problèmes liées aux variations de luminosité entre les vues ainsi que des problèmes d'échantillonnage liées à la nature discrète du signal délivré par le capteur. Ces distributions de possibilités s'appuient sur une modélisation sémantique des contraintes stéréoscopiques basée sur une graduation des niveaux de gris des pixels via les sous-ensembles flous. Une fonction des poids de support est proposée en se basant sur l'estimation des disparités initiales.

Les essais réalisés pour tester les performances de notre méthode nous ont permis de remarquer que les performances de score de mise en correspondance proposé sont très satisfaisantes. En effet, les résultats sont quasiment toujours meilleurs que ceux obtenus par les méthodes traditionnelles. En plus, l'utilisation de poids de support adaptatifs a donné à notre méthode un bon comportement vis-à-vis des scènes fortement texturées et ayant de fortes variations de disparité.

Le temps de calcul pour tous les couples d'images utilisés est très satisfaisant, ce qui est adapté pour une application temps réel pour la navigation d'un robot mobile. Dans le chapitre suivant, nous embarquons ce système de stéréovision dans un système de cartographie pour un robot mobile utilisant la grille d'occupation 3D.

Chapitre 4

Cartographie de l'environnement par grille d'occupation

4.1. Introduction

Une étape fondamentale de la plupart des systèmes de vision par ordinateur est d'engendrer une description synthétique d'une image, plus exploitable que l'ensemble des pixels. Il n'est pas concevable pour un robot de réagir face au monde de manière autonome sans percevoir et identifier les principales composantes de l'environnement (l'espace libre et les obstacles). Rappelons que nos travaux concernent uniquement les déplacements d'un robot dans un environnement d'intérieur. Dans ce contexte, notre robot autonome doit disposer d'une fonction de perception afin qu'il puisse, sans intervention humaine, déterminer à partir de l'image courante, où se trouvent les espaces navigables, notamment les couloirs, les entités spécifiques (murs, bureau, humains. . .) liés à la tâche qui doit être exécutée dans cet environnement.

Dans le chapitre précédent nous avons développé un système de perception de l'environnement qui se sert d'un capteur stéréoscopique pour générer des informations 3D de la scène observée. Ces informations doivent nous servir à acquérir au robot une vision fidèle de l'environnement dans lequel il évolue. Dans le premier chapitre, plusieurs types de cartographie ont été présentés, et parmi ces différents 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 mais ne traite pas le problème du SLAM. Nous supposons à tout moment que la position du robot est donnée et nous utilisons cette information pour mettre à jour le modèle de l'environnement. Nous commençons par un bref état de l'art sur les grilles d'occupation, puis nous exposons les détails techniques de l'élaboration de notre grille d'occupation. Enfin, nous

illustrons la méthode avec laquelle nous construisons une carte de navigation 2D à partir de la grille 3D construite.

4.2. Les grilles d'occupations : état de l'art

Les grilles d'occupation étaient présentées au départ par Elfes [Elf86] qui a proposé une représentation discrète 2D de l'espace navigable pour un robot mobile à l'aide des capteurs à ultrasons. Une extension sous le nom de *grille d'inférence* a été proposée dans [Elf92] par Elfes lui-même. Il a attribué à chaque cellule d'une grille d'inférence 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 [Sta00]. La méthode la plus employée pour la construction de grilles d'occupation consiste à traduire le modèle de capteur en probabilités et de combiner ces probabilités grâce aux règles de Bayes. Soit m une carte de l'environnement, m_{xy} l'état d'occupation de la cellule $< x, y >$ de la grille d'occupation et soit z une mesure issue du capteur. Un modèle inverse du capteur est utilisé pour calculer la probabilité d'occupation d'une cellule $p(\ m_{xy}\ \mid\ z\ )$. Ce modèle inverse ne tient pas compte de l'état d'occupation des cellules voisines et peut générer des cartes incohérentes. Pour cette raison, Thrun a présenté dans [Thr03] le modèle direct du capteur (*forward model*). Ce modèle décrit la probabilité de recevoir une mesure d'un objet présent sachant l'état de la carte (dans laquelle cet objet existe) a priori $p(\ z\ \mid\ m\ )$.

Si la majorité des contributions dans le domaine de la construction de grilles d'occupation utilisent le formalisme de Bayes, d'autres travaux ont tenté d'employer la logique floue pour évaluer l'état d'occupation des cellules de la grille. Oriolo et al. [Ori97] ont modélisé l'état d'occupation de la cellule par un ensemble flou. Ainsi, chaque cellule peut exprimer à la fois deux états partiels $(E = Vide\ et\ O = Occupée)$. Dans cette approche chaque cellule peut avoir des données conflictuelles fournies par les sonars, et sera considérée comme une cellule ni libre ni occupée. Éliminer 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. [Pag98] 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 1. 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 [Rib01]. 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.

Les grilles d'occupation ont été implémentées avec des télémètres lasers [Sch10], les capteurs stéréoscopiques et même en combinant les données issues de sonar, de capteurs infrarouges et des capteurs stéréoscopiques. Les travaux récents concernant les grilles d'occupations se sont focalisés sur la stéréovision comme capteurs. Franco et Bayer ont présenté une méthode pour construire une grille d'occupation visuelle en utilisant plusieurs caméras [Fra05]. Leur idée était de considérer chaque pixel de la caméra comme étant un capteur statique d'occupation. Les observations des pixels sont ensuite fusionnées pour inférer l'état final de l'occupation. Kohora et al. [Koh10] ont essayé d'éliminer le problème de *faux positifs* dans la détermination des obstacles par stéréovision. Ils ont proposé une méthode qui génère une grille d'occupation à partir des données issues d'un système stéréoscopique. Cette méthode s'est avérée robuste pour la détection des obstacles. Dans [Oni09], les auteurs ont utilisés une grille d'occupation qui génère trois types de cellules : *route*, *îlot de traffic* et *obstacle.* Ils ont effectué un filtrage temporel des faux *îlots de trafic* présents dans la grille. Les cellules de type *obstacle* ont été séparées en des obstacles statiques (infrastructure) et dynamiques. Une grille d'occupation a été construite, contenant des cellules de type *route*, *îlot de traffic*, *obstacle statiques* et *obstacle dynamiques.* Lategahn et al. présentent dans [Lat10] une chaîne de traitement complète pour générer une grille d'occupation 2D à partir de séquences d'images. En premier temps, les points 3D construits à partir des images seront situés sur la grille d'occupation. Ensuite, une mesure virtuelle est calculée pour chaque cellule pour réduire la complexité de calcul et pour rejeter les potentielles observations aberrantes. Enfin, un profil d'élévation est mis à jour pour partitionner les cellules en *sol* ou *obstacle*.

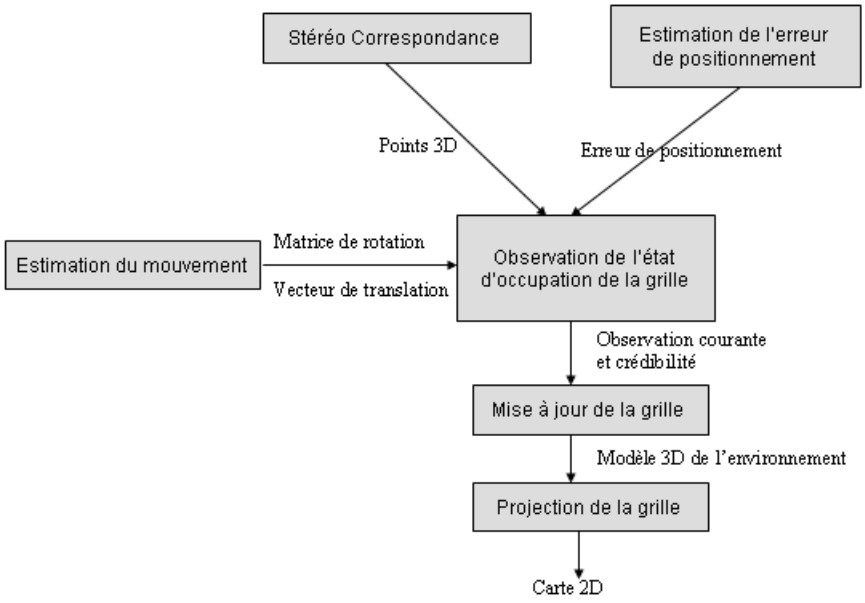

FIGURE 4.1. Vue d'ensemble du système de cartographie

4.3. Proposition d'un système de cartographie par grille d'occupations

Dans [Lat10], les auteurs illustrent les différentes techniques pour construire des cartes de l'environnement en utilisant les grilles d'occupation. Ces techniques sont basées soit sur la théorie Bayésienne (probabiliste) soit sur la théorie de l'évidence de Dempster-Shafer (théorie de l'évidence). L'utilisation de l'approche probabiliste dans les techniques par grille d'occupation a été critiquée pour différentes raisons. Tout d'abord, il est trés difficile de déterminer un modèle pour les nouveaux capteurs. Les caractéristiques de capteurs ultrasons sont connues mais des simplifications irréalistes sont indispensables pour modéliser le comportement complexe de la stéréovision. En conséquent, quelques chercheurs ont décidé de fuir la théorie des probabilités et d'inventer leurs propres règles de mise à jour de la grille [Gua10]. D'autre part, une seule probabilité ne permet pas de distinguer entre l'état *inconnu* et *incertain*. Ainsi, on ne peut pas déterminer si une zone n'a pas été du tout balayée (par exemple dû à l'occultation) ou les mesures des capteurs n'ont pas été fiables. Pour ces raisons, nous proposons dans cette section une nouvelle approche pour construire et mettre à jour une grille d'occupation de l'environnement [Gha10c]. La Figure 4.1 donne une vue d'ensemble des différents modules de notre système de cartographie qui sera détaillé par la suite.

4.3.1. Modélisation de l'imprécision du capteur stéréoscopique

Tous les capteurs fournissent des informations imprécises et entachées d'erreurs [Far08] ; la difficulté majeure rencontrée lors de la construction des cartes d'occupation consiste à représenter fidèlement cette imprécision. Une bonne représentation permettra de manipuler de facon précise et rigoureuse les données contenues dans le modèle et fournira les bases d'une qualification des algorithmes exploitant ces modèles. Cette qualification, indispensable, permettra d'établir un modèle des algorithmes et permettra ainsi de propager, le long des chaînes de traitement, la nature incertaine et non-déterministe des perceptions.

Afin de modéliser les données perçues et l'imprécision qui les caractérise, il est nécessaire de construire un modèle du capteur utilisé. Le rôle de ce modèle est de décrire la nature des données perçues, c'est à dire l'information fournie par chaque perception. Pour le cas de la stéréovision, un tel modèle est difficilement calculable de façon analytique : la compréhension du système physique, électronique et logiciel sous-jacent est un processus trop complexe pour permettre d'en déduire d'une façon rigoureuse un modèle. Cependant, en combinant expérience et analyse du phénomène physique mesuré, il est possible d'obtenir un modèle réaliste. Essentiellement, ce capteur perçoit une distance. Cette distance est ensuite interprétée comme étant la signature d'un objet de l'environnement, situé à cette distance. Étant donné un point 3D et l'écart-type modélisant sa précision le modèle de son incertitude peut alors être représenté.

Durant les dernières années, quelques travaux se sont focalisés sur l'analyse de l'imprécision dans les systèmes de vision artificiels basés sur la stéréovision. A titre d'exemple, Blostein et Huang [Blo87] ont examiné l'exactitude pour l'obtention des positions 3D des objets basés sur la triangulation des informations de la mise en correspondance stéréoscopique. Ils ont déterminé une expression rapprochée de la distribution de probabilités des erreurs de positions le long de chaque direction du système de coordonnées du capteur stéréoscopique. Plus récemment, Jianxi et al. ont présenté dans [Jia08] une analyse de l'erreur pour la reconstruction 3D tenant compte seulement de la précision des paramètres de calibrage de la caméra.

Dans un système de stéréovision binoculaire, chaque position $3D$ d'un point physique P est déterminée à partir de ces deux projections $U_l = [u_l, v_l]$ sur l'image de gauche et $U_r = [u_r, v_r]$ sur l'image de droite. A cause de l'imprécision dans les mesures, le système stéréoscopique projette U_l et U_r avec une imprécision qui se propage pour induire une incertitude dans la position du point P. Non examinée, l'incertitude se cumule pour engendrer des fausses cartes. Dans le chapitre

précédent, nous avons développé un algorithme de stéréovision par corrélation de pixels : il se base sur la connaissance précise des positions respectives des caméras (obtenues par calibration) et sur la détermination d'appariements entre les pixels des images gauche et droite. Une fois les pixels appariés, il serait possible de déterminer les profondeurs des points si on connait les paramètres intrinsèques et extrinsèques du capteur. Considérons deux pixels (u, v) et $(u, v + d)$ appariés dans l'espace de disparité $3D$ (défini dans le chapitre précédent), il est possible de déterminer la position $3D$ du point physique associé par une simple triangulation (4.1).

$$\begin{cases} x = \frac{zu}{f} \\ y = \frac{zv}{f} \\ z = \frac{fb}{d} \end{cases} \tag{4.1}$$

$U = (u, v)$ est la position du pixel dans le plan de l'image de référence, $X = (x, y, z)$ est la position du point $3D$ dans le plan référentiel de la caméra de gauche et d est la disparité calculée pour le pixel (u, v). Les équations précédentes ne tiennent pas compte de l'imprécision des mesures. Les deux sources d'imprécision dans un système stéréoscopique sont l'algorithme de la mise en correspondance lui-même c'est-à-dire les erreurs dans les disparités déterminées, et les erreurs dans les paramètres de calibrage. Partant de cette analyse, nous définissons un modèle d'incertitude de notre capteur stéréoscopique basé sur deux partie : l'*erreur de pointage (p)* et *l'erreur d'appariement (m).* L'erreur de pointage est l'erreur sur la position de la caméra de référence qui est liée à l'incertitude du calibrage des caméras. L'erreur d'appariement est l'imprécision sur la disparité et est liée directement à l'exactitude de l'algorithme de mise en correspondance. L'erreur de pointage, est une caractéristique du capteur stéréoscopique utilisé. Afin d'avoir une estimation de l'erreur d'appariement, nous avons étudié les variations de disparités lors de l'application de notre algorithme de mise en correspondance à une centaine d'images d'une même scène d'intérieur. Une moyenne de 0.06 pixels a été définie pour l'erreur d'appariement (m). Compte tenu de ce modèle d'imprécision, la matrice de covariance dans l'espace $3D$ (u, v, d) est donnée par (4.2).

$$C_U = \begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & m \end{bmatrix} \tag{4.2}$$

Pour obtenir la matrice de covariance C_X du point (x, y, z) associé avec le pixel (u, v, d) dans l'espace de disparité $3D$, nous propageons l'erreur de l'espace (U, V, D)

à l'espace (X, Y, Z) en appliquant la méthode définie par Faugeras (Faugeras, 1997). La matrice de covariance C_X est donnée par (4.3).

$$C_X = J_{X,U} C_U J_{X,U}^T \tag{4.3}$$

$J_{X,U}$ étant la matrice Jacobienne donnée par (4.4).

$$J_{X,U} = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} & \frac{\partial x}{\partial d} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} & \frac{\partial y}{\partial d} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} & \frac{\partial z}{\partial d} \end{bmatrix} = \begin{bmatrix} \frac{b}{d} & 0 & \frac{-ub}{d^2} \\ 0 & \frac{b}{d} & \frac{-vb}{d^2} \\ 0 & 0 & \frac{-fb}{d^2} \end{bmatrix} \tag{4.4}$$

On obtient la matrice C_X donnée par (4.5).

$$C_X = \begin{bmatrix} A + Bu^2 & uvB & ufB \\ uvB & A + Bv^2 & vfB \\ ufB & vfB & A + Bf^2 \end{bmatrix} \tag{4.5}$$

Avec $A = (\frac{bp}{d})^2$; $B = m\left(\frac{b}{d^2}\right)^2$.

La matrice C_X dont l'expression est donnée par (4.5), représente le modèle d'erreur d'un capteur stéréoscopique dont l'erreur de pointage est p et l'erreur d'appariement est m dans l'espace (X, Y, Z). Nous pouvons calculer l'erreur sur la position de tout point $3D$ calculé en utilisant ce modèle. En particulier, nous pourrons calculer l'erreur sur la distance ρ_i qui sépare un point 3D de coordonnées (x, y, z) dont la position est donnée par le capteur stéréoscopique et un point fixe (x_i, y_i, z_i) dont la position est certaine. L'expression de la distance ρ_i est donnée par (4.6). L'erreur $\triangle\rho_i$ sur la distance ρ_i est donnée par (4.7).

$$\rho_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \tag{4.6}$$

$$\triangle\rho_i = J_{\rho,X} C_X J_{\rho,X}^T \tag{4.7}$$

Avec $J_{\rho,X}$ est donnée par (4.8). L'expression de $\triangle\rho_i$ est donnée par (4.9).

$$J_{\rho,X} = \begin{bmatrix} \frac{\delta\rho}{\delta x} \\ \frac{\delta\rho}{\delta y} \\ \frac{\delta\rho}{\delta z} \end{bmatrix}^T = \frac{1}{\rho} \begin{bmatrix} x \\ y \\ z \end{bmatrix}^T \tag{4.8}$$

$$\triangle\rho_i = \frac{Cx^2 + Dy^2 + Ez^2 + 2Fxy + 2Gxz + 2Hyz}{\rho_i^2} \tag{4.9}$$

Avec :
$C = (\frac{bp}{d})^2 + m\left(\frac{ub}{d^2}\right)^2$
$D = (\frac{bp}{d})^2 + m\left(\frac{vb}{d^2}\right)^2$
$E = (\frac{bp}{d})^2 + m\left(\frac{fb}{d^2}\right)^2$
$F = uvm\left(\frac{b}{d^2}\right)^2$
$G = ufm\left(\frac{b}{d^2}\right)^2$
$H = vfm\left(\frac{b}{d^2}\right)^2$

L'imprécision sur la distance calculée entre un point $3D$ fournie par le capteur stéréoscopique et un point fixe de l'espace de travail dont la position est donnée par $i = (x_i, y_i, z_i)$ est donnée par (4.9). Dans le prochain paragraphe, les points i représenteront les centres des cubes qui composent la grille d'occupation $3D$. L'erreur calculée sera utilisée pour déterminer l'état d'occupation de la grille à partir des points $3D$ fournis par notre système de mise en correspondance stéréoscopique décrit dans le chapitre précédent.

4.3.2. Grille d'occupation spatio-temporelle

Dans notre approche, une grille d'occupation 3D est utilisée pour représenter l'environnement. La représentation en grille d'occupation emploie une partition multidimensionnelle de l'espace de travail en voxels. Chaque voxel stocke une valeur estimative de son état d'occupation. L'environnement est discrétisé en des cubes uniformes (les voxels) qui sont les plus petites entités du modèle 3D.

La taille des voxels est choisie selon la résolution désirée du modèle. Suivant le volume de l'environnement de travail, il peut être décomposé en millions de voxels.

$$W_{3D} = \bigcup_{i=1}^{n} V_i, \forall i, j \in [1, n], V_i \cap V_j = \phi \tag{4.10}$$

W_{3D} est l'espace de travail tridimensionnel, V_i représente le voxel i du modèle. Le robot obtient les informations concernant son environnement à travers ses caméras. Il utilise une approche de stéréovision pour acquérir les informations concernant les points 3D de son environnement. Une observation $O_t(V_i)$ sur l'occupation d'un voxel i est calculée en se basant sur les images stéréoscopiques prises à l'instant t. L'observation sur l'occupation d'un voxel admet un champ de valeurs entre 0 et 1. Une observation de 1 signifie que le voxel est *occupé* à l'instant t et vis versa. L'observation sur les occupations des voxels sert à mettre à jour l'état du voxel à l'instant t défini par la fonction $S_t(V_i)$. Ainsi, notre modèle permet de gérer une

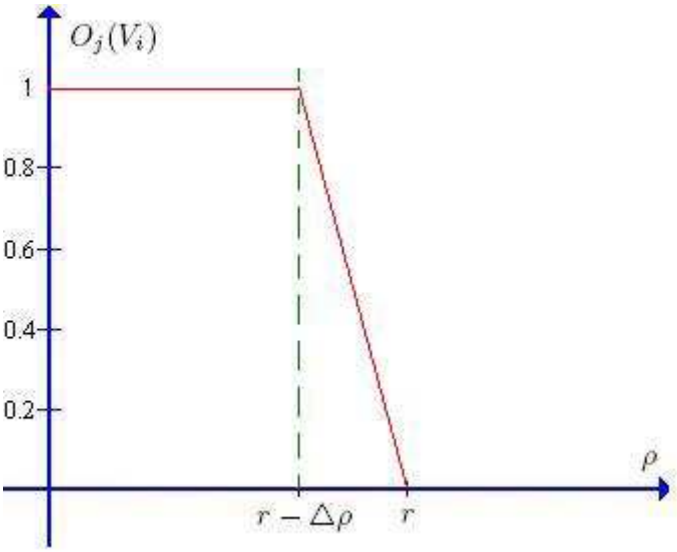

Figure 4.2. Observation sur l'occupation d'un voxel compte tenu de l'incertitude dans la position du point $3D$.

mise à jour spatio-temporelle du modèle $3D$ de l'environnement à travers une grille d'occupation dynamique.

4.3.3. Observation sur l'occupation d'un voxel

Le robot mobile reçoit les informations sur son environnement à partir d'un système stéréoscopique qui lui fournit des points $3D$. Par une simple triangulation, il calcule les coordonnées de chaque point j fourni. Un voxel est *occupé* s'il contient au moins un point 3D, c'est-à-dire si la distance ρ (donnée par l'équation (4.6)) qui sépare son centre d'un des points $3D$ est inférieure à r ($2r$ étant la résolution de la grille d'occupation). L'observation sur l'occupation d'un voxel en se basant sur la position d'un point $3D$ tient compte de l'incertitude $\triangle\rho$, dont l'expression est donnée par (4.9). L'expression de l'observation sur l'occupation d'un voxel V_i en se basant sur la position d'un point j est donnée par (4.11).

$$O_j(V_i) = \begin{cases} 1 & si \quad 0 \leq \rho < r - \triangle\rho \\ 1 - \frac{\rho}{r} & si \quad r - \triangle\rho \leq \rho < r \\ 0 & autrement \end{cases} \tag{4.11}$$

L'observation sur l'occupation d'un voxel est déterminée en utilisant toute l'information $3D$ fournie par l'algorithme de mise en correspondance. Un ou plusieurs points peuvent co-exister à l'intérieur d'un même voxel. Plus le voxel contient de points, plus l'observation sur son occupation est forte. L'expression de l'observa-

tion globale à un instant t, l'instant auquel les images stéréoscopiques sont prises, est déterminée en utilisant l'opérateur classique de l'union (max).

$$O_t(V_i) = min\left(\max_j \left[O_j(V_i)\right] + \lambda\frac{N(V_i)}{6r^3}, 1\right) \tag{4.12}$$

Une gratification de $\lambda\frac{N(V_i)}{6r^3}$ est attribuée à l'observation basée sur la plus grande quantité d'information $3D$ fournie. $N(V_i)$ étant le nombre de points $3D$ fournis par le système stéréoscopique et qui se trouve à l'intérieur du voxel V_i. λ étant une constante empirique. L'observation $O_t(V_i)$ renseigne sur l'état d'occupation du voxel V_i en se basant sur les informations de profondeur fournies par le système de mise en correspondance à l'instant t. Nous avons donc pu déterminer l'observation sur l'état d'occupation de la grille à un instant t. Reste à utiliser cette information pour mettre à jour l'état de la grille après plusieurs observations similaires. Dans le paragraphe suivant nous détaillons notre méthode pour la mise à jour des états d'occupation des voxels en tenant compte de l'observation courante et des informations a priori.

4.3.4. Mise à jour de la grille d'occupation

La principale hypothèse utilisée dans les approches par grille d'occupation est que l'état de chaque cellule est supposé indépendant de l'état des autres cellules. Ainsi la complexité de l'estimation de la grille complète est "cassée". La probabilité d'occupation de chaque cellule est estimée indépendamment de celle des cellules voisines. Estimer la grille complète revient à appliquer N fois l'estimation d'une seule cellule, N étant le nombre de cellules composant l'environnement complet du robot. Si cette hypothèse est nécessaire pour assurer une estimation de la grille d'occupation en un temps raisonnable, elle peut entraîner un manque de précision des cartes obtenues. Il est possible de vérifier l'interdépendance des cellules sans toucher trop à la complexité de l'algorithme. Notre idée est de vérifier l'homogénéité de la grille dans un voisinage local. En effet, il est très peu probable de trouver un voxel occupé isolé au milieu d'un espace vide. Outre l'homogénéité, une des caractéristiques d'un modèle de l'environnement est la fréquence de la mise à jour. Un modèle se basant sur des vieilles mesures ou sur un nombre non élevé de mesures n'est pas tout à fait fiable. Dans notre approche de mise à jour, nous tenons compte de l'homogénéité des mesures dans un voisinage local, la quantité des mesures a priori ainsi que l'âge de la dernière observation. Pour cela nous définissons *une valeur de crédibilité* $k_{i,t}$ donnée par (4.13) qui renseigne sur la confiance qu'on a dans l'observation $O_t(V_i)$.

FIGURE 4.3. Scène représentant un bureau dans un laboratoire

$$k_{i,t} = \frac{N_{i,t}(1 - \mathcal{H}_{i,t})}{\sqrt{2\pi}} exp\left(-\frac{t_0}{2\sigma^2(t - t_{last})}\right) \tag{4.13}$$

La valeur $k_{i,t}$ est située entre 0 et 1. Elle dépend de l'homogénéité $\mathcal{H}_{i,t}$ (dont l'expression est donnée par (4.14)) dans un voisinage local N et de l'âge de la dernière observation $t - t_{last}$ sur le voxel. Plus les informations a priori sont vieilles, plus on accorde d'importance à la nouvelle observation. La fonction de mise à jour de la grille d'occupation est donnée par (4.15).

$$\mathcal{H}_{i,t} = \frac{\sum_{V_j \in N} \left| O_t(V_i) - O_t(V_j) \right|}{|N|} \tag{4.14}$$

$$\begin{cases} S_{t+1}(V_i) = (1 - k_{i,t+1}) S_t(V_i) + k_{i,t+1} O_{t+1}(V_i) \\ S_{t_0}(V_i) = 0 \end{cases} \tag{4.15}$$

La Figure 4.4 représente les états d'occupation dans une grille $3D$ modélisant la scène de bureau décrite par la Figure 4.3 L'état d'un voxel est représenté sur une échelle de niveau de gris de 1 à 255. Les points blancs représentent des pixels occupés.

Le modèle de mise à jour de la grille, présenté dans ce chapitre, donne la possibilité de modéliser des environnements dynamiques au contraire des grilles d'occupation classiques qui sont utilisées pour représenter des environnements statiques.

4.3.5. Génération d'une carte 2D par projection

En trois dimensions, l'utilisation d'une grille d'occupation est cependant relativement coûteuse : le volume de données à traiter et la quantité de calculs engendrés sont loin d'être négligeables. Une carte d'occupation $3D$ a besoin d'être initialisée,

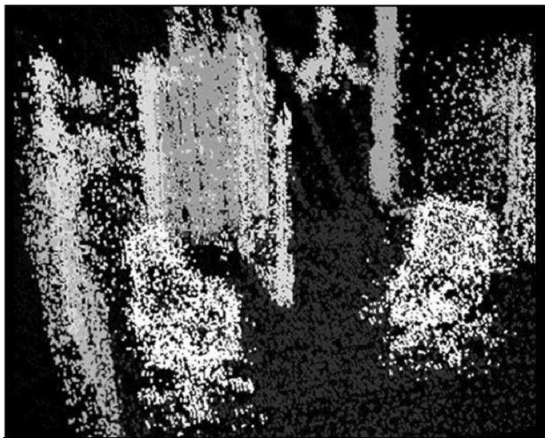

FIGURE 4.4. Résultats pour la représentation des états des voxels en utilisant 3 pairs d'images stéréoscopiques prises à partir de 3 positions différentes, les voxels blancs ont une valeur élevée de l'état d'occupation.

ainsi sa taille est aussi importante que la region à cartographier. Dans les cas où on a besoin d'une résolution fine la consommation de mémoire devient prohibitive. Dans ces cas là, la diminution de la grille d'occupation $3D$, obtenue à partir du capteur stéréoscopique, devient une nécessité pour répondre au besoin d'efficacité dans des environnements temps réel.

Dans notre application, la grille d'occupation $3D$ est réduite à une carte visuelle $2D$ pour réduire le temps de calcul. La carte 2D est obtenue par projection orthogonale de la grille déterminée comme décrit dans les paragraphes précédents. L'espace de travail 2D est décomposé en cellules comme décrit dans (4.16).

$$W_{2D} = \bigcup_{k=1}^{m} C_{k,} \forall k, l \in [1, m], C_k \cap C_l = \phi \tag{4.16}$$

L'état d'occupation d'une cellule est donnée par les états des voxels qui se projètent en elle. L'opérateur max est utilisé pour déterminer l'état d'occupation de la cellule comme décrit dans (4.17).

$$S(C_k) = \underset{V_i \in P(C_k)}{max} (S(V_i)) \tag{4.17}$$

Avec $S(C_k)$ est l'état de la cellule C_k et $P(C_k)$ est l'ensemble de tous les voxels dont les projections horthogonales sont sur la cellule C_k.

Chaque cellule est assignée à plusieurs variables :

- Un vecteur d'états binaire $(Libre, inconnu, occupé)$, une valeur de $(1, 0, 0)$ du vecteur d'états, par exemple, exprime que la cellule en question est libre.
- Les coordonnées de la cellule cartésiennes et angulaires.

FIGURE 4.5. Environnement de test : hall du laboratoire LIRMM

FIGURE 4.6. Le robot Pioneer3 utilisé pour l'expérimentation

– Une variable booléenne V_k qui exprime l'état de visibilité de la cellule par rapport au robot ($masquée = 0, visible = 1$).

4.4. Résultats expérimentaux

4.4.1. Environnement d'expérimentation

Afin de valider notre approche, nous avons testé notre méthode de cartographie incrémentale dans un environnement d'intérieur fermé et structuré. Il s'agit d'un hall du laboratoire LIRMM qui dessert des bureaux (voir Figure 4.5). Cette expérimentation est basée sur des acquisitions stéréoscopiques à une fréquence de 5Hz par une caméra stéréoscopique montée sur le robot Pioneer 3 (voir Figure) qui se déplace à une vitesse de 0.2m/s. La taille des voxels choisie est de 2,5 cm.

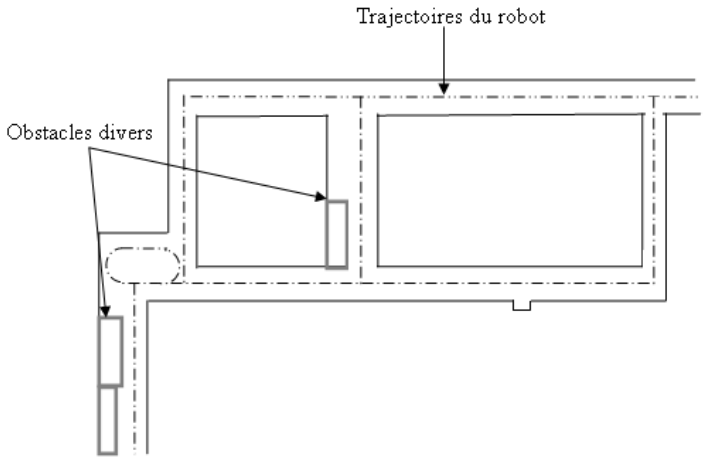

FIGURE 4.7. Plan de l'environnement d'expérimentation

4.4.2. Choix d'une résolution

La résolution des cartes détermine la précision à laquelle l'environnement va être modélisé. Plus cette précision est grande, plus le modèle sera sensible au bruit des données. Il est donc nécessaire de trouver un compromis entre modélisation fine mais bruitée et une modélisation plus grossière.

4.4.3. Analyse et comparaison des résultats

Pour évaluer notre approche de cartographie, nous avons utilisé un banc d'essai pour la cartographie basée sur les grilles d'occupation proposé par Thomas Collins et al. [Col07]. Ce banc d'essai renferme une méthode de corrélation entre la carte réelle et la carte générée [Col05], une méthode de comparaison directe appelée *map scoring (MS)* basé sur le travail de [Mar96] et une technique d'analyse de chemin qui teste l'utilité de la carte générée comme moyen de navigation plutôt qu'une image.

La métrique de corrélation est calculée par une mise en correspondance entre la carte générée et la carte théorique en utilisant le score de corrélation dans (4.18).

$$C = \frac{\langle M.T \rangle - \langle M \rangle \langle T \rangle}{\sigma(M) \times \sigma(T)} \tag{4.18}$$

Où M est la carte à mettre en correspondance, T est la carte théorique, $<>$ est l'opérateur de moyenne et σ est la déviation standard sur la surface à mettre en correspondance. C est un pourcentage qui spécifie la similarité entre deux cartes.

La méthode *map score* utilise une carte normalisée pour calculer la somme des

	Corrélation	MS total	MS Occupé	Faux positif	Faux Négatif
Clé	A	B	C	D	E
Notre approche	53.56	17.21	19.63	44.47	14.93
Rang	1	1	2	1	1
Moravec et Elfes, 85	39.18	33.73	23.56	72.84	21.34
Matthies et Elfes, 88	40.69	28.27	24.82	69.17	25.91
Thrun 93	38.34	25.97	29.71	77.13	27.93
Konolige 97	40.54	20.25	19.69	63.45	22.64
Thrun 01	50.13	18.56	17.39	50.15	16.68

Table 4.1. Comparaison de nos résultats avec ceux des différentes approches de cartographie en utilisant le band d'essai de Collins et al.

différences absolues entre les cellules correspondantes. Le *map score* est donnée par l'expression (4.19).

$$MapScore = \sum_{m_{xy} \in M, n_{xy} \in N} \left(m_{xy} - n_{xy}\right)^2 \tag{4.19}$$

Où m_{xy} est la valeur de la cellule à la position (x, y) sur la carte M et n_{xy} est la valeur de la cellule à la position (x, y) sur la carte normalisée N. Le *map score* donne une valeur positive représentant la différence entre deux cartes (généralement la carte idéale de l'environnement et la carte à évaluer). Donc, plus la valeur est basse plus les deux cartes sont similaires.

Ce banc d'essai calcule aussi une valeur de *faux positifs* qui exprime le degré auquel le chemin crée dans la carte générée peut causer la collision entre le robot et un obstacle structuré dans l'environnement réel, une valeur de *faux négatifs* qui exprime le degré auquel le robot est capable de planifier un chemin entre deux positions en utilisant la carte générée alors qu'il n'existe pas réellement sur la carte réelle. La méthode qui permet de calculer les faux positifs et les faux négatifs est détaillée dans [Col07].

Les expérimentations consistent à tester notre paradigme de cartographie dans un hall du laboratoire LIRMM (Figure 4.5).

Les Figures 4.8 (a) et 4.8 (b) montrent respectivement la carte idéale et la carte normalisée pour l'environnement de test. La Figure 4.9 présente les cartes générées en utilisant différentes approches de l'état de l'art dans l'environnement d'expérimentation.

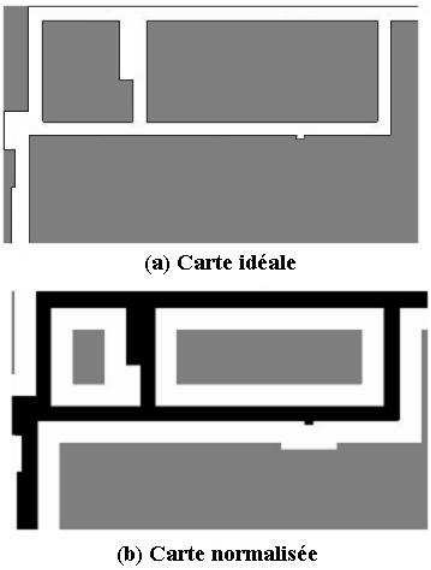

FIGURE 4.8. Normalisation de la carte idéale pour les tests

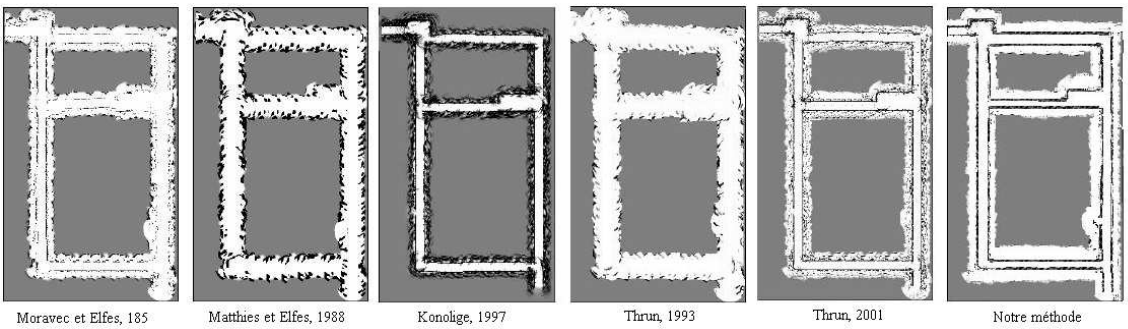

FIGURE 4.9. Cartes générées en utilisant différentes approches dans l'environnement d'expérimentation

4.4.3.1. Analyse des résultats de corrélation

Comme le montre le Tableau 4.1 (clé A), notre méthode réalise la plus haute corrélation suivie de celle de Thrun, 2001 [Thr01]. Le paradigme de Thrun [Thr93] réalise la plus basse corrélation parmi toutes les méthodes testées. Selon Collins et al. [Col07] cela peut être rapporté à deux causes. La première est que les paradigmes qui ont un coût de corrélation bas ont tendance à surestimer l'espace libre comme le montre la Figure 4.9. La deuxième est que ces paradigmes modélisent les extrémités des capteurs comme libres. Le problème de la surestimation de l'espace libre est dû au mécanisme de mise à jour de la grille d'occupation. En effet, selon Collins et al. [Col07], l'approche Bayésienne de mise à jour rend le paradigme susceptible aux fluctuations dans les valeurs d'occupation quand elle est utilisée en conjonction avec une approche qui a un penchant à la surestimation de l'espace libre ou occupé. Dans notre paradigme, nous avons utilisé une approche de mise à jour basée sur les valeurs de crédibilité qui prend en compte l'age de la dernière observation et l'homogénéité dans un voisinage local. Notre approche de mise à jour ne souffre pas du problème de la surestimation de l'espace libre comme le montre la Figure 4.9.

4.4.3.2. Analyse des résultats du *Map Score*

Une valeur basse du MS indique moins de différence entre la carte générée et la carte réelle. La Table 4.1 (clé B) présente les résultats du MS sur toute la carte. Comme on peut le remarquer, notre méthode a la meilleure performance en réalisant la plus basse valeur de MS suivie par celle de Thrun [Thr01] puis celle de Konolige [Kon97]. Le MS compare la carte générée avec la carte théorique. Les raisons de cette performance sont les mêmes soulignées pour les résultats de corrélation. La Table 4.1 (clé C) présente les résultats du MS pour les cellules occupées. la méthode de Thrun [Thr01] réalise la meilleure performance suivie de notre méthode. Ceci est parce que nous utilisons un seuil relativement élevé pour marquer une cellule comme occupée. En effet, le premier critère à satisfaire dans notre travail est la robustesse des chemins. Pour cette raison, touts les états des cellules sont marqués comme inconnus au départ. L'état d'une cellule est modifié pour occupé ou libre uniquement lorsqu'on a les informations suffisantes qui favorisent ce changement.

4.4.3.3. Analyse des résultats des faux positifs

Les approches de cartographie ayant une faible performance concernant la métrique des faux positifs sont ceux qui ont tendance à mettre à jour excessivement l'espace libre[Col07]. Notre approche de cartographie a réalisé la meilleure perfor-

mance comme montré dans la Table 4.1 (clé D) suivie de l'approche décrite dans [Thr01]. La structure de l'espace occupé dans les cartes des autres paradigmes est la cause d'un nombre de chemins inconsistants créés et qui est du à la tendance de modéliser les extrémités des capteurs comme des zones libres. Par conséquence, des chemins, irréalisables sur l'environnement réel, sont creés sur la carte générée. Les chemins crées par notre approche sont des chemins utilisables et robustes.

4.4.3.4. Analyse des résultats des faux négatifs

La métrique des faux positifs a été utilisée pour décrire l'utilisabilité de la carte comme base pour une navigation sans collision. Cette métrique décrit le pourcentage de chemins qui sont des faux-négatifs dans la carte et est reliée à l'utilisabilité de la carte comme base pour la planification d'un chemin dans l'environnement réel du robot. Cette métrique informe sur le nombre de chemins qui ne peuvent pas être déterminés à partir de la carte alors que, réellement, le robot peut parcourir ces chemins dans son environnement. La Table 4.1 (clé E) montre que notre approche réalise la meilleure performance en générant la carte la plus appropriée pour la planification de chemins.

4.5. Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche pour la construction d'une grille d'occupation à partir d'un capteur stéréoscopique. L'algorithme proposé est robuste et répond aux besoins de la navigation dans un environnement temps réel. Une représentation 3D intrinsèquement incertaine basée sur la propagation de l'erreur stéréoscopique a été utilisée. Une nouvelle méthode de mise à jour, basée sur une valeur de crédibilité, a été employée pour mettre à jour le modèle de l'environnement. Une carte de navigation 2D est enfin obtenue par projection de la grille d'occupation. Les résultats expérimentaux ont été rapportés pour illustrer la performance satisfaisante de la méthode. Les cartes obtenues par notre approche ont été précises et permettent au robot de différencier entre l'espace libre et l'espace occupé.

Chapitre 5

Détection et suivi d'objets en temps réel

5.1. Introduction

Le suivi d'objets dans une séquence d'images occupe une place prépondérante dans plusieurs domaines en relation avec la vision artificielle : surveillance, robotique, etc. Deux contraintes se posent pour développer un algorithme de suivi robuste : la première est la qualité du suivi et la deuxième est l'aspect temps réel exprimé via la rapidité et la complexité de l'algorithme. Nous nous intéressons dans ce chapitre au suivi d'objets basé sur la couleur et la forme. Le suivi basé couleur a été principalement considéré selon deux approches. Dans l'algorithme du mean shift la recherche de l'objet est effectuée en minimisant une distance entre histogrammes de couleur selon une méthode de type descente de gradient, et conduit à une bonne précision de suivi. Cependant, cette recherche étant déterministe, elle ne permet pas d'être robuste aux occultations importantes, et l'algorithme peut échouer en présence d'un autre objet de couleurs similaires, ou dans le cas de grands déplacements. Dans un autre cadre, d'autres méthodes utilisent le même type de critère de ressemblance entre histogrammes, mais l'intègrent dans un filtre particulaire. La densité de probabilité a posteriori de la position de l'objet est discrétisée en un ensemble de particules. L'évolution de ces particules et l'estimation de leur moyenne remplacent ici la minimisation effectuée dans l'algorithme du mean shift. L'utilisation de ce cadre probabiliste du filtrage particulaire induit une meilleure robustesse vis-à-vis des occultations ou de la présence d'objets similaires. Cependant ces dernières méthodes restent difficiles à mettre au point et peu adaptée au temps réel.

Dans ce chapitre nous présentons une approche de suivi de balle colorée en utilisant une méthode d'optimisation tout en essayant de palier les problèmes d'occultations, d'encombrement de fond et de présence d'autres objets de même couleur que la balle. Pour cette fin, nous commençons par présenter les principes de la vision active dans lequel s'inscrivent ces types de problèmes. Nous décrivons ensuite les méthodes existantes pour le suivi d'objet. Enfin nous donnons une description

détaillée de notre approche de suivi d'objets et les expérimentations aux quelles elle a aboutit.

5.2. Vision active

Le processus de vision occupe plus d'un tiers du cerveau humain [Fin03]. Pas surprenant, puisque ce processus est l'opération de collecte d'informations la plus importante par laquelle le cerveau est sensibilisé sur son monde extérieur ! La vision chez l'homme est un processus dynamique pendant lequel un échantiollonnage en continu de l'environnement est effectué par l'oeil humain. Ce processus a inspiré les chercheurs qui ont compris qu'il est très important d'avoir un tel système de vision active chez les robots.

C'est avec le développement d'ordinateurs de plus en plus rapides et avec de plus grandes capacités de stockage, qu'il est devenu possible d'embarquer des fonctions de vision active sur des machines. Contrairement à l'idée initiale de la vision passive [Mar82], utilisée au début de la robotique et principalement dans les premières approches de la vision par ordinateur, la vision active [Bla92] s'intéresse au contrôle des algorithmes ainsi que des structures optiques et mécaniques des caméras, afin de simplifier (a) le processus de la vision par ordinateur et en même temps, (b) l'interaction avec le monde extérieur. Il est important de signaler qu'un système de vision active, crée lui-même des processus actifs dans le monde.

Citons les caractéristiques essentielles d'un tel système :

- *opération en continu.* Le système est toujours en marche.
- *filtre d'information.* Un système de vision active agit comme un filtre d'informations, ne retenant que ce qui est pertinent pour la fonction visuelle en cours d'exécution ; c'est la notion de purposive vision introduite par Y.Aloimonos [Alo92, Baj88].
- *temps réel.* Afin d'être utile, un système de vision active doit retourner les résultats dans un délai fixe, qui dépend de l'application.
- *contrôle du traitement.* Afin de réaliser des fonctions en temps fixe ou au moins, borné, le traitement est restreint à des régions d'intérêt. ; le capteur est configuré pour faciliter le choix et la taille de cette région (centrage, zooming.... pour garantir une résolution fixe).

Pour qu'un robot puisse agir et intervenir dans un monde dynamique, par nature très complexe, il est vital qu'il ait la capacité de commander activement les paramètres de la caméra. Les systèmes de vision active doivent donc avoir des mécanismes pour commander les paramètres tel que l'orientation, le zoom, l'ouverture

et la convergence (pour la stéréovision) en réponse aux besoins de la tâche et aux stimulus externes.

Plus généralement, la vision active renferme des mécanismes d'attention, de perception sélective dans l'espace, la résolution et le temps. Ces mécanismes sont introduits en modifiant soit les paramètres physiques de la caméra, soit la manière avec laquelle les données sont traitées après acquisition.

L'étroit accouplement entre perception et action, proposé dans le paradigme de la vision active, ne se limite pas au contrôle du processus visuel. Ce processus doit aussi interagir fortement avec les activités qui exploitent les données visuelles (la navigation, la manipulation, la signalisation du danger, etc.) ; ceci permet des simplifications dans les algorithmes de commande et les représentations de scènes, des temps de réaction rapides, et finalement, des taux de succès plus élevé pour ces activités.

L'application de la vision active facilite certaines tâches qui seraient impossibles en utilisant la vision passive. L'amélioration de la performance peut être mesurée en termes de fiabilité ou robustesse, de répétabilité et de réactivité (ou temps d'exécution) lorsque sont traitées certaines tâches spécifiques, mais aussi par la généricité, ou capacité à traiter une grande diversité de tâches.

Le suivi d'objets mobiles entre dans le cadre de la vision active. Le robot mobile doit détecter l'objet, estimer sa position et la distance qui le sépare de lui et générer la commande nécessaire qui lui permet de suivre l'objet. Toutes ces tâches doivent répondre aux caractéristiques déjà décrites de la vision active à savoir, le temps réel, l'opération en continu, le filtrage des informations visuelles... Dans le paragraphe suivant, nous donnons une classification des méthodes de suivi d'objets.

5.3. Suivi d'objets : Etat de l'art

Dans la littérature, de nombreuses méthodes de suivi d'objets ont été présentées ; une grande partie d'entre elles, peuvent être utilisées pour suivre des objets précis en temps réel [Gam12].

Plusieurs classifications des méthodes de suivi visuel d'objets ont été proposées dans la littérature ; elles dépendent autant des auteurs, que du but pour lequel ces méthodes ont été conçues. Nous considérons la classification donnée dans [Mcl73], où selon les auteurs, les méthodes de suivi visuel peuvent être divisées en quatre classes :

– *Méthodes de suivi fondées sur des modèles*. Ces méthodes repèrent des caractéristiques connues dans la scène et les utilisent pour mettre à jour la position de

l'objet. Parmi ces méthodes, citons celles qui exploitent les modèles géométriques fixes [Bla92], et les modèles déformables [Bla98].

- *Méthodes de suivi de régions ou blobs.* Cette sorte de méthodes se caractérise par la définition des objets d'intérêt comme ceux qui sont extraits de la scène en utilisant des méthodes de segmentation. Citons les nombreuses méthodes qui détectent une cible à partir de son mouvement sur un fond statique ou quasiment statique [Kol94].
- *Méthodes de suivi à partir de mesures de vitesse.* Ces méthodes peuvent suivre les objets en exploitant les mesures de leur vitesse dans l'image, avec des mesures telles que le flux optique ou des équivalents [Smi95].
- *Méthodes de suivi de caractéristiques.* Ces méthodes suivent certaines caractéristiques de l'objet, comme des points, des lignes, des contours... [Bas94], caractéristiques ou primitives image auxquelles il est possible aussi d'imposer de restrictions globales [dru02]. Ces caractéristiques peuvent être aussi définies par la texture ou la couleur [Del98].

Cette classification n'est pas exhaustive, et à ce jour, il existe de nombreux recouvrements entre les classes, c'est-à-dire, des méthodes qui peuvent être classifiées dans deux ou plusieurs classes. Nous considérerons que ces méthodes sont des combinaisons des approches existantes.

C'est à partir des définitions des environnements et des cibles pour chaque problématique de la navigation évoquée précédemment, qu'il est possible de s'apercevoir, que certaines méthodes, comme le suivi de blobs, seront difficilement utilisables. Par ailleurs, il est très difficile d'utiliser des méthodes fondées sur la différence objet/fond, parce que le robot est en mouvement et donc, le fond ou l'arrière plan n'est pas statique (du point de vue de l'image), et même les cibles peuvent être statiques par rapport au fond.

De manière similaire, les méthodes fondées sur des mesures de vitesse, seront difficilement exploitables pour la navigation de robots mobiles. Il existe quelques approches pour le suivi à partir de flux optique, qui ont été essayées pour la navigation d'un robot [Dao03] ; mais, la plupart d'entre elles utilisent une méthode de suivi de caractéristiques comme des lignes droites, et c'est à partir de ces primitives éparses dans l'image, que le calcul de la vitesse est fait.

Nous favorisons donc, l'utilisation des deux autres sortes de méthodes afin de réaliser les tâches de suivi depuis un robot se déplaçant dans les environnements d'intérieur : suivi fondé sur un modèle de la cible et suivi de primitives image. Une analyse des méthodes de suivi ainsi que des méthodes de segmentation et reconnaissance des cibles pour la navigation robotique est décrite dans [Des03]. Nous

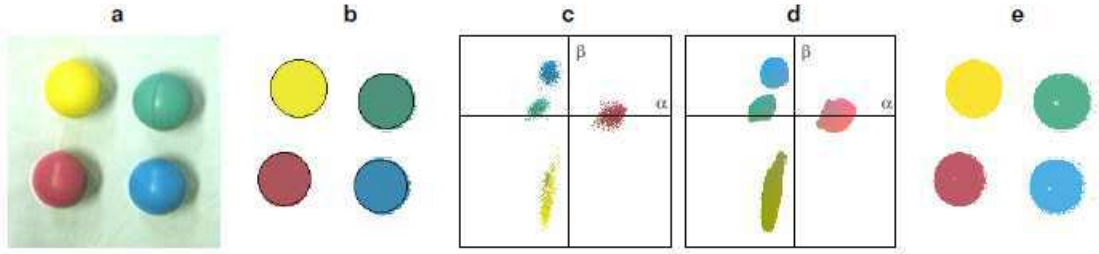

FIGURE 5.1. Calibrage couleur : (a) image couleur d'entrée, (b) segmentation d'image par l'algorithme mean-shift et détermination des cercles, (c) distribution non paramétrique de couleurs mesurée, (d) cinq groupes de couleurs distincts, (e) classification pixel par pixel résultante.

proposons dans ce qui suit une approche robuste qui peut être classée à la fois sous les deux classes retenues, à savoir le suivi fondé sur un modèle de la cible et le suivi de primitives d'objets. Notre approche se déroule en deux étapes ; une première étape hors ligne pour l'apprentissage du modèle d'objet (couleur et forme) et une deuxième étape qui permet d'estimer les paramètres de l'objet pour le suivi temps réel.

5.4. Proposition d'une approche pour la détection et le suivi de balle unicolore

Dans cette section nous introduisons notre nouvelle approche pour le suivi temps réel d'une balle colorée. L'idée clef qui permet de trouver un compromis entre la robustesse et la rapidité du traitement est la détection de la couleur basée sur une segmentation couleur rapide qui produit un nombre beaucoup plus réduit de pixels de contours par rapport à des approches standard basées sur la luminance. Cette réduction diminue considérablement le nombre de votes requis pour une détection robuste en temps réel des paramètres du cercle, même dans le cas où de nombreuses balles de couleur sont présentes dans l'image.

L'approche consiste en deux phases principales. Dans la première phase de calibrage, qui s'effectue hors ligne, les paramètres intrinsèques de la caméra ainsi que la distorsion radiale sont estimés, et une simple classification de couleurs est apprise à partir d'un exemplaire d'une image de balles colorées. Ensuite, dans la phase de suivi en ligne et temps réel, la classification des couleurs est appliquée aux images en entrée, la balle est détectée et sa position 3D est retournée. Dans la suite, nous expliquons en détails ces différentes étapes.

5.4.1. Calibrage

La phase de calibrage hors ligne s'effectue en deux étapes. En premier temps, les paramètres intrinsèques et la distorsion radiale de la caméra sont estimés à partir de plusieurs images d'un damier prises à partir de positions et d'orientations différentes. Nous exploitons l'outil de calibrage de caméra GML [Vez05] pour l'analyse et l'optimisation de l'image. Les paramètres de la caméra extraits sont utilisés immédiatement pour pré-calculer une table de correspondances qui permet de déterminer les positions des pixels lors de l'exécution.

Dans la deuxième étape, nous estimons la distribution de couleur de la balle qui doit être reconnue lors de la phase du suivi en ligne. A cette fin, nous prenons une image couleur de la balle à partir de la caméra (voir Figure 5.1a). Ensuite, nous effectuons une segmentation couleur de l'image en utilisant une version modifiée de l'algorithme mean-shift [Com97]. Dans notre cas, seule la consistance des couleurs des régions est importante, raison pour laquelle l'algorithme des mean-shift ne tient pas compte de la luminance et utilise seulement les composantes couleurs de l'image (espace couleur LUV [Wys82]). Cette modification, fait acquérir à l'algorithme de segmentation une robustesse face aux variations de luminosité tout au long de la surface de la balle (voir Figure 5.1b).

Une fois l'image est segmentée, nous effectuons une analyse en composantes connexes [Ros82] pour étiqueter les différentes régions. Dans chaque région suffisamment large, un algorithme (voir section 5.4.2.2) permet de déterminer si la région est un cercle ou non. Il permet de générer un rapport d'ajustement qui est proche de 1 dans le cas où la région est de forme circulaire. Dans ce cas, la région est reconnue en tant que balle. Ensuite, les pixels détectés à l'intérieur du cercle vont servir d'échantillons de couleur (voir Figure 5.1c) pour la distribution non paramétrique représentée par une image $2D$, où les lignes et les colonnes représentent les composantes de couleur α et β dans un espace de couleur sélectionné. On pourra utiliser n'importe quel espace de couleur, il est envisageable par exemple d'utiliser a et b de l'espace de couleur *CIE Lab*.

Quand tous les échantillons de couleur sont rassemblés, une opération de fermeture est appliquée sur l'image pour remplir les petits trous et filtrer le bruit (voir Figure 5.1d). Finalement, la distribution non paramétrique de couleurs résultantes est utilisée pour calculer une classification RGB qui permet de convertir une couleur d'entrée en indexe unique. La classification est implémentée comme étant une table $3D$ de correspondance qui convertit les triplets RGB en des valeurs entières. Pour éviter une consommation abusive de la mémoire, une représentation sur 6 bits est

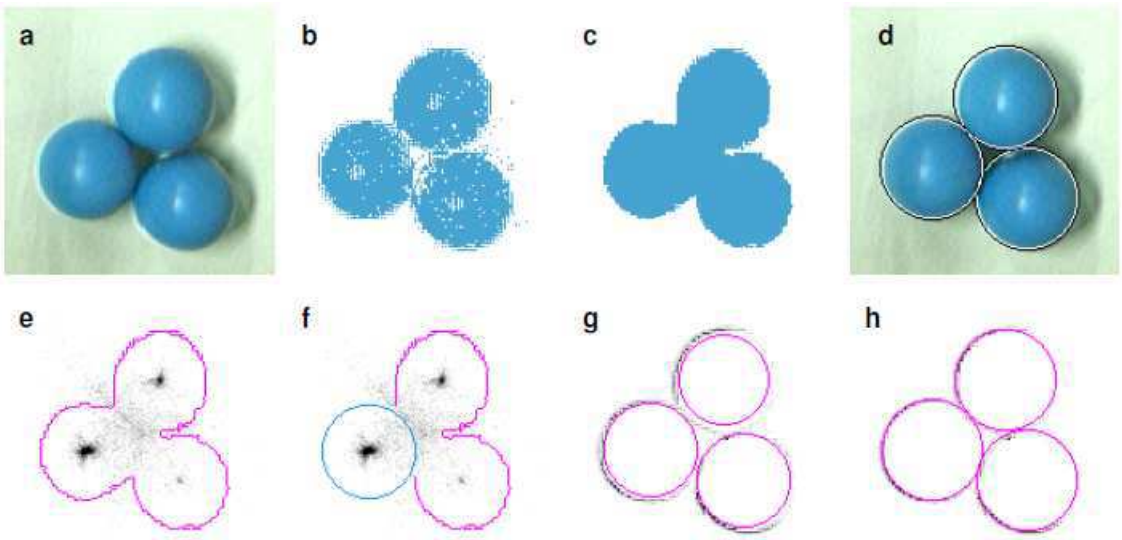

FIGURE 5.2. Détection de balle : (a) image couleur d'entrée, (b) classification pixel par pixel, (c) réduction de bruit et remplissage de trous, (d) détection des balles, (e) contours des régions et l'histogramme du centre, (f) premier cercles détectés et élimination d'une partie du contour d'origine, (g) Tous les cercles détectés et fermeture des gradients de l'image, (h) contours ajustés aux contours réels.

utilisée pour chaque composante de couleur. La Figure 5.1e montre le résultat de la segmentation quand la classification *RGB* est appliquée sur l'image d'entrée.

5.4.2. Suivi temps réel

La phase de suivi temps réel consiste en quatre étapes principales ; (1) segmentation couleur dans laquelle l'image d'entrée est convertie en plusieurs régions, (2) estimation robuste des paramètres du cercle, (3) raffinement des paramètres du cercle, et (4) le suivi de la balle. Dans ce qui suit, chaque étape est décrite avec plus de détails.

5.4.2.1. Segmentation

Une fois l'image est acquise à partir de la caméra (voir Figure 5.2a), on applique la classification RGB pour obtenir un seul indexe couleur par pixel (voir Figure 5.2b). Ensuite, et comme déjà décrit dans la phase de calibrage, une analyse en composantes connexes est effectuée [Ros82] pour obtenir une liste de régions et leurs relations de voisinage. Après, on obtient un ensemble de régions dont l'une d'entre elle au moins représente une balle (voir Figure 5.2c). Pour le reste du traitement, chaque région est représentée par l'ensemble de ses pixels de bord non déformés.

Le problème est maintenant plus simple si on le compare au problème de recherche de cercle dans les images de contours lumineux. En effet, nous allons ef-

fectuer une simple recherche dans un sous ensemble très réduit de l'espace paramétrique 3D. Par contre, une estimation robuste des paramètres du cercle s'avère nécessaire face à des problèmes comme la présence d'autres objets de même couleur, l'occultation, où l'encombrement de l'arrière plan.

Pour la détermination des paramètres du cercle, plusieurs approches ont été utilisées. Citons, les approches qui se basent sur la transformée de Hough, et celles basées sur les moindres carrés. Ces méthodes, quoique robustes, leur convenance pour le temps réel reste discutable dans quelques cas. Nous avons donc, essayé de développer une nouvelle approche dont la première étape consiste à estimer de façon robuste un cercle initial en utilisant un processus de vote aléatoire avec réduction des dimensions de l'espace paramétrique. Ensuite les paramètres du cercle sont raffinés en utilisant une technique par moindres carrés pour un meilleur réajustement aux contours réels de la balle. Une telle combinaison permet d'obtenir un équilibre entre robustesse, précision et rapidité d'exécution. Chaque étape de l'algorithme proposé va être décrite avec détails dans les paragraphes qui suivent.

5.4.2.2. Estimation des paramètres du cercle

Comme indiqué précédemment, l'hypothèse de base est que les limites d'une des régions détectées trace exactement un cercle. Partant de cette hypothèse, si nous prenons aléatoirement plusieurs pixels de bord et nous traçons un cercle qui passe par ces pixels, la probabilité d'avoir des cercles similaires répétitivement serait considérable. Sachant que le calcul de la position du centre d'un cercle (c_x, c_y) à partir de trois points (x_i, y_i) avec des coordonnées entiers, peut se faire rapidement :

$$c_x = \frac{d_1 y_{32} + d_2 y_{13} + d_3 y_{21}}{2(x_1 y_{32} + x_2 y_{13} + x_3 y_{21})} \tag{5.1}$$

$$c_y = \frac{d_1 x_{32} + d_2 x_{13} + d_3 x_{21}}{2(y_1 x_{32} + y_2 x_{13} + y_3 x_{21})} \tag{5.2}$$

Où $x_{ij} = x_i - x_j$ et $y_{ij} = y_i - y_j$ et $d_i = x_i^2 + y_i^2$. Par conséquence, nous pouvons prendre aléatoirement plusieurs votes et recueillir les solutions dans un histogramme 2D avec la même résolution que l'image originale (voir Figure 5.2e). Le processus de vote est arrêté quand l'accumulateur de l'entrée incrémentée dépasse une limite de votes ou quand on dépasse un total bien déterminé de votes. Ce dernier cas est rare et survient lorsque les régions de deux ou plusieurs balles interfèrent dans l'image (voir Figure 5.2e,f). Une fois le cercle est estimé, nous procédons à un simple vote déterministe pour calculer le rayon, nous calculons alors la distance

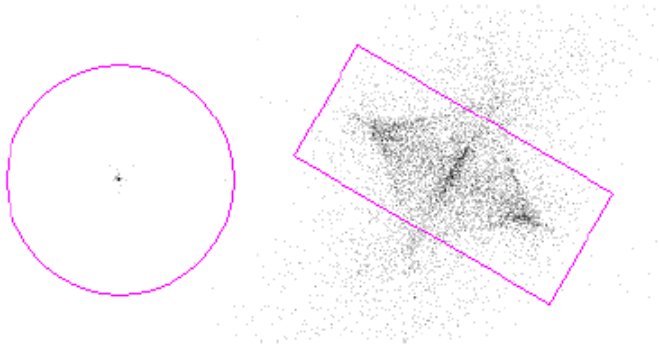

FIGURE 5.3. Histogramme de vote pour le centre d'un cercle : histogramme avec un pic franc pour une région circulaire (gauche) ; l'histogramme est bruité dans le cas où la région n'est pas circulaire et (droite)

(en pixels) de chaque pixel de contour vers le centre du cercle et on incrémente au même temps l'accumulateur de l'entrée correspondante dans un histogramme 1D. Finalement, on retient l'indexe de l'entrée qui a eu le plus de votes comme une estimation du rayon du cercle.

Un problème qui peut survenir est l'existence d'un autre objet non circulaire de même couleur que la balle dans l'image. Dans ce cas, la phase de vote finira par dépasser un nombre maximum de votes aléatoires, et l'histogramme de votes sera bruité sans aucun pic signifiant et le maximum global sera considérablement faible (voir Figure 5.3). Pour reconnaître cette situation on propose une métrique robuste qui témoigne de la qualité du cercle :

$$Q_c = \frac{c_{max} r_{max}}{N_{votes} N_{points}} \tag{5.3}$$

Ici c_{max} et r_{max} sont les maxima globaux des histogrammes du centre et du rayon du cercle. N_{votes} est le nombre de votes du centre du cercle, et N_{points} est le nombre des pixels de contour de la région. Typiquement Q_c pour les régions quasi circulaires est quatre fois plus grande que celle pour les régions non circulaires. La situation la plus douteuse est lorsque il y'a interférence de la balle avec une région de même couleur, cependant dans ce cas aussi l'amplitude de Q_c est deux fois plus grande (voir Figure 5.2c).

5.4.2.3. Raffinement des paramètres du cercle

Une fois l'estimation du cercle est effectuée, nous procédons à un raffinement de son centre et de son rayon pour les faire correspondre au mieux au contour et au centre réels de la balle. Pour simplifier cette tâche, nous assumons que les pixels de

contour de la balle ont un gradient d'intensité relativement élevé et sont au même temps près du cercle estimé. Selon cette hypothèse de base, on applique la méthode des différences centrées sur un petit anneau autour du cercle estimé (voir Figure 5.2g) et on sélectionne les pixels avec les réponses les plus grandes. Généralement, cette approche ne garantit pas qu'on sélectionne toujours le contour réel du cercle puisque des problèmes comme l'occultation ou l'encombrement du fond de l'image peuvent violer notre hypothèse de base. Toutefois, la forme de l'anneau généralement restreint la surface de sélection de façon à ce que les valeurs marginales n'affectent pas de façon sensible la solution finale.

Nous formulons la tâche comme étant une optimisation non linéaire de moindres carrés avec la fonction d'énergie suivante :

$$E(c) = \sum_{p} \|\nabla I.C(c)\|^2 \tag{5.4}$$

Où ∇I est le gradient de l'image au pixel p et $C(c)$ est la distance du pixel p au cercle c :

$$C(c) = \sqrt{(p_x - c_x)^2 + (p_y - c_y)^2} - c_r \tag{5.5}$$

Pour minimiser une telle fonction, nous utilisons la méthode itérative de Gauss-Newton [Pre92]. Où dans chaque itération les dérivés du premier ordre de (5.4) sont mis à zéro et la fonction est linéarisée en utilisant un développement de Taylor du premier ordre :

$$\nabla E(c) = 2\sum_{p} \nabla C. \|\nabla I\|^2 .(C + \nabla C.\triangle c) = 0 \tag{5.6}$$

Avec :

$$\nabla C = \left(\frac{1}{d}(c_x - p_x), \frac{1}{d}(c_y - p_y), -1\right) \tag{5.7}$$

et

$$d = \sqrt{(p_x - c_x)^2 + (p_y - c_y)^2} \tag{5.8}$$

A partir de cette équation nous pourrons obtenir facilement le déplacement incrémental des paramètres du cercle dans la notation matricielle :

$$\triangle c = -\left(\nabla C^t.W.\nabla C\right)^{-1}\left(\nabla C^t.W.C\right) \tag{5.9}$$

Où ∇C est une matrice $3 \times N$ de dérivés du premier ordre $3 \times N$ (5.7), $W = diag()\left\|\nabla I\right\|^2$ est une matrice de pondération diagonale $N \times N$, et C un vecteur colonne $1 \times N$ de distances des points au cercle c (5.5).

En pratique, le calcul de (5.9) est très rapide même pour un nombre étendu de pixels, puisque une seule inversion de matrice de taille 3×3 est calculée et une seule itération est nécessaire pour réduire le décalage avec une précision sub-pixel (voir Figure 5.2h). Par ailleurs, nous avons remarqué dans nos expériences que ces itérations peuvent être coûteuses dans le cas où des valeurs marginales ou un fond encombré sont enregistrés dans l'image originale, c'est pourquoi nous suggérons d'effectuer une seule itération même si le coût en temps d'exécution n'est pas si élevé.

Une fois, les paramètres du cercle c_x, $c_y$et c_z sont déterminés, la position 3D du centre peut être estimée en utilisant un modèle de perspective :

$$x = c_x\frac{z}{f},\ \ y = c_y\frac{z}{f},\ \ z = r\sqrt{1+\frac{f^2}{c_r^2}} \tag{5.10}$$

Où r est le rayon de la balle réelle et f est la distance focale de la caméra.

5.4.2.4. Suivi de la balle

Pour suivre les instances de la balle de manière cohérente et pour éviter les échecs de détection à court terme causés par les occultations de la balle ou par des changements brusques de luminosité, nous essayons de faire fonctionner cette approche à une fréquence haute et nous enregistrons à chaque fois la dernière position de la balle pour déterminer la nouvelle position en cas de confusion avec un autre objet de même classe de couleur et de même forme.

5.5. Résultats expérimentaux

Dans cette section nous essayons de décrire par des expériences la performance et la précision de notre algorithme. Nous présentons aussi les limites révélées par l'expérimentation. Enfin, nous essayons de montrer les résultats donnés par cette approche sur le simulateur du robot Pioneer 3.

5.5.1. Performance et précision

Le principal avantage de l'algorithme proposé par rapport aux approches précédentes est le temps d'exécution global. En moyenne, cet algorithme prend 2ms pour analyser une image de 0.7 Mpix avec une balle rouge en utilisant un ordinateur double coeur (2.3GHz, 1GHz, FSB, 2MB Cache).

La phase d'apprentissage hors ligne, qui renferme la classification des pixels et l'analyse en composantes connexes, consomme en moyenne 6ms par image et peut être considérée comme un chargement constant. L'estimation de la position de la balle a l'influence la plus importante sur la vitesse de traitement. Elle prend en moyenne 0.5ms pour une image dont la région de la balle est modérément grande et avec un seuil de 16 pour l'accumulateur du centre du cercle. Pour atteindre cette valeur, il faut en moyenne effectuer environ 300 votes aléatoires pour le centre du cercle.

La précision de l'algorithme dépend principalement de la résolution de la caméra utilisée, des conditions d'éclairage et du degré d'occultation de la balle. Dans nos expérimentations, nous avons utilisé une caméra avec une résolution de 640x480. Grâce à la phase de raffinement des paramètres du cercle, la déviation entre la balle estimée et la balle réelle ne dépasse typiquement pas le 1mm. Cette valeur est valable pour une balle non occultée de 7cm de diamètre éclairé avec une lumière du jour, positionnée à une distance de 1m de la caméra (diamètre de 60 pixels dans l'espace de l'image). Dans le cas d'une occultation importante (plus de 50%) et/ou de mauvaises conditions d'éclairage, la déviation par rapport à la position réelle peut être encore plus grande.

5.5.2. Limites de l'approche

Une première limitation de l'algorithme est la détection d'une forme bien déterminée d'objet qui est le cercle. Mise à part la non généricité de l'approche, la principale limitation de l'algorithme proposé est qu'il repose sur l'hypothèse qui stipule que la lumière entrante ainsi que le spectre de couleurs sont constants. Cette limitation est étroitement liée au nombre de couleurs distinctes qui devraient être reconnues. Il y'a un compromis entre le nombre de couleurs actives et la taille des classes de couleurs. Une grande taille de la classe apporte plus de robustesse aux changements de couleur mais diminue le nombre de couleurs distinctes et augmente la probabilité de collision avec des objets de fond. D'après nos expériences 4 classes de couleurs distinctes permettent un bon compromis entre le nombre de couleurs et la robustesse du système. Si la lumière du jour est utilisée pour l'éclairage

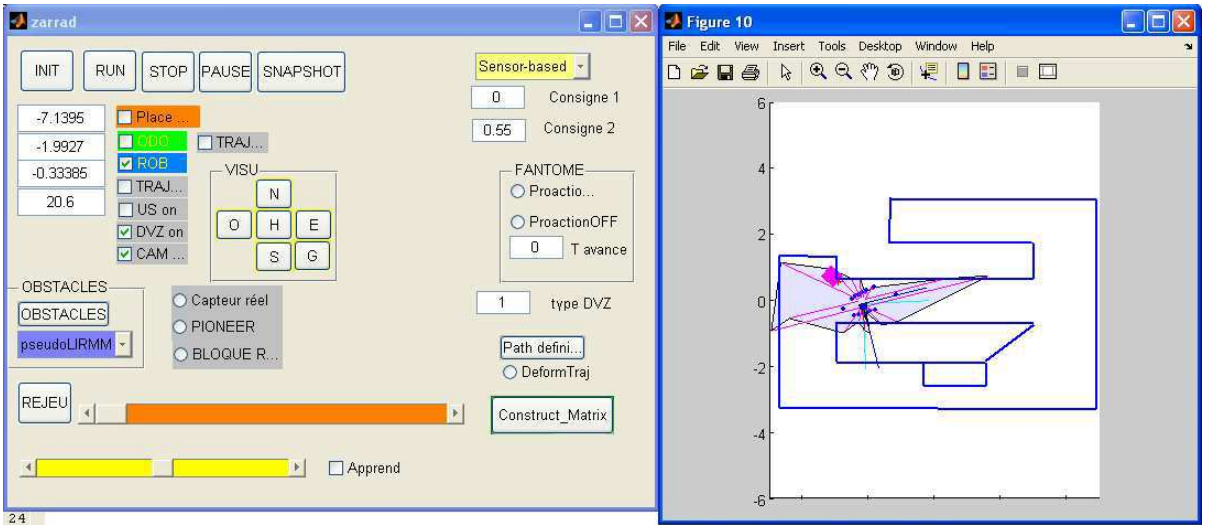

FIGURE 5.4. Simulateur du robot mobile dans l'environnement du laboratoire LIRMM

et plusieurs couleurs sont exigées pour l'apprentissage, il faut exécuter le calibrage à plusieurs reprises dans différents moments de la journée.

Une autre limitation de cette approche est la précision des paramètres estimés de la balle. Quand la balle est très loin de la caméra ou fortement occultée, l'estimation de la position du cercle et du rayon peut être erronée et la profondeur qui en résulte peut être notablement différente de la profondeur réelle. Une situation similaire se produit également, lorsque des mauvaises conditions d'éclairage et/ou des changements dans le spectre de la lumière génèrent une classification bruitée de couleurs.

5.5.3. Simulateur (environnement virtuel d'intérieur)

Pour montrer que notre système de suivi de balle couleur répond aux besoins des applications temps réel, nous avons utilisé un simulateur pour des environnements d'intérieur du robot mobile Pioneer 3. Dans la figure 5.4, nous montrons les résultats de la simulation : le robot Pioneer 3 portant une Webcam et une balle rouge exposée devant la caméra. Nous utilisons une seule balle, donc seulement deux classes de couleurs sont à définir ; celle des pixels de la balle et celle des pixels qui n'appartiennent pas à la balle.

La Figure 5.5 montre les étapes de traitement de l'image pour la détection de la balle. La figure 5.5a montre l'image de la balle prise par la caméra avec des objets de couleur semblable à la balle. L'image 5.5e montre la transformation de la balle en image en noir et blanc selon l'apprentissage de la couleur de la balle (l'étape qui est effectuée hors ligne au début). Les images 5.5c et 5.5d sont obtenues après

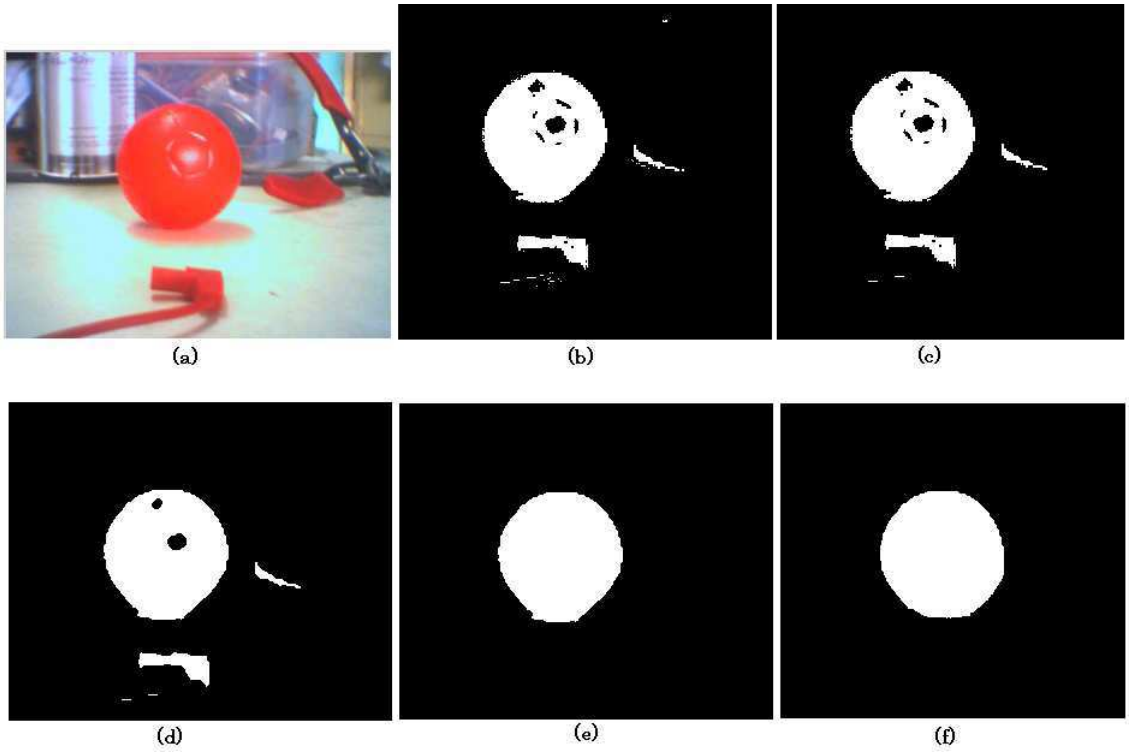

FIGURE 5.5. Détection de la balle : (a) image prise par la webcam ; (b) image après classification en deux classes de couleur ; (c) image après ouverture ; (d) image après fermeture ; (e) estimation des paramètres du cercle ; (f) raffinement du paramètres du cercle.

ouverture, fermeture et élimination du bruit sur l'image 5.5b. L'image 5.5e montre la première estimation des paramètres du cercle de la balle. L'image 5.5f montre le cercle de l'image après raffinement des paramètres par la méthode présentée dans la section 5.5.2.3.

Le robot Pioneer 3 est équipé d'un module d'évitement d'obstacles en temps réel en utilisant ses capteurs ultrasons. L'intégration du module de suivi de la balle n'a pas affecté l'aspect temps réel et le robot parvient à éviter les obstacles tout en suivant la balle comme le montre la Figure 5.6.

5.6. Conclusion

Dans ce chapitre, nous avons présenté les aspects et les caractéristiques de la vision active. Nous avons ensuite, donné une classification des méthodes de suivi d'objets selon les techniques et les approches utilisées. Enfin nous avons proposée une approche hybride pour le suivi d'une balle colorée en temps réel. Cette approche repose sur deux phases ; une phase d'apprentissage hors ligne qui permet de donner une classification de couleurs selon les couleurs des cibles ; et une phase

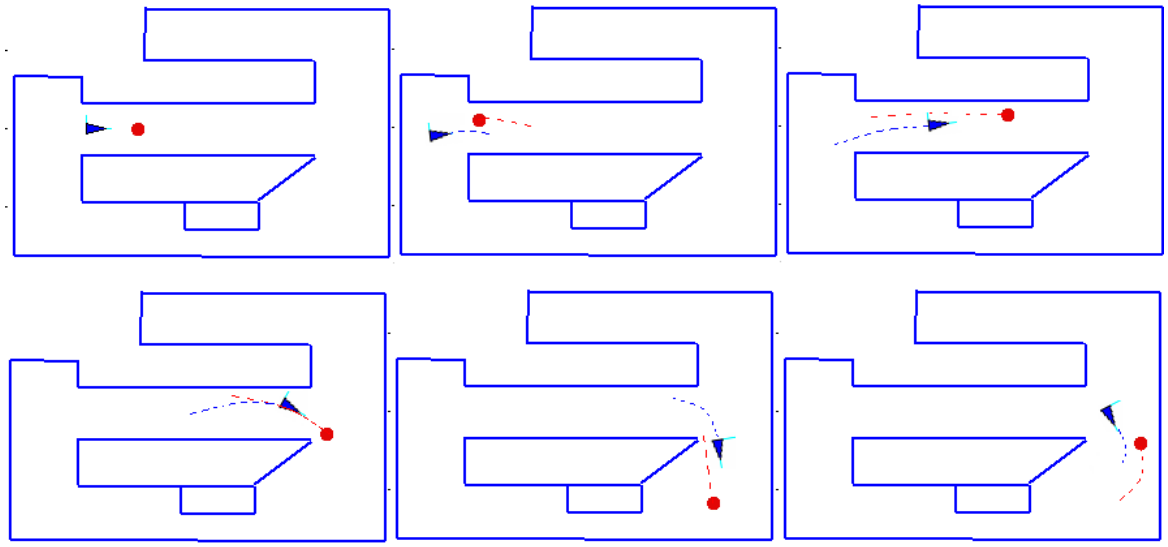

FIGURE 5.6. Séquence de suivi de balle rouge par le robot mobile Pioneer 3

en ligne qui permet de détecter une ou plusieurs balles et donner avec une précision sub-pixel l'estimation des cercles correspondant. Les expériences effectuées ont montré que cette approche convient pour les applications temps réel et convient même d'être embarquée avec d'autres modules temps réel sans autant affecter le temps d'exécution.

Chapitre 6

Conclusion et perspectives

Dans ce manuscrit, nous avons présenté, les travaux développés sur les applications de la vision à la navigation d'un robot mobile. Nos contributions, concernent l'utilisation de fonctionnalités visuelles dans le système embarqué à bord d'un robot mobile. Les travaux en ce domaine se focalisent généralement sur un type d'environnement ; pour notre part, nous avons analysé des applications de nos travaux sur la vision dans un environnement d'intérieur. Nous nous sommes d'abord intéressé à la méthode d'acquisition des informations visuelles et de la restitution de l'information 3D. Nous avons opté pour la vision stéréoscopique pour les raisons déjà évoquées dans le deuxième chapitre.

La caractéristique commune à ces travaux est la nécessité d'intégrer dans la couche fonctionnelle du système embarqué sur le robot, plusieurs fonctionnalités visuelles, ayant des temps de réponse différents ;

- les unes réalisent une analyse globale des images acquises, et génèrent une carte 3D dense et dont le temps de réponse est de l'ordre de 0.5s environ.
- la construction de carte et sa mise à jour à une fréquence de 5Hz.
- les autres réalisent le suivi visuel d'une balle mobile colorée. La fréquence de traitement est typiquement de 10Hz.

Nous avons présenté dans cette thèse, notre contribution sur ces fonctionnalités visuelles. En sus d'être rapide pour assurer une bonne réactivité du système, la restitution des informations 3D par la stéréovision doit aussi être robuste. Ces deux contraintes ne peuvent être satisfaites que par l'intégration des contraintes globales de mise en correspondance dans les méthodes locales rapides sans autant affecter le temps de réponse. La modélisation de l'environnement doit être fidèle à la réalité. Cette caractéristique ne peut être obtenue sans modéliser l'incertitude et les erreurs de mesures. Enfin, concernant le suivi d'objets, la détection doit être robuste et au même temps rapide, afin de minimiser les ruptures de suivi.

Dans le deuxième chapitre, nous avons présenté les méthodes utilisées pour la navigation dans un milieu d'intérieur, nous avons décrit aussi les différentes ap-

proches pour modéliser l'environnement. Suivant, les approches explorées, nous avons pu fixer notre choix sur la vision stéréoscopique comme méthode de perception (de restitution des informations tridimensionnelles) et sur les grilles d'occupation comme méthode de modélisation 3D de l'environnement, utilisant les informations 3D issues de la stéréovision.

Dans le troisième chapitre, nous nous sommes concentrés sur la vision stéréoscopique. Nous avons présenté le principe de la stéréovision, ensuite nous avons exploré les travaux effectués concernant les méthodes globales et les méthodes locales de mise en correspondance stéréoscopique. De cela, nous a venu l'idée d'utiliser les contraintes stéréoscopiques des méthodes globales en vue d'accorder plus de robustesse aux méthodes locales. Nous avons alors défini des distributions de possibilités qui s'appuient sur une modélisation sémantique des contraintes stéréoscopiques basée sur une graduation des niveaux de gris des pixels via les sous-ensembles flous. Une estimation initiale des disparités, basée sur les distributions de possibilités définies, est utilisée pour déterminer des disparités finales plus fiables. L'approche que nous avons proposée dans ce chapitre nous à permis de gagner en précision par rapport aux méthodes globales tout en restant dans la même marge de temps des approches locales. Nous avons présenté dans cette thèse, notre contribution sur ces fonctionnalités visuelles. En sus d'être rapide pour assurer une bonne réactivité du système, la restitution des informations 3D par la stéréovision doit aussi être robuste. Ces deux contraintes ne peuvent être satisfaites que par l'intégration des contraintes globales de mise en correspondance dans les méthodes locales rapides sans autant affecter le temps de réponse. La modélisation de l'environnement doit être fidèle à la réalité. Cette caractéristique ne peut être obtenue sans modéliser l'incertitude et les erreurs de mesures. Enfin, concernant le suivi d'objets, la détection doit être robuste et au même temps rapide, afin de minimiser les ruptures de suivi.

Dans le quatrième chapitre, nous nous sommes attaqué au problème de modélisation de l'environnement. Les grilles d'occupation ont été fixées comme méthodes de modélisation. Nous avons donc présenté l'état de l'art sur les travaux les plus réputés utilisant les grilles d'occupation. Ensuite, nous avons présenté notre approche. Une représentation 3D intrinsèquement incertaine basée sur la propagation de l'erreur stéréoscopique a été utilisée dans cette approche. Une nouvelle méthode de mise à jour, basée sur des valeurs de crédibilité, a été employée pour mettre à jour le modèle de l'environnement.

Dans le cinquième chapitre, nous avons donné une classification des différentes méthodes de suivi d'objets. Ensuite, nous avons présenté une nouvelle technique

pour le suivi de balle colorée, adaptée aux systèmes robotiques autonomes temps réel. En utilisant une seule caméra, on peut estimer la position 3D d'une ou plusieurs balles colorées en temps réel. Cette technique repose sur deux étapes. La première étape est un calibrage hors ligne pour l'apprentissage de la couleur de la balle. La deuxième phase, en ligne, est la détection et l'estimation des paramètres du cercle de la balle. La technique proposée est assez rapide et peut être facilement combinée aux modules de perception et modélisation de l'environnement.

Enfin, nous souhaitons terminer en évoquant des travaux que nous pourrions mener dans le futur sur la thématique présentée dans cette thèse. Tout d'abord, de nombreuses améliorations devraient être apportées aux modules logiciels que nous avons développés, notamment :

- Développer une bibliothèque offrant des API's paramétrables pour calculer les distributions de possibilités et pouvant être utilisées dans le calcul des correspondances stéréoscopiques.
- Pour la méthode de suivi d'objets : modifier la méthode de façon à la rendre générique pour suivre différents types de cibles. Augmentation de la base de cibles prises en compte afin d'ajuster les paramètres de la sélection de la méthode de suivi visuel, incorporation de nouveaux contextes, amélioration de la méthode de suivi par corrélation 1D..., incorporation d'autres mesures de complexité, concernant par exemple la structure de la scène, ou l'intérieur de la cible, utiliser des mécanismes pour commander les paramètres de la caméra tel que l'orientation, le zoom...

A plus long terme, plusieurs études pourraient être poursuivies, notamment sur les points suivants :

- Utiliser un système de deux robots formé par un robot suiveur et un robot cible. Les deux robots seront dotés d'un module de perception 3D comme décrit dans le troisième chapitre et d'un module de modélisation de l'environnement décrit dans le quatrième chapitre. Le premier robot va suivre la balle déposée sur le second robot, et les deux robots vont essayer de modéliser de façon coopérative leur environnement par fusion de cartes.
- Développer un module pour l'estimation en temps réel de la trajectoire d'une cible en trois dimensions basé sur la vision stéréoscopique. La cible est toujours une balle colorée, qui serait segmentée par soustraction du fond, et le ballon serait détecté avec la même méthode que celle décrite dans le cinquième chapitre. Le suivi de la balle serait réalisé par un filtrage de Kalman tridimensionnel, ce qui permettra d'améliorer la détection avec une recherche

locale basée sur la prédiction de la position 3D et de filtrer la trajectoire pour réduire l'erreur d'estimation.

Bibliographie

[Alo92] Y. Aloimonos, "Purposive active vision," in *CVGIP : Image Understanding*, vol. 56, No. 1, pp. 840-850, August 1992.

[Asc93] P. Aschwanden et W. Guggenbuhl. "Experimental results from a compa- rative study on correlation-type registration algorithms." *Robust Computer Vision*, pp. 268–289, 1993.

[Aya89] N. Ayache et O. D. Faugeras, "Maintaining Representations of the Environment of a Mobile Robot", *IEEE Transactions Robotics and Automation*, vol. 5, no. 6, pp. 804-819, 1989.

[Bai02] T. Bailey. "Mobile robot localisation and mapping in extensive outdoor environments". *PhD thesis*, Australian Center for Field Robotics, University of Sydney, Sydney (Australie), août 2002.

[Baj88] R. Bajcsy, "Active Perception", in Proceedings of the IEEE, Special issue on Computer Vision, Vol. 76, No. 8, August 1988.

[Bak81] H. H. Baker and T. O. Binford, "Depth from edge and intensity based stereo. *7th International Joint Conference on Artificial Intelligence*, pp. 631–636, 1981.

[Ban01] J. Banks et P. Corke. Quantitative evaluation of matching methods and validity measures sor stereo vision. Int'l J. Robotics Research, 20(7) :512– 532, 2001.

[Bas94] B. Bascle, P. Bouthemy, R. Deriche, et F. Meyer, "Tracking complex primitives in an image sequence", *Technical Report 2428, INRIA, Sophia-Antipolis*, France, December 1994.

[Bek07] T. Bekaert, S. Gautama, R. Goossens, and W. Philips, Dense and reliable stereo matching in urban areas using adaptive windows, *18th Annual Workshop on Circuits, Systems and Signal Processing* . Veldhoven, Nov 2007.

[Bel96] P. N. Belhumeur, "A bayesian approach to binocular stereopsis." *International Journal of Computer Vision*, 19(3) :237–262, 1996.

[Bet96a] S. Betgé-Brezetz, P. Hebert, R. Chatila et M. Devy. "Uncertain map making in natural environments." *In IEEE International Conference on Robotics and Automation (ICRA'96)*, Minneapolis (USA), pp. 1048-1053, April 1996.

[Bet96b] S. Betgé-Brezetz. "Modélisation incrémentale et localisation par amers pour la navigation d'un robot mobile autonome en environnement naturel." *Thèse de doctorat, Université Paul Sabatier, LAAS/CNRS, Toulouse (France)*, Février 1996.

[Bha98] D. N. Bhat et S. K. Nayar. "Ordinal measures for visual correspondence." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 :415– 423, 1998.

[Big96] F. Bigone, O. Henricsson, et P. Fua. "Automatic extraction of generic house roofs from high resolution aerial imagery." *Proc. European Conf. Computer Vision*, pp. 85–96, 1996.

[Bir98] S. Birchfield et C. Tomasi. "A pixel dissimilarity measure that is intensitive to image sampling." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 20(4) :401–406, 1998.

[Bla92] A. Blake, et A. Yuille (editors), "Active Vision", *MIT press*, November, 1992.

[Bla98] A. Blake, and M. Isard, "Active Contours", *Springer-Verlag*, London, 1998.

[Blo87] D. S. Blostein, et S. T. Huang, "Error analysis in stereo determination of 3D point positions", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9(6), 752-765, 1987.

[Bob99] A. F. Bobick et S. S. Intille. "Large occlusion stereo." *International Journal on Computer Vision (IJCV)*, 33(3) :181–200, 1999.

[Bol93] R. C. Bolles, H. H. Baker, et M. J. Hannah, "The JISCT stereo evaluation." *In Image Understanding Workshop*, pp. 263–274, Morgan Kaufmann Publishers, 1993.

[Bor89] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots", *IEEE Transactions Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179-1187, 1989.

[Bor90] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments", *in Proceedings of IEEE International Conference in Robotics and Automation*, pp. 572-577, 1990.

[Bor91] J. Borenstein and Y. Koren, "The Vector Field Histogram–Fast Obstacle Avoidance for Mobile Robots", *IEEE Transactions Robotics and Automation*, vol. 7, no. 3, pp. 278-288, June 1991.

[Bor96] J. Borenstein, H. R. Everett and L. Feng, "Navigating Mobile Robots : Systems and Techniques", *eds. Wellesley, Mass. :* AK Peters, 1996.

[Borg02] G. A. Borges and M.-J. Aldon. Optimal pose estimation using geometrical maps. In IEEE Transactions on Robotics and Automation, février 2002.

[Bos03] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, et S. Teller, "An atlas framework for scalable mapping." *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1899-1906, 2003.

[Boy98] Y. Boykov, O. Veksler, et R. Zabih. “Markov random fields with efficient approximations.” *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 648–655, 1998.

[Boy99] Y. Boykov, O. Veksler, et R. Zabih. “Fast approximate energy minimiza- tion via graph cuts.” *Proceedings of International Conference on Computer Vision*, 1 :377–384, 1999.

[Boy01] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1222-1239, 2001.

[Bue02] C. Buehler, S. Gortler, M. Cohen, et L. McMillan. “Minimal surfaces for stereo.” *Proceedings of European Conference on Computer Vision*, pp. 885–899, 2002.

[Can86] J. A Canny, “Computational approach to edge detection.” *Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986.

[Cel02] E. Celaya et C. Torras. “Visual Navigation Outdoors : The Argos Project.” *In The 7th International Conference on Intelligent Autonomous Systems (IAS-7)*, pp. 63-67, Marina del Rey, California, USA, March 25-27 2002.

[Cham05] S. Chambon. ”Mise en correspondance stéréoscopique d’images couleur en présence d’occultations”. *Thèse de doctorat*, Université de Toulouse, 2005.

[Cha95] R. Chatila, S. Lacroix, T. Siméon et M. Herrb. “Planetary exploration by a mobile robot : Mission teleprogramming and autonomous navigation.” *Autonomous Robots Journal*, vol. 2, no. 4, pp. 333-344, 1995.

[Che99] Q. Chen et G. Medioni. “A volumetric stereo matching method : Appli- cation to image-based modeling.” *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 29–34, 1999.

[Cho97] K. S. Chong, et L. Kleeman, “Large scale sonarray mapping using multiple connected local maps.” *In International Conference on Field and Service Robotics*, pp. 538-545, ANU, Canberra, Australia, 1997.

[Cle98] M. Clerc. “Wavelet-based correlation for stereopsis.” *Proceedings 7th European Conference on Computer Vision*, 2 :495–509, 2002.

[Com97] D. Comaniciu, P. Meer : Robust analysis of feature spaces, ”Color image segmentation”. *In Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 750–755, (1997).

[Col05] T. Collins, J. J. Collins, S. O’Sullivan, et M. Mans eld, ”Evaluating techniques for resolving redundant information and specularity in occupancy grids”, *Advances in Artificial Intelligence* , pp. 235-244, 2005.

[Col07] T. Collins, J. J. Collins, C. Ryan, "Occupancy Grid Mapping : An Empirical Evaluation", *Proceedings of the 15th Mediterranean Conference on Control and Automation*, July 27-29, Athens, Greece, 2007.

[Cox92] I. J. Cox, S. L. Hingorani, B. M. Maggs, et S. B. Rao. "Stereo without disparity gradient smoothing : a bayesian sensor fusion solution." *Proceedings of British Machine Vision Conference*, pp. 337–346, 1992.

[Cox94] I. J. Cox, "Modeling a Dynamic Environment Using a Bayesian Multiple Hypothesis Approach", *Artificial Intelligence*, vol. 66, no. 2, pp. 311-344, April 1994.

[Cox96] I. J. Cox, S. L. Hingorani, et S. B. Rao. "A maximun likehood stereo algorithm." *Computer Vision and Image Understanding*, 63(3) :542–567, 1996.

[Dao03] N. X. Dao, BJ.You, SR.Oh and M. Hwangbo, "Visual Self-Localization for Indoor Mobile Robots Using Natural Lines", *in Proc. of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems IROS'2003*, pp. 1252-1257, Las Vegas, Nevada, October, 2003.

[Del98] F. Dellaert, C. Thorpe, and S. Thrun, "Super-Resolved Texture Tracking of Planar Surface Patches", in Proc. of IEEE/RSJ International Conference on Intelligent Robotic Systems, October, 1998.

[Der90] R. Deriche et O. Faugeras, "2d curve matching using high curvature points." *Proceedings of International Conference on Pattern Recognition*, pp. 240–242, June 1990.

[Des02] G. N. DeSouza and A. C. Kak, "Vision for Mobile Robot Navigation : A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237-267, February 2002.

[Dev97] A. Dev, B. J. A. Krose et F.C.A. Groen, "Navigation of a Mobile Robot on the Temporal Development of the Optic Flow", *in Proceedings of IEEE International Conference in Intelligent Robots and Systems,* pp. 558-563, September 1997.

[Deve97] F. Devernay, "Vision stéréoscopiques et propriétés différentielles des surfaces." *Thèse de doctorat : Ecole polytechnique*, 190p, 1997.

[Dho89] U. R. Dhond et J. K. Aggarwal. "Structure from stereo - a review." *IEEE Trans ; Systems, Man and Cybernetics,* 19 :1489–1510, 1989.

[Don11] A. Donate, "Efficient Path-Based Stereo Matching With Subpixel Accuracy", *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, Vol. 41, No. 1, pp. 183-195, 2011.

[Dru02] T. Drummond, et R. Cipolla, "Real-time tracking of complex structures with online camera calibration", in Image and Vision Computing, vol. 20, No. 5-6, pp. 427-433, 2002.

[Duc00] T. Duckett, et A. Saffiotti, “Building globally consistent gridmaps from topologies.” *In Proceedings of the International Proc. IFAC Symposium on Robot Control*, pp. 357-361, Wien, Austria, 2000.

[Duf05] D. Dufourd, ”Des Cartes Combinatoires Pour La Construction Automatique De Modèles D’Environnement Par Un Robot Mobile”. *Thèse de doctorat, Institut National Polytechnique de Toulouse*. 2.4.2. (2005).

[End12] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, St. Paul, MA, USA, May 2012.

[Elf86] A. Elfes ”A sonar based mapping and navigation system”, *In Proc. of IEEE Intrl. Conf. on Robotics and Automation (ICRA’86)*, 1986.

[Elf92] 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.

[Eyn10] D. Eynard, P. Vasseur, C. Demonceaux, V. Fremont, ”Estimation temps reel de l’altitude d’un drone a partir d’un capteur de stereovision mixte”, *17ème congrès francophone AFRIF-AFIA RFIA*, France, 2010.

[Fab02] E. Fabrizi, et A. Saffiotti, ”Augmenting topology-based maps with geometric information.” *Robotics and Autonomous Systems*, 40(23) :9197, 2002.

[Fal12] M. F. Fallon, H. Johannsson, and J. J. Leonard. Efficient scene simulation for robust Monte Carlo localization using an RGB-D camera. *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, May 2012.

[Far08] I. R. Farah, W. Boulila, B. K. Saheb Ettabaâ, B. Solaiman et M. B. Ahmed, "Interpretation of multisensor remote sensing images : Multi-approach fusion of uncertain information". *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 46, No. 12, pp 4142-4152, December 2008.

[Fau93a] O. Faugeras, “Three-Dimentional Computer Vision, a Geometric Viewpoint.” *MIT Press*, 1993.

[Fau93b] O. Faugeras, B. Hotz, Z.Zhan, et al. “Real time correlation-based stereo : Algorithm, implementations and applications.” *Technical Report, INRIA,* 1993.

[Fer95] C. Fernandez-Maloigne. “Texture and neural network for road segmentation.” *In Intelligent Vehicles ’95 Symposium*, pp. 344-349, September 1995.

[Fin03] J. M. Findlay, I. D. Gilchrist, ”Active Vision - The Psychology of Looking and Seeing”, *Oxford University Press*, London, august, 2003.

[For62] L. Ford et D. Fulkerson. “Flows in Networks.” *Princeton University Press*, 1962.

[Fra05] J-S. Franco, E. Boyer, ”Fusion of multi-view silhouette cues using a space occupancy grid”, *In International Conference on Computer Vision*, pp. 1747-1753, 2005.

[Gam12] D. M. Gámez, Michel Devy, "Active visual-based detection and tracking of moving objects from clustering and classification methods", *Advanced Concepts for Intelligent Vision Systems*, Lecture Notes in Computer Science Volume 7517, pp 361-373, 2012.

[Gas99] J. A. Gasós, "Integrating fuzzy geometric maps and topological maps for robot navigation." I*n proc. of the 3rd International Symposium on Soft Computing*, 1999.

[Gau97] P. Gaussier, C. Joulain, S. Zrehen et A. Revel, "Visual Navigation in an Open Environment without Map", *In Proceedings of IEEE International Conference in Intelligent Robots and Systems*, pp. 545-550, September 1997.

[Gei01] D. Geiger, B. Ladendorf, et A. Yuille. "Occlusions and binocular stereo." *International Journal on Computer Vision (IJCV)*, 14 :211–226, 1995.

[Gha07] H. Ghazouani, M. Tagina, "A Behavior-Based Controller with Evolutionary Adapted Fuzzy Rule Base for a Car-like Mobile Robot," *Proceedings of the 2007 International Conference on Artificial Intelligence, ICAI 2007*, Volume II, June 25-28, 2007 : 508-514, Las Vegas, Nevada, USA. CSREA Press 2007, ISBN 1-60132-024-8,2007.

[Gha10] H. Ghazouani , R. Zapata, M. Tagina, "Fuzzy Sets Based Improvement of a Stereo Matching Algorithm with Balanced Correlation Window and Occlusion Detection," *In Proceedings of the 2010 International Conference on Image Processing, Computer Vision, Pattern Recognition, IPCV 2010*, July 12-15, 2010, Las Vegas Nevada, USA.,2010.

[Gha10a] H. Ghazouani, M. Tagina, R. Zapata, A Theory of Possibility for Reliable Correspondence Search, *International Journal of Signal and Image Processing (IJSIP)*, Volume 1, Issue 4, Page(s) : 232-237, ISSN : 1737-9253, HyperSciences_Publisher, July2010.

[Gha10b] H. Ghazouani, M. Tagina, R. Zapata, Evolution-Based Vision Algorithm with Fuzzy Fitness Function for Obstacle Detection, *The 3rd International Conference on Metaheuristics and Nature Inspired Computing, META'10,* Djerba, , Tunisia, 28-31 October, 2010.

[Gha10c] H. Ghazouani, M. Tagina, R. Zapata. "Robot Navigation Map Building Using Stereo Vision Based 3D Occupancy Grid", *Journal of Artificial Intelligence : Theory and Application (JAITA)*. Vol. 1, No. 3, pp. 63-72, 2010, ISSN : 1737-9334, HyperSciences Publisher, 2010.

[Gha11] H. Ghazouani, M. Tagina, R. Zapata, Fast and Robust Semi- Local Stereo Matching Using Possibility distributions, *International of Computational Vision and Robotics (IJCVR)*, Vol.2, No.3, pp.237-253, 2011, DOI : 10.1504/IJCVR.2011.042841

[Gir79] G. Giralt, R. Sobek et R. Chatila. "A Multi-Level Planning and Navigation System for a Mobile Robot ; A First Approach to Hilare." *In 6th International Joint Conference on Artificial Intelligence*, volume 1, pp. 335-337, 1979.

[Gua10] S. Guadarrama, A. Ruiz-Mayor, "Approximate robotic mapping from sonar data by modeling perceptions with antonyms", *Information Sciences : an International Journal*, Volume 180 Issue 21, November, 2010.

[Gui04] J. Guivant, E. Nebot, J. Nieto, and F. Masson. Navigation and mapping in large unstructured environments. In The International Journal of Robotics Research, volume 23 (4-5), pp. 449-472, avril-mai 2004.

[Glu92] V. S. Gluth, G. W. Kunkel, et U. A. Rauhala. "Global least squares matching." *Proceedings of International Geoscience and Remote Sensing Symposium*, 2 :1615–1618, 1992.

[Gri85] W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm." *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-*7(1), 17–34, 1985.

[Gu08] Z. Gu, X. Su, Y. Liu, Q. Zhang. "Local stereo matching with adaptive support-weight, rank transform and disparity calibration," *Pattern Recognition Letters*, v.29 n.9, p.1230-1235, July 2008.

[Had98] H. Al Haddad. "Contrôle par vision du mouvement d'un robot mobile en environnement naturel." T*hèse de doctorat, Université Paul Sabatier (Automatique et Informatique Industrielle), LAAS/CNRS, Toulouse, France*, 5 Novembre 1998.

[Har88] C. Harris et M. Stephens "A combined corner and edge detector." P*roceedings of the 4th ALVEY vision conference*, pp. 147–151, 1988.

[Hen12] P. Henry, M. Krainin, E. Herbst, X. Ren et D. Fox, "RGB-D mapping : Using Kinect-style depth cameras for dense 3D modeling of indoor environments", *The International Journal of Robotics Research*, 2012.

[Hir05] H. Hirschmueller, "Accurate and Efficient Stereo Processing by Semi-global Matching and Mutual Information," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol II :807-814, 2005.

[Hir01] H. Hirschmuller. "Improvements in real-time correlation-based stereovision." *Proceedings of Workshop on Stereo and Multi-Baseline Vision*, pp. 141– 148, 2001.

[Hof89] W. Hoff and A. Narendra, "Surfaces from stereo : integrating feature matching, disparity estimation, and contour detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, pp. 121-136, 1989.

[Hor95] R. Horaud, et O. Monga, "Vision par Ordinateur : outils fondamentaux," *2ème édition. Hermès*, (page 16), 1995.

[Horn95] J. Horn et G. Schmidt, "Continuous Localization of a Mobile Robot Based on 3D-Laser-Range-Data, Predicted Sensor Images, and Dead-Reckoning", *Robotics and Autonomous Systems*, vol. 4, no. 2-3, pp. 99-118, May 1995.

[Hsi92] Y. C. Hsieh, D. McKeown, et F. P. Perlant, "Performance evaluation of scene registration and stereo matching for cartographic feature extraction." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 214–238, 1992.

[Hub95] E. Huber et D. Kortenkamp, "Using Stereo Vision to Pursue Moving Agent with a Mobile Robot," *in Proceedings of IEEE International Conference in Robotics and Automation*, vol. 3, pp. 2340-2346, May 1995.

[Ish98] H. Ishikawa et D. Geiger. "Occlusions, discontinuities and epipolar lines in stereo." *Proceedings of 5th European Conference on Computer Vision*, pp. 232–246, 1998.

[Ito86] M. Ito et A. Ishii. "Three-view stereo analysis". *Pattern Analysis and Machine Intelligence*, pp. 524–532, 1986.

[Jia08] Y. Jianxi, L.Jianting, et S. Zhendong, "Calibrating method and systematic error analysis on binocular 3D position system", *Proceedings of the 6th International Conference on Automation and Logistics*, China, 2310-2314, 2008

[Jou97] C. Joulian, P. Gaussier, A. Revel et B. Gas, "Learning to Build Visual Categories from Perception-Action Associations", *in Proceedings of IEEE International Conference in Intelligent Robots and Systems,* pp. 857-864, September 1997.

[Kan94] T. Kanade et M. Okutomi. "A stereo matching algorithm with an adaptative window : Theory and experiment." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9) :920–932, 1994.

[Kim94] D. Kim and R. Nevatia, "Representation and Computation of the Spatial Environment for Indoor Navigation", *in Proceedings of International Conference in Computer Vision and Pattern Recognition*, pp. 475-482, 1994.

[Kim95] D. Kim and R. Nevatia, "Symbolic Navigation with a Generic Map", *in Proceedings of IEEE Workshop Vision for Robots*, pp. 136-145, August 1995.

[Kim98] D. Kim et R. Nevatia, "Recognition and Localization of Generic Objects for Indoor Navigation Using Functionality", *Image and Vision Computing,* vol. 16, no. 11, pp. 729-743, August 1998.

[Koh10] K. Kohara, N. Suganuma, T. Negishi, et T. Nanri, "Obstacle Detection Based on Occupancy Grid Maps Using Stereovision System", *International Journal of Intelligent*, 2010

[Kol94] D. J. Koller, Weber, et J. Malik, "Robust multiple car tracking with occlusion reasoning" *in Proc. 3rd European Conf. on Computer Vision (ECCV'94)*, Stockholm, vol. 1, pp. 189-196, May, 1994.

[Kol01] V. Kolmogorov et R. Zabih. "Computing visual correspondance with occlusions using graph cuts." *Proceedings of the 8th International Conference on Computer Vision*, pp. 508–515, 2001.

[Kol02a] V. Kolmogorov et R. Zabih. "Multi-camera scene reconstruction via graph cuts." *Proceedings of European Conference on Computer Vision*, pp. 82– 96, 2002.

[Kol02b] V. Kolmogorov et R. Zabih. "What energy functions can be minimized via graph cuts." *Proceedings of the Europeanl Conference on Computer Vision*, pp. 65–81, 2002.

[Kon97] K. Konolige,."Improved occupancy grids for map building", *Autonomous Robots*, no. 4, pp. 351-367, 1997.

[Kwo97] 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.

[Las96] P. Lasserre. "Vision pour la robotique mobile en environnement naturel." *Thèse de doctorat, Université Paul Sabatier, LAAS/CNRS, Toulouse, France,* 26 Septembre 1996.

[Lat10] H. Lategahn, W. Derendarz, T. Graf, B. Kitt, J. Effertz, "Occupancy grid computation from dense stereo and sparse structure and motion points for automotive applications", *2010 IEEE Intelligent Vehicles Symposium (IV)*, 1931-0587, San Diego, CA 21-24 June 2010.

[Lep99] P. Lepinay et R. Zapata. "Realisation d'une carte de correlation visuelle en temp reel." *Technical Report 6578, LIRMM*, 1999.

[Lev73] M. Levine, D. A. O'Handeley, and G. M. Yagi, "Computer determination of depth maps," *Computer Graphics and Image Processing*, vol. 2, pp. 131-150, 1973.

[Lhu02] M. Lhuillier et L. Quan. "Match propagation for image-based modeling and rendering." *IEEE Transactions on Pattern Analysis and Machine In- telligence*, 24(8) :1140–1146, 2002.

[Lor97] L. M. Lorigo, R. A. Brooks et W. E. L. Grimson. "Visually-Guided Obstacle Avoidance in Unstructured Environments." I*n IEEE/RSJ International Conference on Intelligence Robots and Systems*, volume 1, pp. 373-379, Grenoble, France, September 1997.

[Luc81] B. D. Lucas et T. Kanade. "An iterative image registration technique with an application to stereovision." *Int'l Joint Conf. Artificial Intelligence*, pp. 674–679, 1981.

[Luo08] G. Luo, X. Yang, Q. Xu, "Fast Stereo Matching Algorithm Using Adaptive Window Information," *The 2008 International Symposiums on Information Processing (ISIP)*, 23-25 pp. 25–30, May 2008.

[Mcl97] P. F. McLauchlan, et J. Malik, "Vision for Longitudinal Vehicle Control", *in Proceedings of the Eighth British Machine Vision Conference (BMVC'97)*, 1997.

[Mal00] A. Mallet, S. Lacroix et L. Gallo. "Position estimation in outdoor environments using pixel tracking and stereovision." *In IEEE International Conference on Robotics and Automation*, volume 4, pp. 3519-3524, San Francisco (USA), April 2000.

[Mar96] M. Martin et H. Moravec, "Robot Evidence Grids", Technical report CMU-RI-TR-96-06 , *The Robotics Institute, Carnegie Mellon University, Pittsburgh*, PA, March 1996.

[Marr77] D. Marr et T. Poggio, "A theory of human stereo vision." *A. I. MemoArtifical Intelligence Lab, M. I. T.*, 451, November, 1977.

[Marr79] D. Marr, T. Poggio, "A computational theory of human stereo vision." *In Proc. Of the Royal Society of London B*, vol. 204, pp. 301-328, 1979.

[Marr80] D. Marr, et E. Hildreth, " Theory of edge detection." *Proceedings Royal Society London*, 207 :187–217, 1980.

[Mar82] D. Marr, "Vision", W H Freeman & Co., June 1982.

[Mat10] S. Mattoccia, Fast locally consistent dense stereo on multicore, *Sixth IEEE Embedded Computer Vision Workshop (ECVW2010)*, CVPR workshop , San Francisco, USA, June 13, 2010.

[Mez09] Y. Mezouar, "Optimal Camera Trajectory under visibility constraint in visual servoing : a variational approach", *Advanced robotics*, 23(12-13) : 534-549, 2009.

[Mez03] Y. Mezouar et F. Chaumette, "Optimal camera trajectory with image based control" *International Journal of Robotics Research*. 22(10-11) :781-805, October-November 2003.

[Mor83] H. P. Moravec, "The Stanford Cart and the CMU Rover", *Proc. IEEE*, vol. 71, no. 7, pp. 872-884, July 1983.

[Mor85] H. P. Moravec and A. Elfes, "High Resolution Maps from Wide Angle Sonar", *in Proceedings of IEEE International Conference in Robotics and Automation*, pp. 116-121, 1985.

[Mul02] K. Mühlmann, D. Maier, J. Hesser, and R. Männer, "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation," *International Journal of Computer Vision*, vol. 47 pp. 79-88, 2002.

[Mur98] R. Murrieta-Cid. "Contribution au développement d'un système de vision pour robot mobile d'extérieur." *Thèse de doctorat, Institut National Polytechnique de Toulouse, LAAS/CNRS, Toulouse, France*, Novembre 1998.

[Mur01] R. Murrieta-Cid, C. Parra, M. Devy et M. Briot. "Scene modeling from 2D and 3D sensory data acquired from natural environments." *In IEEE The 10th International Conference on Advanced Robotics,* pp. 221-228, Budapest, Hungary, August 22-25 2001.

[Nak95] T. Nakamura et M. Asada, "Motion Sketch : Acquisition of Visual Motion Guided Behaviors", *in Proceedings of 14th International Joint Conference in Artificial Intelligence*, vol. 1, pp. 126-132, August 1995.

[Nak96] T. Nakamura et M. Asada, "Stereo Sketch : Stereo Vision-based Target Reaching Behavior Acquisition with Occlusion Detection and Avoidance", in Proceedings of IEEE International Conference in Robotics and Automation, vol. 2, pp. 1314-1319, April 1996.

[Oht85] Y. Ohta, et T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2), 139–154, 1985.

[Ohy98] A. Ohya, A. Kosaka and A. Kak, "Vision-Based Navigation by Mobile Robots with Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing", *IEEE Transactions Robotics and Automation*, vol. 14, no. 6, pp. 969-978, December 1998.

[Oku01] M. Okutomi, Y. Katayama, et S. Oka. "A simple stereo algorithm to re- cover precise object boundaries and smooth surfaces." *Proceedings of Work- shop on Stereo and Multi-Baseline Vision*, pp. 158–165, 2001.

[Oni09] F. Oniga, S. Nedevschi, R. Danescu, M. Meinecke, "Global map building based on occupancy grids detected from dense stereo in urban environments", *IEEE 5th International Conference on Intelligent Computer Communication and Processing, 2009. ICCP 2009*. 111-117 Cluj-Napoca, 27-29 Aug, 2009.

[Ori95] G. Oriolo, G. Ulivi and M. Vendittelli, "On-Line Map Building and Navigation for Autonomous Mobile Robots", *in Proceedings of IEEE International Conference in Robotics and Automation*, pp. 2900-2906, May 1995.

[Ori97] 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), pp. 179-197, 1997.

[Pan78] D. J. Panton, "A flexible approach to digital stereo mapping," *Photogram. Eng. Remote Sensing*. Vol 44, no. 12, pp. 1499-1512, Dec 1978.

[Pag98] 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), pp. 623-629, 1998.

[Par02] S. Paris et F. Sillion. "Optimisation a base de flot de graphe pour l'acquisition d'informations 3d à partir de séquences d'images." *Actes des 15emes journées de l'AFIG*, 2002.

[Pra85] K. Pardzny, Detection of binocular disparities, *Biological Cybern.*, vol 52 pp. 93-99, 1985.

[Pra78] W. K. Pratt, "Digital Image Processing", *John Wiley & Sons*, New York, 1978.

[Pre92] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, "Numerical Recipes in C : The art of scientific computing". *Cambridge University Press*, 1992.

[Riz98] A. Rizzi, G. Bianco et R. Cassinis, "A Bee-Inspired Visual Homing Using Color Images", *Robotics and Autonomous Systems*, vol. 25, no. 3-4, pp. 159-164, November 1998.

[Rib01] 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), pp. 201-209, 2001.

[Ros82] A. Rosenfeld., A. C. Kak, "Digital Picture Processing", vol. 1. *Academic Press*, Orlando, USA, 1982.

[Roy98] S. Roy et I. J. Cox. "A maximun-flow formulation of the n-camera stereo correspondance probleme." *Proceedings of 6th International Conference on Computer Vision*, pp. 492–499, 1998.

[Roy99] S. Roy. "Stereo without epipolar lines : A maximun-flow formulation." *International Journal of Computer Vision*, 34(2) :147–161, 1999.

[San93] J. Santos-Victor, G. Sandini, F. Curotto et S.Garibaldi, "Divergent stereo for robot Navigation : learning from bees", *In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York 1993.

[Sar02] R. Sara. "Finding the largest unambiguous component of stereo matching." *Proceedings 7th European Conference on Computer Vision*, 3 :900–914, 2002.

[Sim98] S. Simhon, et G. Dudek, "A global topological map formed by local metric maps." *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 17081714, Victoria, BC, Canada, 1998.

[Smi95] S. M. Smith et J.M. Brady, "ASSET-2 : Real-Time Motion Segmentation and Shape Tracking", *in IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, No. 8, pp. 814-820, August 1995.

[Sta00] C. Stauffer and W. E. L. Grimson. "Learning patterns of activity using realtime tracking". *IEEE Trans. On PAMI*, 22(8) :747–757, 2000.

[Sze02] R. Szeliski et D. Scharstein. Symmetric sub-pixel stereo matching. Proceedings 7th European Conference on Computer Vision, 2 :525–540, 2002.

[Sch96] C. Schmid, "Appariement d'images par invariants locaux de niveau de gris." *PhD thesis, INPG*, 1996.

[Sch98] C. Schmid et A. Zisserman. "The geometry and matching of curves in multiple views." *Proc. European Conf. Computer Vision*, pp. 104–118, 1998.

[Sch10] M.R. Schmid, M. Maehlisch, J. Dickmann, H. J. Wuensche, "Dynamic level of detail 3D occupancy grids for automotive use", *2010 IEEE Intelligent Vehicles Symposium (IV)*, 269-274, San Diego, CA, 21-24 June 2010.

[Tao01] H. Tao, H. S. Sawhney, et R. Kumar. "A global matching framework for stereo computation." *Proceedings of the 8th International Conference on Computer Vision,* pp. 532–539, 2001.

[Thr02] S. Thrun, "Robotic mapping : A survey," *In Lakemeyer G. et Nebel, B., éditeurs : Exploring Articial Intelligence in the New Millenium. Morgan Kaufmann,* 2002.

[Thr93] S. Thrun, "Exploration and model building in mobile robot domains". *in Proceedings of IEEE International Conference on Neural Networks* , Seattle, Washington, USA : IEEE neural Network Council, pp. 175-180, 1993.

[Thr98] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation." *Artificial Intelligence*, 99(1) :2171, 1998.

[Thr01] S. Thrun, "Learning occupancy grids with forward models", *in Proceedings of the Conference on Intelligent Robots and Systems (IROS'2001)* , 2001.

[Thr03] S. Thrun, Learning occupancy grids with forward sensor models, Autonomous Robots, vol. 15, 2003.

[Tom96] C. Tomasi et R. Manduchi. "Stereo without search." *In ECCV,* volume 1, pp. 452–465, 1996.

[Tom01] N. Tomatis, "Hybrid, Metric-Topological, Mobile Robot Navigation." *Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne*, Switzerland, 2001.

[Vek99] O. Veksler. "Efficient Graph-Based Energy Minimization Methods in Computer Vision." *PhD thesis, Cornell University*, 1999.

[Vek01] O. Veksler, "Stereo matching by compact windows via minimum ratio cycle," *International Conference of Computer Vision*, pp. 540-547, 2001.

[Vek02] O. Veksler, Stereo Correspondence with Compact Windows via Minimum Ratio Cycle, IEEE Transactions on Pattern Analysis and Machine Intelligence , vol. 24, 2002.

[Vek03] O. Veksler, "Fast Variable Window for Stereo Correspondence using Integral Images," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. I, pp.556-561, 2003.

[Vek05] O. Veksler, "Stereo Correspondence by Dynamic Programming on a Tree," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 384-390, 2005.

[Vez05] V. Vezhnevets, A. Velizhev , "GML C++ camera calibration toolbox", 2005. *http ://research.graphicon.ru/calibration*.

[Vin01] E. Vincent et R. Laganiere. "Matching feature points in stereo pairs : A comparative study of some strategies." *Machine Graphics and Vision*, 10(3) :237–259, 2001.

[Wat98] A. Watt and F. Policarpo, "The Computer Image", *Addison-Wesley*, 1998.

[Wil92] B. Wilcox, L. Matthies, D. Gennery, B. Cooper, T. Nguyen, T. Litwin, A. Mishkin et H. Stone. "Robotic vehicles for planetary exploration." *In IEEE International Conference on Robotics and Automation*, volume 1, pp. 175-180, May 12-14 1992.

[Wys82] G. Wyszecki, W. S. Stiles, "Color Science : Concepts and Methods, Quantitative Data and Formulae". *Wiley*, 2nd Edition, 1982.

[Xie89] M. Xie. Contribution a la visio dynamique : Reconstruction d'objets 3D polyedriques par une camera mobile. *PhD thesis*, Rennes, 1989.

[Yan09] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation, and Occlusion Handling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, Issue 3, pp. 492-504, 2009.

[Yoo05] K. J. Yoon, and I. S. Kweon, "Locally Adaptive Support-Weight Approach for Visual Correspondence Search" *IEEE Conference Proceedings of Computer Vision and Pattern Recognition (CVPR05)*, volume 2, San Diego, USA, pp. 924-931, June 2005.

[Yoo06] K. Yoon and I. S. Kweon, Adaptive Support-Weight Approach for Correspondence Search, *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 28, pp. 650-656, 2006.

[Zab94] R. Zabih et J. Woodfill. "Non-parametric local transforms for computing visual correspondance." *Third European Computer Vision*, pp. 150–158, 1994.gg

[Zha02] Y. Zhang et C. Kambhamettu. "Stereo matching with segmentation-based cooperation." *Proceedings 7th European Conference on Computer Vision*, 2 :556–571, 2002.

[Zha09] Z. Zhai, Y. Lu, and H. Zhao, Stereo Matching with Adaptive Arbitrary Support-Pixel Set and Adaptive Support-Weight, *IEEE International Conference on on Computational Intelligence and Security*, Beijing, China. December 11-14, vol. 1, pp.598-602, 2009.

[Zin98] P. Zingaretti et A. Carbonaro, "Route Following Based on Adaptive Visual Landmark Matching", *Robotics and Autonomous Systems*, vol. 25, no. 3-4, pp. 177-184, November 1998.

[Zit00] C. L. Zitnick et T. Kanade. "A cooperative algorithm for stereo matching and occlusion detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7) :675–684, 2000.