Cartographie hybride métrique topologique et sémantique pour la navigation dans de grands environnements

Romain Drouilly

▶ **To cite this version:**

Romain Drouilly. Cartographie hybride métrique topologique et sémantique pour la navigation dans de grands environnements. Autre. Université Nice Sophia Antipolis, 2015. Français. NNT : 2015NICE4037 . tel-01176848

HAL Id: tel-01176848
https://theses.hal.science/tel-01176848

Submitted on 16 Jul 2015

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

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

UNIVERSITÉ NICE - SOPHIA ANTIPOLIS

ÉCOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

T H È S E

pour l'obtention du grade de

Docteur en Sciences

de l'Université de Nice-Sophia Antipolis

Mention : AUTOMATIQUE TRAITEMENT DU SIGNAL ET DES IMAGES

Présentée et soutenue par

Romain DROUILLY

Cartographie hybride métrique topologique et sémantique pour la navigation dans de grands environnements

Thèse dirigée par Patrick RIVES

préparée à l'INRIA Sophia Antipolis, Équipe LAGADIC

soutenue le 29 Juin 2015

Jury :

Rapporteurs :	David FILLIAT	-	ENSTA ParisTech
	Simon LACROIX	-	LAAS/CNRS
Directeur :	Patrick RIVES	-	INRIA Méditerranée
Président :	*Pas encore défini*		
Examinateurs :	Philippe BONNIFAIT	-	UTC
	Benoit MORISSET	-	PixMap
	Bruno VALLET	-	IGN

Remerciements

En tout premier lieu je tiens à remercier messieurs David Filliat et Simon Lacroix pour leur travail de rapporteur ainsi que messieurs Philippe Bonnifait et Bruno Vallet pour avoir examiné mon manuscrit. Je tiens à remercier tout particulièrement mon directeur de thèse, Patrick Rives et mon encadrant industriel, Benoit Morisset, qui m'ont suivit tout au long de ces trois années et m'ont permis de travailler sur un sujet passionnant et plus que jamais d'actualité. Nos échanges ont été une forte source de motivation et leurs conseils avisés m'ont toujours permis d'avancer efficacement dans mon travail. Je tiens aussi à remercier chaleureusement Panagiotis Papadakis avec qui j'ai eu le plaisir de travailler en étroite collaboration ainsi que Tawsif Gokhool pour son amitié partagée tout au long de ces trois années de thèse.

Je remercie l'équipe Lagadic pour le cadre de travail agréable qu'elle m'a offert tout au long de ces trois années et pour la richesse de nos échanges techniques. Mes sincères remerciements vont à toute l'équipe d'ECA Saclay pour l'accueil chaleureux dont j'ai bénéficié et à toutes les personnes avec qui j'ai pu partager ma passion pour la robotique. Enfin je tiens à remercier ma famille, qui m'a toujours soutenu et encouragé tout au long de ces années d'études.

Table des matières

Introduction

Introduction

Que ce soit pour explorer des mondes éloignés comme la surface de Mars ou des satellites Joviens ou pour intervenir dans des zones dangereuses, la conception de machines autonomes capables d'explorer des environnements inaccessibles à l'homme a été un sujet de recherche très actif depuis de nombreuses années. Aujourd'hui, l'intérêt pour le développement de telles machines est accru par l'émergence des véhicules intelligents et le déploiement d'un nombre croissant de robots dans notre univers quotidien. De nombreux projets ont été lancés pour accélérer la mise au point de véhicules automatiques parmi lesquels on peut citer le DARPA Grand Challenge aux États-Unis ou des projets Européens comme CyberCars et plus particulièrement Français, comme MobiVIP. Plusieurs prototypes ont démontrés leur capacité à évoluer avec une relative autonomie dans des environnements réels. Mais ces résultats ne doivent pas tromper sur l'ampleur des défis qui restent à relever. La Google Car, par exemple, qui a réalisée plusieurs dizaines de milliers de kilomètres dans le désert et en milieu urbain avec un minimum d'intervention humaine, est incontestablement un succès. Cependant la route sur laquelle elle évolue est un environnement contraint où le comportement des agents est le plus souvent codifié et prévisible et la survenu d'événements inattendus rare. Par ailleurs, dans ce cas, le problème de la localisation est résolu par l'utilisation d'un gps ce qui n'est pas toujours possible pour des robots.

La conception de machines capables d'évoluer en totale autonomie dans un environnement quelconque reste donc aujourd'hui un défi technologique majeur, sans doute l'un des plus grands auxquels l'homme n'ait jamais été confronté. Du fait de la diversité des situations rencontrées dans des environnements réels, les approches consistant à définir des schémas de comportement pour chacune d'elles sont inadaptées et il est nécessaire de doter les machines de la capacité à évoluer et à s'adapter à leur environnement. Pour cela les robots doivent pouvoir apprendre à partir de leurs perceptions et construire une représentation interne du monde. L'apprentissage a été utilisé très tôt pour acquérir de façon automatique une carte de l'environnement dont l'objectif est de permettre au robot de se localiser et de naviguer. La construction de cette carte, connue sous l'acronyme SLAM, a fait l'objet de nombreuses recherches depuis des années. Aujourd'hui il existe une multitude de modèles, dont beaucoup peuvent être regroupés en deux familles, représentant l'environnement par sa géométrie ou sa topologie. Mais ces approches souffrent de nombreux problèmes et la modélisation d'environnements réels n'est pas encore maîtrisée. La grande dimension des lieux dans lesquels les robots pourraient être déployés, impose la construction de modèles de grande dimension et la gestion de la masse d'information qu'ils représentent est délicate. Par ailleurs la variabilité des ces environnements modifie régulièrement leur apparence et est encore très mal pris en compte par ces modèles qui ne sont donc pas adaptés à la navigation en environnements réels.

Récemment, l'enrichissement de ces modèles avec de l'information sémantique a permis d'envisager de nouvelles utilisations de ces cartes et une amélioration des interactions hommes/robots en dotant ces derniers d'un langage commun. Mais l'introduction d'information sémantique a surtout amorcé un changement de paradigme profond quant à la nature de ces représentations. Initialement destinées à ne modéliser que l'apparence de l'environnement, notamment son étendue spatiale, elles représentent aussi la nature des objets rencontrés et identifiés. Peu utilisée jusqu'ici pour améliorer les performances des robots, cette information additionnelle peut s'avérer d'une importance majeure pour augmenter leur autonomie, notamment pour la navigation, en permettant la réalisation de comportements complexes, fruits d'une meilleure adaptation des robots à leur environnement.

Objectifs

L'objectif poursuivi dans cette thèse est d'utiliser l'interprétation de scène pour enrichir les représentations du monde qu'utilisent les robots dans le but d'améliorer les performances de navigation dans des environnements réalistes, arbitrairement grands et potentiellement dynamiques. Il faut pour cela concevoir un modèle de carte qui intègre les contraintes de ces milieux et qui représente l'information sémantique sous une forme adaptée à la navigation. Le premier problème lorsque l'on souhaite construire une telle carte, vient de la nécessité de structurer une grande quantité de données qui doivent être rapidement accessibles par le robot. La sélection de l'information utile à la navigation et sa représentation sous une forme compacte et discriminante est alors fondamentale. Le second problème vient de la nature intrinsèquement dynamique des environnements réels dans lesquels sont amenés à évoluer les robots. Une représentation statique devenant rapidement obsolète il faut disposer d'un moyen de mettre à jour la carte. Ceci est particulièrement difficile dans de vastes environnements où l'accumulation de multiples observations pour l'ensemble du territoire couvert par la carte n'est possible qu'au prix d'un accroissement considérable du volume de données et d'une mise à jour permanente de celles-ci, ce qui est rarement compatible avec la réalité des moyens opérationnels engagés. Dans un second temps, il faut être en mesure d'utiliser cette représentation pour naviguer. Pour cela des algorithmes de localisation et de planification de chemin s'appuyant sur cette représentation doivent être développés.

Contributions

Les travaux réalisés pendant cette thèse ont fait l'objet de plusieurs publications portant sur la cartographie, la localisation et la planification de chemin en utilisant l'information sémantique :

1. Dans [Drouilly et al 2014a] est introduite une nouvelle représentation de l'environnement basée sur l'information sémantique locale. Un algorithme de localisation

conçu pour exploiter les spécificités de cette représentation et dont les performances dépassent celles de méthodes classiques basées sac-de-mots, est présenté.

2. Dans [Drouilly et al 2014b] le procédé d'extrapolation de carte est introduit. En utilisant des indices sémantiques et la présence d'objets dynamiques, il permet au robot d'étendre la cartographie au delà de ses limites perceptuelles. Ce travail participe à montrer que l'intérêt d'ajouter de l'information sémantique dans une représentation ouvre de nouvelles perspectives pour la modélisation et la compréhension d'environnements.
3. Dans [Drouilly et al 2015] sont présentés deux algorithmes de planification et de description de chemin exploitant la représentation sémantique proposée précédemment. L'objectif est de développer une alternative à l'approche traditionnelle consistant à sélectionner le meilleur chemin sur la base de critères purement métriques, en interprétant la scène pour prendre en compte des contraintes de haut niveau.

Organisation du manuscrit

Ce manuscrit est divisé en quatre parties principales :

1. Dans la première partie, les principaux modèles d'environnements proposés dans la littérature sont détaillés. Cet état de l'art permet de souligner l'évolution de la manière dont a été traité le problème de la cartographie en distinguant notamment les premiers modèles, purement spatiaux, des modèles récents intégrant de l'information de nature non spatiale comme les cartes sémantiques. Il permet de justifier la stratégie adoptée dans ce manuscrit.
2. La deuxième partie détaille le nouveau modèle d'environnement proposé dans cette thèse en s'appuyant sur un jeu de fonctions de cartographie intervenant à différentes étapes du processus de construction du modèle. Après un rappel des résultats de [Meilland 2012] au chapitre 3, un procédé pour extraire de manière robuste une information sémantique fiable d'une image sphérique est présenté au chapitre 4. Un modèle local et compact de l'environnement basé sémantique est proposé au chapitre 5 ainsi qu'un algorithme de supervision de la qualité de la labelisation. Un procédé permettant la mise à jour de la représentation, adapté aux grands environnements, est développé au chapitre 6. Enfin, une méthode permettant d'exploiter la richesse de l'information sémantique pour étendre la surface cartographiée par le robot et augmenter la robustesse de la cartographie en exploitant le dynamisme de la scène est présenté au chapitre 7.
3. La troisième partie propose des algorithmes de navigation s'appuyant sur le modèle introduit dans la seconde partie. Le chapitre 8 présente un algorithme de localisation permettant de trouver une position avec des performances égales, si ce n'est supérieures, aux algorithmes les plus performants tout en autorisant des requêtes de

haut niveau compréhensibles par l'homme. Dans le chapitre 9 sont introduits des algorithmes de planification et de description de chemin s'appuyant sur la représentation compacte développée précédemment pour sélectionner la meilleure route suivant des critères de haut niveau et la décrire dans un langage compréhensible par l'homme.

4. La quatrième partie regroupe les conclusions et met en perspective les contributions de ces travaux de thèse par rapport à l'état de l'art présenté en première partie. Outre un bilan des avancées réalisées, la trajectoire globale des recherches de ces dernières années est analysée et plusieurs pistes de réflexion pour la poursuite de ces travaux à court et moyen terme sont proposées.

Première partie

Modélisation d'environnements pour la navigation : des cartes spatiales aux modèles sémantiques

CHAPITRE 1

Modèles spatiaux

1.1 Introduction

La navigation est la tâche la plus fondamentale que doit réaliser un robot mobile autonome. Elle peut recouvrir des réalités très diverses selon le contexte dans lequel évolue le robot. Ici, naviguer s'entend au sens le plus large, c'est à dire réaliser un ensemble de tâches qui permettent, à partir d'une positon donnée, d'atteindre une position quelconque, dans un environnement arbitrairement grand et dans lequel évoluent d'autres agents intelligents, biologiques ou artificiels. Pour naviguer efficacement, le robot doit être capable, d'une part, de se localiser dans son environnement et d'autre part, de planifier sa trajectoire vers une position cible en tenant compte des contraintes de l'environnement. A défaut de disposer d'un moyen de localisation externe, la navigation demande donc que le robot possède une représentation interne de son environnement. La construction automatique de ce modèle est ardue pour au moins trois raisons :

- La dimensionnalité du problème : les environnements explorés peuvent être arbitrairement grands et les données acquises représenter un volume d'information conséquent. Se pose donc la question de savoir comment représenter l'environnement de manière compacte.
- L'environnement est intrinsèquement dynamique : le modèle ne peut donc pas être statique. Comment capturer la dynamique de la scène dans la représentation ?
- Le robot n'observe pas les objets directement, ses capteurs ne fournissent que la mesure de certaines grandeurs physiques. Faut-il représenter les données ou une interprétation des données ?

Depuis le début de la robotique mobile, le problème de l'acquisition automatique d'un modèle de l'environnement par un robot a occupé une part significative des recherches. Selon le contexte opérationnel, la nature de l'environnement et des capteurs disponibles, de nombreuses représentations différentes ont été proposées, chacune conduisant à des stratégies de navigation, de localisation et de planification de trajectoire différentes. Il existe un très grand nombre de facteurs permettant de classer les différents modèles de carte proposés, entre ceux développés pour l'intérieur ou pour l'extérieur, ceux acquis avec des lasers ou des caméras ou encore en fonction du type de plateforme utilisée (UAV,UGV...). Dans ce chapitre les principaux modèles sont présentés de façon à souligner le fil conducteur ayant guidé les recherches en cartographie jusqu'à aujourd'hui, dans le but de mettre en perspective la démarche globale et d'identifier les objectifs actuels.

1.2 Cartes métriques

La première classe de modèles a avoir émergé regroupe les cartes métriques. Elles représentent l'environnement par sa géométrie avec un point de vue géocentré, c'est à dire que l'ensemble des données est repositionné par rapport à un même référentiel global, souvent le point de départ du robot. Il s'agit sans doute de la représentation la plus intuitive de l'espace. Une très grande variété de représentations métriques ont été proposées que l'on peut regrouper en deux familles : les méthodes qui représentent l'espace lui même, le plus souvent par une discrétisation de celui-ci, les grilles d'occupation ; les méthodes représentant les objets dans l'espace, le plus souvent sous forme de nuages de points ou de primitives géométriques.

1.2.1 Grilles d'occupation

Grille 2D L'un des premier modèle d'environnement est la *grille d'occupation*, proposée dans [Elfes 1987, Elfes 1989]. Comme pour beaucoup d'algorithmes de cartographie, les grilles d'occupations reposent sur le filtrage bayésien. L'espace est représenté par une grille 2D dont chaque cellule possède une probabilité d'occupation dépendant des observations et qui peut être mise à jour au fur et à mesure de l'exploration. Du fait de sa relative simplicité et de sa robustesse, ce modèle été très largement utilisé, notamment avec des lasers et des sonars [Burgard 1999]. Cependant, l'impossibilité de modéliser la hauteur des obstacles le rend aussi difficilement exploitable par d'autres robots que celui ayant acquis le modèle : en plaçant le capteur à deux hauteurs différentes, la carte obtenue n'est pas nécessairement la même.

Cartes d'élévation Pour corriger ce problème, une première extension des grilles d'occupation 2D a été envisagée dans le cas où certaines hypothèses peuvent être faites sur l'environnement [Herbert 1989]. La carte résultante est une représentation 2.5D de l'environnement, c'est à dire une grille dans laquelle chaque cellule, au lieux de coder la probabilité d'occupation, mesure la hauteur du terrain. Ce type de carte a été utilisée avec succès dans le cas de robot navigant en extérieur [Hadsell 2009]. Le modèle de la carte d'élévation nécessite, dans sa version de base, de faire l'hypothèse assez restrictive que l'environnement peut être modélisé par une seule surface continue. Cependant d'autres possibilités ont été explorées en relâchant l'hypothèse d'une surface unique comme dans [Pfaff 2007] où plusieurs surfaces sont admises ou dans [Dryanovski 2010] qui stocke pour chaque cellule une liste des voxels occupés et libres dans une *grille d'occupation multivolume*. D'autres représentations ont été proposées en s'appuyant sur l'hypothèse *manhattan-world*, c'est à dire un monde où les structures sont purement verticales. Ce modèle a été utilisé pour reconstruire une scène en 3D dans [Furukawa 2009].

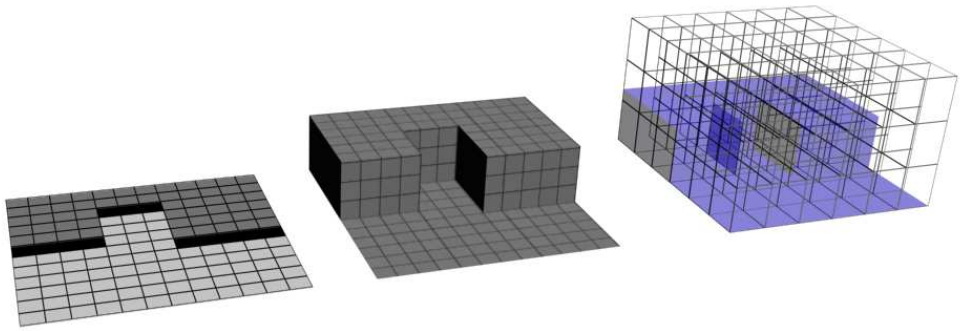


FIGURE 1.1 – *Les différents type de représentations basés sur des grilles. A gauche : grille d'occupation 2D modélisant l'espace libre, l'espace occupé et l'espace non observé. Au milieu : grille d'élévation donnant pour chaque cellule la hauteur de l'espace associé. A droite : voxellisation de l'espace sous forme d'une grille 3D dont seules certaines cellules, matérialisée en bleu, sont occupée.*

Extension 3D Le succès des grilles d'occupation 2D a conduit à étendre le formalisme à la 3D [Moravec 1994]. Les cases 2D sont remplacées par des *voxels* de taille homogène dans tout l'espace. Tout comme dans le cas 2D, la taille du maillage doit être défini au préalable de façon à ce que la grille soit au moins aussi grande que l'espace à explorer. Dans le cas d'environnement de grande dimension et lorsqu'une précision locale importante est requise, la taille du modèle devient vite très importante tout comme dans le cas de points 3D. Une approche plus sophistiquée à base d'octree a été utilisée avec OctoMap [Hornung 2013] pour résoudre ce problème, l'idée ayant été initialement proposée dans [Meagher 1982]. Au lieu d'utiliser un maillage homogène de l'espace, la grille d'occupation est remplacée par un octree. Ceci présente plusieurs avantages. Premièrement le modèle global est beaucoup plus compacte qu'avec une grille régulière puisque les vaste zones vides de la carte peuvent être encodée dans un petit nombre de cases tandis que les zones occupées bénéficient d'un découpage plus fin. Deuxièmement l'accès aux données est beaucoup plus rapide qu'avec une grille régulière du fait de la structure hiérarchique des octrees et de l'existence d'algorithmes de recherche puissants. Enfin, avec un octree la taille de l'environnement n'a pas besoin d'être connue initialement et peut être ajustée au fur et à mesure de l'accumulation des mesures en ajoutant une branche à la structure.

En dépit d'évolutions fructueuses des modèles basés sur une discrétisation de l'espace, de nombreux inconvénients demeurent qui rendent peu envisageable leur utilisation à grande échelle dans des cas réels. En premier lieu, les grilles d'occupation n'offrent aucun

moyen de prendre en compte l'incertitude sur la pose ou les mesures, supposant un algorithme de localisation parfait et des capteurs d'une extrême précision, ce qui n'est généralement pas le cas dans la pratique. Dans le cas de grand environnements, la localisation est sujette à une dérive qui rend ces modèles difficilement utilisables. Ensuite, ces représentations utilisent comme unité de base des éléments d'espace discrétisés qui les rendent inutilisables à grandes échelles. Le compromis nécessaire quant à la taille des cellules et de l'espace modélisé conduit soit à des discontinuités du modèle local qui rendent l'analyse de la géométrie délicate, soit à un modèle trop lourd. Enfin et dans une moindre mesure, une hypothèse de départ de toutes ces représentations est l'indépendance statistique des cases de la grille, qui n'est, la plupart du temps, pas vraie.

1.2.2 Représentations basées mesures

La deuxième famille de modèle ne représente plus l'espace lui même mais les objets [1] qu'il contient, soit en représentant directement les mesures dans l'espace euclidien, soit en représentant leur interprétation.

Cartes de points La première manière de représenter le contenu de l'espace consiste simplement à représenter les mesures des capteurs sous forme d'un ensemble de points épars, notamment les amers utilisés pour la navigation. Dans *MonoSlam* [Davidson 2007] ce type de carte est construit à partir d'un système de vision monoculaire. Des points d'intérêt sont extraits des images et leurs positions sont estimées et mémorisées pour former un embryon de modèle d'environnement. Une forme un peu plus évoluée de représentation consiste à construire une carte dense de points, en 2D ou en 3D [Lu 1997, Thrun 2000, Cole 2006]. Ce type de modèle offre une représentation plus réaliste visuellement que les cartes éparses puisque la densité des points permet de simuler la continuité des surfaces. Cependant, l'accumulation des mesures conduit à une redondance de l'information dans les zones où elles se superposent et en fait des modèles gourmands en mémoire. De plus, ce type de modèle n'enregistre que la position des points mesurés à un instant donné, sans estimer ni l'espace libre ni les zones non explorées. Il est alors impossible de mettre à jour cette carte en retirant des mesures qui pourraient être erronées. En outre, sans information sur l'orientation des normales, rien dans le modèle ne permet de faire la différence entre la face observée d'un objet et la partie cachée. Il n'y a donc aucun moyen de savoir si une surface est observable depuis un point de vue quelconque. Ces limitations rendent ces représentations adaptées uniquement pour la modélisation des scènes statiques et observées avec des capteurs de grande précision.

Carte de primitives géométriques L'idée de remplacer les mesures par des primitives géométriques a été introduite très tôt [Chatila 1985] où un ensemble de lignes continues

1. Le terme d'objet est ici à prendre au sens large, c'est à dire l'ensemble de ce que contient l'espace.

FIGURE 1.2 – *De gauche à droite : représentation de l'environnement sous forme de nuage de points, de plans puis de plans texturés. Sur la carte du milieu composée de plans non texturés, le problème de la dérive de l'odométrie dans les cartes métriques est bien visible : le sol est matérialisé par 3 plans distincts au lieu d'un seul du fait des erreurs de reconstruction de la carte.*

est utilisé, se rapprochant davantage de la réalité que les nuages de points. En plus d'offrir un modèle plus compact, la représentation est aussi plus précise car les primitives sont estimées à partir de nombreuses mesures dont l'erreur individuelle est ainsi atténuée. Ce type de carte permet une interprétation à plus haut niveau de la scène utile pour la navigation. Dans le même esprit mais en 3D cette fois [Martin 2002] propose d'extraire les plans de l'environnement et d'approximer les zones non planes par des polygones. L'algorithme EM est utilisé pour estimer les zones planes et reconstruire en ligne une carte à l'aspect réaliste. Dans [Biber 2003] est introduit une nouvelle représentation, la *Normal Distribution Transform* ou NDT. Au lieu de d'utiliser les points d'un nuage, cette représentation est basée sur un ensemble de distributions normales qui encodent la probabilité de trouver une surface à une certaine position. En plus d'offrir une représentation lisse, cette modélisation est aussi beaucoup plus compacte qu'un nuage de point équivalent. Elle a été étendu au cas 3D dans [Magnusson 2007].
Dans le même esprit, une modélisation à base de surfaces planes a été proposée dans [Habbeck 2007] où la forme des objets est approximée par croissance de région en utilisant un ensemble de disques plans nommés *Surfels*. Ces *Surfels* sont des entités ayant comme caractéristique une position, une orientation,une taille et potentiellement une couleur. Cette représentation associée à un algorithme de SLAM a été utilisée dans [Peter 2010] pour la reconstruction en temps réel d'environnement intérieur. Bien que plus précises que les représentations utilisant une discrétisation de l'espace car modélisant des *surfaces*, ces méthodes présentent tout de même l'inconvénient de ne pas faire la différence entre espace libre et inconnu. Une représentation alternative permettant de modéliser à la fois l'espace occupé, vide ou inconnu et permettant de caractériser la surface a été introduite dans [Curless 1996]. Les objets sont modélisés par une fonction, la *signed distance function*, ou SDF, qui s'annule au niveau de leur surface, est positive lorsque l'espace est libre et négative lorsqu'il est inconnu. Une représentation similaire a été utilisée KinectFusion [Newcombe 2011].

L'avantage majeur des méthodes basées primitives géométriques par rapport aux cartes

de points ou aux grilles d'occupation est qu'elles permettent de capturer la géométrie de la scène en tant qu'espace continue. En ce sens elles constituent une véritable amélioration par rapport aux modèles d'environnement discrétisés. Cependant, aussi précis que soient ces modèles et bien que des expériences de navigation aient été menées avec succès pour ce type de carte [Stoyanov 2010], au moins deux problèmes majeurs s'opposent encore à l'utilisation de carte métriques à grande échelle pour la navigation autonome. Premièrement, la modélisation de grands environnements est contrainte par le problème de l'accumulation des données. En effet représenter l'ensemble des informations dans un référentiel unique suppose une carte de taille modeste pour être facilement utilisable par la suite. Deuxièmement, la dynamique des objets n'est pas modélisée dans ces représentations. Il est donc nécessaire de les mettre continuellement à jour pour éviter leur obsolescence ce qui n'est que très difficilement envisageable dans un contexte opérationnel réaliste. Les cartes métriques sont donc plutôt destinées à des espaces de dimension modeste, comme les environnements intérieurs et peu dynamiques.

1.3 Carte topologique

Un second paradigme pour la modélisation spatiale de l'environnement a été développé en parallèle de la cartographie métrique. Il s'agit de la cartographie topologique. Dans ce type de modélisation la géométrie exacte de l'environnement n'est plus représentée directement mais l'espace est découpé en lieux qui forment les nœuds d'un graphe. Chaque arête reliant deux nœuds peut être associée à une relation de connectivité entre ces lieux, à une action du robot ou une particularité topologique de l'espace, comme une porte marquant la limite entre deux zones. A la différence des cartes métriques, les données ne sont pas mises en relation dans un référentiel globale. De ce fait le problème de l'estimation métrique de la position ne se pose plus. Par ailleurs de vastes zones peuvent n'être représentées que par un unique nœud dans le graphe, permettant d'obtenir un modèle très compact. Ces cartes se prêtent donc particulièrement bien à la modélisation d'espaces de grande dimension. Elles offrent aussi de sérieux avantages par rapport au cartes métriques pour la planification de chemin, la complexité algorithmique de la recherche dans un graphe étant moindre que dans un espace continu.

Il existe là encore de nombreuses formes de cartes topologiques qui peuvent être regroupées en fonction de multiples critères, notamment la manière dont sont définis les nœuds et les arêtes. On peut distinguer les techniques qui définissent des lieux de manière supervisée de celles où les lieux sont définis automatiquement par le robot. Les différents modèles peuvent aussi être classés en fonction de l'information codée par les arêtes qui peuvent tour à tour matérialiser des relations métriques, d'adjacence ou les actions du robot pour passer de l'une à l'autre.

Définition supervisée des nœuds Un exemple de définition supervisée de lieux peut être trouvé dans [Kunz 1997]. Ici les auteurs définissent les nœuds du graphe comme les zones de transition de l'environnement, c'est à dire les intersections entre le couloir et une pièce donnée. De même dans [Dedeoglu 1999] les couloirs, les portes et les intersections sont utilisés pour définir les nœuds de la carte. D'autres approches proposant une définition de plus haut niveau du concept de lieu ont été proposées [Ullah 2008, Ulrich 2000]. Elles ne reposent plus sur la reconnaissance d'un lieu basée sur une définition formelle mais sur la classification du lieu défini par l'apprentissage automatique à partir d'exemples. Le processus de construction de la carte topologique consiste alors à connecter les différents ensemble identifiés. Il existe aussi des modèles pour lesquels les nœuds modélisent les états du robot, c'est à dire qu'ils regroupent un ensemble de perception internes et externes, comme le niveau de la batterie, la position dans l'espace métrique donnée par l'odométrie ou des paramètres de l'environnement [Shatkay 2002]. Le plus souvent ces approches sont bien adaptées à des scènes intérieures pour lesquels la notion de lieux est clairement définie. Cependant elles s'adaptent assez mal au domaine extérieur pour lequel il n'existe le plus souvent pas de séparation claire entre des espaces successifs.

Définition non supervisée des nœuds La seconde famille d'approches pour la définition des nœuds consiste à laisser le robot décider lui-même ce qui constitue un lieu sans définition préalable [Wichert 1998, Yamauchi 1996]. Deux approches ont été proposées. Dans le premier cas, le robot regroupe ses perceptions en fonction de leur similarité et utilise des critères propres aux données pour estimer s'il est arrivé dans un nouveau lieu [Chapoulie 2012, Gaussier 2000, Bailey 1999]. Il doit donc disposer d'un moyen efficace de comparer les données. Ce n'est pas une tâche triviale, notamment lorsque la taille de l'environnement croit et avec elle la quantité de données. La seconde approche consiste simplement à considérer que le robot a changé de lieux lorsqu'il a parcouru une distance donnée depuis le dernier lieu enregistré [Yamauchi 1996, Wichert 1998]. Cette approche est d'un intérêt moindre puisqu'elle utilise uniquement l'odométrie pour la définition d'un lieu à la place de critères intrinsèques à l'environnement et est à utiliser lorsque les ressources du système ne permettent pas d'autres techniques.

Définition des arêtes Les différents modèles de carte topologique se différencient aussi par la définition des arêtes reliant deux nœuds. On peut citer le cas des modèles utilisant les arêtes pour coder les relations métriques entre les nœuds [Engelson 1992]. L'avantage de ces méthodes est qu'elles souffrent moins de l'imprécision de l'odométrie que les cartes métriques puisque seules les poses relatives, donc estimées sur de courtes distances, sont enregistrées. D'autres méthodes utilisent les arêtes pour modéliser les actions effectuées par le robot pour passer d'un nœud à l'autre [Bailey 1999, Landsiedel 2013]. Ce type de carte fournit un description de haut niveau directement utilisable pour la commande du robot. Un des avantages de cette représentation est qu'elle offre une description directement accessible à l'homme.

Les cartes topologiques présentent des avantages significatifs pour la modélisation de grands environnements par rapport aux cartes métriques. Mais elles ont aussi des inconvénients non négligeables. Naviguer avec des cartes topologiques requiert la capacité à reconnaître des lieux déjà visités et à en mémoriser de nouveaux. De ce fait les représentations topologiques sont sujettes au problème de *perceptual aliasing*, c'est à dire au risque de confondre plusieurs lieux différents. Ce problème devient particulièrement important lorsque l'environnement contient des lieux d'apparence similaire. De plus, la représentation ego-centrée conjuguée à l'absence de modèle métrique des capteurs rend le modèle valide uniquement au voisinage des zones visitées lors de l'acquisition. De ce fait la réalisation de la carte topologique d'un environnement donné requiert une exploration plus complète de l'environnement que dans le cas de carte métrique. Selon la nature des "lieux" définis par le robot, ce peut être plus ou moins handicapant. Du fait de la simplicité du modèle, qui ne représente pas l'espace libre, il faut utiliser en parallèle un système de détection d'obstacle pour la navigation. Enfin, tout comme les cartes métriques, ces modèles ne représentent pas la dynamique de la scène.

1.4 Les cartes hybrides et hiérarchiques

Le deux paradigmes présentés, carte métrique ou topologique, ont chacun des avantages et des inconvénients pour la navigation. Les cartes métriques proposent une description précise et complète de la géométrie de l'environnement mais au prix d'une consommation importante de ressources. Elles nécessitent par ailleurs une excellente odométrie pour une reconstruction correcte du monde et un moyen de corriger la dérive inévitable de l'estimation de la position. Les cartes topologiques quant à elles offrent une approche différente, modélisant la structure de l'environnement sans représenter l'aspect local. Cette vision permet des représentations plus compactes, mieux adaptées aux vastes environnements mais nécessite l'utilisation en parallèle d'un moyen de détecter les obstacles pour la navigation. Rapidement, les idées issues de chacune des approches ont été utilisées simultanément pour proposer des modèles alternatifs, bénéficiant des avantages des deux méthodes. Ces cartes sont dites hybrides ou hiérarchiques car elles mêlent des informations de nature différentes, organisées sur plusieurs niveaux.

1.4.1 Cartes par superposition

En premier lieu on peut citer les cartes par superposition, c'est à dire où les données métriques et topologiques recouvrant la même surface sont accumulées en parallèle. Il s'agit de la simple superposition de deux cartes métrique et topologique qui sont utilisées suivant les besoins. Ce type de représentation est le plus exigeant en terme de ressources puisque deux représentations s'ajoutent et sont plus sensibles aux erreurs puisqu'elles cumulent les inconvénient des deux représentations. Deux exemples de telles approches sont

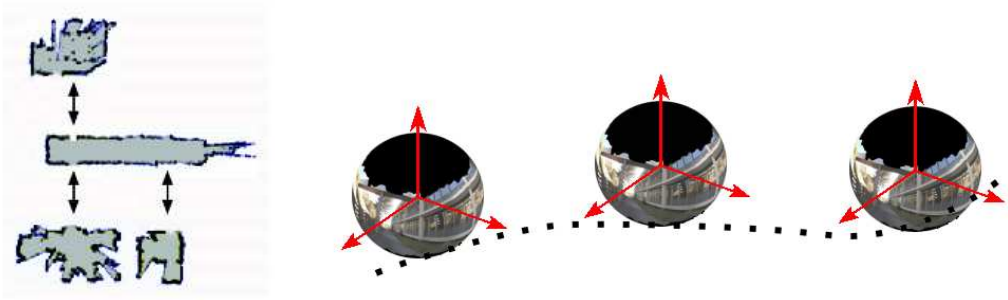
donnés dans [Brunskill 2007, Thrun 1999]. La carte topologique y est construite directement à partir de la représentation métrique en utilisant la segmentation. Les contraintes métriques sont donc implicitement prises en compte dans la carte topologique puisque cette dernière en est extraite. Bien que modélisant simultanément la géométrie et la topologie d'un environnement, ces modèles sont d'un intérêt moindre puisqu'ils nécessitent une carte métrique préalablement acquise.

1.4.2 Carte Pyramidale

Un second type de carte est constitué des structures pyramidales, c'est à dire d'un empilement de couches de même nature modélisant l'environnement à différentes échelles. La topologie est modélisée implicitement par l'approximation grossière de l'occupation spatiale. Par exemple [Nilson 1969] propose une structure hiérarchique dans laquelle le premier niveau est occupé par une grille d'occupation de dimension 4 par 4. Lorsqu'une cellule est occupée elle est découpée en 16 pour affiner la représentation. Ce modèle est à rapprocher des méthodes de cartographie spatiale utilisant des kd-tree [Hornung 2013] dans la mesure ou la discrétisation de l'espace plus ou moins fine est directement imposée par le contenu. Cependant, ici il s'agit bien d'une hiérarchie de cartes à différentes échelles. Ce type de représentation a l'intérêt d'offrir différentes échelles d'analyse de l'environnement, mais nécessite d'avoir une odométrie idéale. En ce sens, elles ne tirent pas partie des avantages des cartes topologiques.

1.4.3 Assemblage de cartes locales

D'autres approches présentent une meilleure synergie entre les différents niveaux de la carte. Ces cartes hybrides sont constituées d'un ensemble de cartes métriques locales connectées via un graphe topologique global. Elles ont été proposées à plusieurs reprises dans la littérature [Tomatis 2002, Meilland 2011, Lisien 2005]. L'idée est de combiner la précision de la représentation métrique, utile pour la navigation locale, tout en s'affranchissant des contraintes dues à la dérive de l'odométrie, en utilisant les principes des cartes topologiques pour connecter les cartes locales. Ce type de représentation résout partiellement le problème des ressources mémoires importantes requises par les cartes métriques en permettant de n'utiliser que des cartes locales de plus petite taille pour la navigation. Dans ce type de représentation chaque couche contribue à l'ensemble de façon à améliorer les performances globales dans la construction et l'utilisation de la carte. Un modèle hybride nommée Atlas hiérarchique, a été proposé dans [Lisien 2005]. La représentation topologique est assurée par un graphe de Voronoï (Reduced Generalized Voronoï Graph) dont les nœuds sont des ensembles de cartes métriques locales. Un modèle basé sur la mémorisation d'images sphériques augmentées de la profondeur et constituant un ensemble de cartes locales, globalement connectées dans un graphe, a été proposé dans [Meilland 2011]. Il permet de bénéficier à la fois des avantages des cartes métriques en reconstruisant



(1) (2)

FIGURE 1.3 – *Différents modèles de cartes hybrides. À gauche, carte hybride par superposition présenté dans [Brunskill 2007]. La carte topologique est déduite de la carte métrique. À droite : carte hybride introduite dans [Meilland 2011] composée de cartes métriques locales sous forme d'images sphériques ego-centrées, augmentées de la profondeur et connectées dans un graphe topologique global.*

l'aspect géométrique de l'environnement local, tout en s'affranchissant des contraintes de la reconstruction de grands environnement en offrant naturellement un moyen de découper la carte en zone d'intérêt. Ce modèle a été utilisé pour la cartographie, la localisation et la navigation temps réel et sera étendu dans le cadre de cette thèse.

1.5 Limites des représentations spatiales

La construction de modèles spatiaux de l'environnement pour la navigation a été le sujet de recherches intenses depuis le début de la robotique mobile. Une multitude de représentations ont été proposées, dont les cartes hybrides sont la forme la plus aboutie. Certaines ont été utilisées par des systèmes ayant démontré leur capacité à naviguer en autonomie dans des environnements spécifiques. Cependant, bien que les cartes hybrides exploitent les avantages des représentations métriques et topologiques sans en avoir tous les inconvénients, ces représentations souffrent encore de défauts qui s'opposent au déploiement à grande échelle de robots autonomes dans des environnements réels. Un problème récurrent est celui de la taille des modèles. Les cartes hybrides permettent de s'affranchir des contraintes liées à la représentation des données dans un référentiel commun en utilisant des cartes locales. Cependant la localisation, du robot ou d'un contenu spécifique, demeure un problème dans de grands environnements, notamment parce qu'il faut disposer d'un moyen efficace de comparer les cartes locales dont le nombre croit rapidement avec la quantité de lieux visités. Par ailleurs, de manière plus importante encore, certains aspects de l'environnement dont la connaissance est utile à la navigation, ne sont pas ou mal modélisés dans ces cartes puisqu'elles ne représentent, par définition, que l'information spatiale. Or d'autres informations sont à prendre en compte pour naviguer, comme la nature des objets rencontrés ou le dynamisme de la scène.

Les limitations des modèles métriques ont donc pour source principale le fait que la navigation requiert des informations de nature autre que spatiale, que ce soit pour la définition des objectifs, la localisation ou tout simplement la modélisation de l'environnement, qui ne peuvent être représenté correctement via ces modèles. Pour cette raison, il est nécessaire de les faire évoluer en les enrichissants de concepts de plus haut niveau. C'est précisément l'objectif de la cartographie sémantique, présentée au prochain chapitre.

CHAPITRE 2

Modèles sémantiques

Aujourd'hui, un changement de paradigme profond est en train de s'opérer dans le domaine de la cartographie. L'objectif n'est plus uniquement de construire une représentation simulant l'apparence et l'étendue spatiale de l'environnement, ce qui s'est avéré être insuffisant pour permettre aux robots d'évoluer en totale autonomie. Le but est de modéliser le contenu sémantique de l'environnement en représentant les objets qu'il contient, pour rendre compte de phénomènes qui ne pouvaient être modélisés par les cartes précédentes. La *cartographie sémantique*, qui intègre des informations quant à la nature des objets observés, offre de nombreux avantages sur les méthodes purement spatiales. C'est un moyen très naturel de partager des représentations avec l'être humain, permettant une interaction homme-robot simple et efficace, essentielle pour envisager le déploiement des robots dans notre environnement quotidien. La connaissance de la nature des objets permet aussi de modéliser les interactions possibles avec ceux-ci et de prendre en compte la dynamique de l'environnement en associant à chaque objet un comportement. Par exemple on peut définir les classes d'objets formant de bons amers pour la navigation, identifier des panneaux de signalisation ou encore caractériser les zones de danger où la navigation sera plus ou moins contrainte. Le recours à l'information de haut niveau permet aussi d'améliorer l'efficacité des algorithmes en modélisant l'environnement de manière compact. La *cartographie sémantique* est donc une voie très prometteuse pour l'amélioration des représentations de l'environnement, notamment pour la navigation.

2.1 Approche moderne de la cartographie

2.1.1 Modèles de cartes sémantiques

L'exploitation d'information sémantique pour la modélisation d'environnement a été envisagée assez tôt dans le domaine de la cartographie [Kuipers 2000]. Cependant, le sens donné au terme sémantique y est assez éloigné de l'acception moderne. Dans ces travaux, la modélisation spatiale de l'environnement est envisagée au travers des expériences sensori-motrices du robot qui, conjointement, permettent de donner du sens aux observations. Mais ce n'est que récemment que le développement de modèles d'environnement intégrant la sémantique en tant qu'information sur la nature des objets observés, n'a connu de véritable essor. Plusieurs structures différentes ont été proposées. La majorité se présente comme des

cartes hybrides dans lesquelles de l'information sémantique a été inclue [Galindo 2005, Vasudevan 2008, Pronobis et Jensfelt 2012], le plus souvent en ajoutant une couche à une carte hybride métrique/topologique . L'un des premiers modèles proposés [Galindo 2005] consiste en une représentation regroupant deux structures hiérarchiques parallèles, illustrée à la figure 2.1. La première est une représentation spatiale dont le niveau le plus bas regroupe des perceptions (images ou grille d'occupation locale), le second niveau la carte topologique de l'environnement et le troisième est un unique nœud représentant l'environnement tout entier. La seconde structure, la hiérarchie conceptuelle, regroupe les concepts comme les lieux, les objets et leur relations. Les deux structures sont liées par *l'ancrage* des concepts dans la scène. Une représentation similaire a été proposée pour la classification des lieux [Vasudevan 2008]. Les objets sont regroupés suivant deux dimensions, sémantique et spatiale. Le regroupement des objets permet de capturer la nature du lieu, définie par les objets qui s'y trouvent, à des échelles de plus en plus grandes. Un autre modèle a été présenté mais avec une approche orientée vision [Ranganathan 2007] se basant sur une extension 3D du modèle de constellation. Là encore les objets sont les entités de base du modèle permettant d'inférer la classe des lieux visités. L'un des modèles les plus complets est présenté dans [Pronobis et Jensfelt 2012]. Il s'agit d'une représentation spatiale à quatre couches. La première est la couche capteur, correspondant à une carte métrique. La seconde est une carte topologique dont les nœuds sont constitués d'éléments de l'espace discrétisé. La couche catégorie contient des modèles statiques, visuels ou géométriques, des objets et des différents types de lieux. Enfin vient la couche conceptuelle qui représente à la fois des concepts spatiaux et leurs instances dans la carte, ainsi que les relations qui les lient. Enfin, d'autres approches ne reposant pas sur une carte hybride mais sur une carte métrique ont aussi été proposées [Nuchter et Hertzerg 2008]. Elles consistent à annoter les points d'un nuage 3D pour obtenir une carte métrique augmentée par la sémantique.

2.1.2 Synergie des cartes sémantiques

L'exploitation conjointe des informations des différentes couches d'une carte hybride sémantique a fait l'objet de recherches significatives pour la construction de modèles évolués de l'environnement. Dans la majorité des cas, l'information métrique est utilisée pour inférer la sémantique, mais certains travaux utilisent ensuite l'information sémantique pour corriger l'information métrique à grande échelle. Un exemple d'une telle approche est donné dans [Nuchter et Hertzerg 2008]. Les plans extraits d'un nuage de points 3D sont annotés puis utilisés pour contraindre la construction de la carte métrique en faisant l'hypothèse raisonnable que le sol est plan et les murs droits et parallèles. Ici l'interprétation à un haut niveau de la nature de l'environnement, comme la détection d'un mur, permet de corriger les défauts de la couche métrique. De même, l'information sémantique concernant la présence d'objets particuliers comme des portes, peut être utilisée pour inférer la nature du lieu visité et construire la couche topologique [Vasudevan 2008]. Mais la topologie des lieux peut aussi être utilisée pour inférer la nature d'une pièce. Un exemple intéressant est

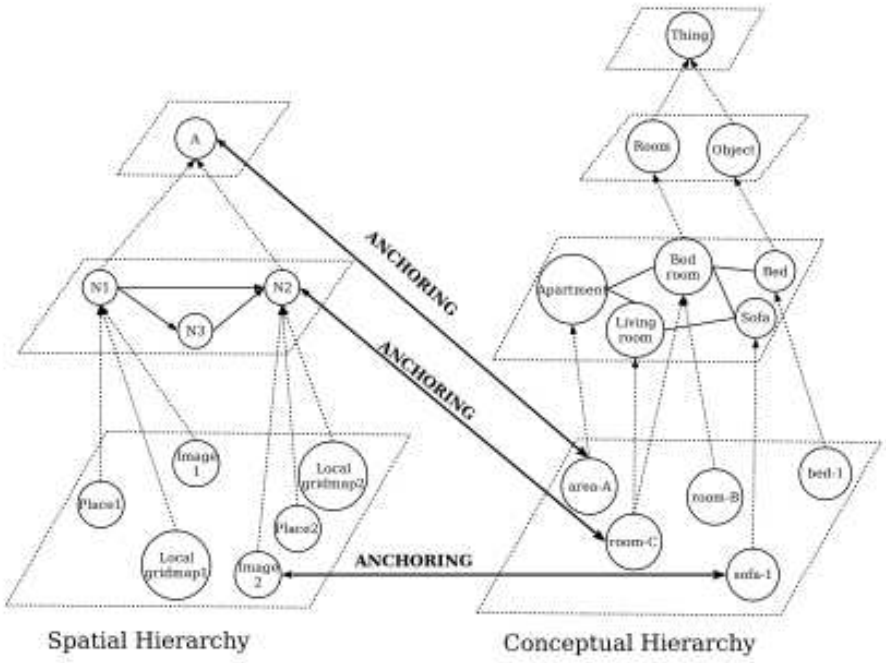


FIGURE 2.1 – *Double structure hiérarchique introduite dans [Galindo 2005] pour modéliser l'environnement. Le modèle représente à la fois l'espace et des concepts plus abstraits.*

FIGURE 2.2 – *Exemple d'utilisation de l'information sémantique pour contraindre la construction 3D ([Nuchter et Hertzerg 2008]). Comprendre la planéité du sol et des murs permet de corriger la dérive de l'odométrie.*

donné dans [Aydemir 2012], où la connaissance de statistiques quant à la topologie des environnements intérieurs permet d'inférer la nature d'une pièce non encore visitées. Enfin, d'autres approches utilisent l'estimation conjointe de la sémantique et de la géométrie pour maximiser la cohérence globale du modèle [Hane 2013]. La communication entre les couches de différentes natures est donc bien exploitée pour la construction des cartes.

Cependant peu d'algorithmes hybrides, c'est à dire exploitant les informations des dif-

férentes couches simultanément, ont été développés pour la navigation. Dans le cadre de la localisation, l'information sémantique a été utilisée de trois manières. Plusieurs méthodes [Yi 2008, Dong 2013, Anati 2012] ont proposé d'estimer la position métrique du robot par rapport à des objets identifiés dans la scène. Les objets sont associés à une position spécifique, comme le sont des points d'intérêt et la localisation se fait de manière classique. L'information sémantique ne sert ici qu'à définir des amers visuels en remplacement et pas en complément de l'information spatiale usuelle. D'autres méthodes [Vasudevan 2008] ont proposé une localisation topologique basée sur l'information sémantique. Elles consistent à inférer la nature du lieu où se trouve le robot en fonction des objets observés dans la scène. Cette stratégie repose sur l'hypothèse qu'on peut toujours trouver des objets qui caractérise un lieu particulier, ce qui n'est pas garanti dans le monde réel et rarement vrai en extérieur. Une troisième méthode est présentée dans [Filliat 2007] où des images sont classées en fonction des mots visuels qui y sont identifiés. Le principal inconvénient de cette méthode est qu'elle repose sur un apprentissage supervisé de la carte plutot que des objets, ce qui est difficile à concevoir pour des environnements de grande dimension.

L'information sémantique a aussi été utilisée de plusieurs manières pour la planification de trajectoire. En premier lieu, elle est utilisée pour définir les actions du robot en remplaçant dans un graphe topologique les relations entre les lieux par des actions décrites à haut niveau [Ulh 2011, Posada 2014, Borkowski 2010]. De ce fait la trajectoire peut être décrite par un ensemble d'actions effectuées par le robot. La seconde manière d'utiliser l'information sémantique qui ait été proposée consiste à remplacer une position à atteindre, décrite de façon métrique, par une description sémantique du lieux, le plus souvent un simple label associé à une pièce ou un objet [Nuchter et Hertzerg 2008, Galindo 2011]. L'intérêt principal de ces approches est de fournir des algorithmes indépendants du robot ou du capteur utilisé. Par exemple la consigne "sortir de la pièce" peut être adaptée à n'importe quelle plateforme. La sémantique est donc utilisée comme un moyen de définir des algorithmes génériques. De manière plus générale, l'information sémantique a aussi montré son utilité [Galindo 2011] pour améliorer l'efficacité de la planification de tâches et étendre les capacités de planification d'un système en inférant l'existence d'instances partiellement observées.

Cependant, il est surprenant de constater que l'information sémantique n'a été que très peu utilisée pour améliorer les performances de navigation des robots. Dans le cas de la localisation, l'information sémantique est le plus souvent utilisée pour la localisation topologique, mais sous des contraintes fortes, ou pour remplacer l'information métrique au sacrifice de la précision, la position des objets n'étant estimée qu'approximativement [Anati 2012]. Pour la planification de trajectoire, elle n'a été que très brièvement utilisée comme un moyen de contraindre la trajectoire du robot, par exemple dans [Siemiatkowska 2011] où elle permet de pondérer l'intérêt d'un chemin en fonction de la nature du sol (route, herbe etc). La plupart des approches conservent des contraintes métriques pures pour la planification de trajectoire. Pour la cartographie, l'information sémantique est essentiellement utilisée comme une information additionnelle permettant de contraindre la

construction des modèles, que ce soit au niveau topologique ou métrique mais elle n'a que peu été utilisée pour son fort pouvoir expressif et discriminant. Par ailleurs, les modèles intégrant de l'information sémantique sont assez mal adaptés à la modélisation d'environnement extérieurs. Soit il s'agit de cartes métriques annotées, qui présentent les mêmes inconvénients que les cartes métriques pures mais pour un volume de données accru. Soit ces modèles reposent sur des concepts mal définis en extérieur comme le concept de lieu. Les algorithmes de navigation qui en découlent se trouvent donc eux aussi assez mal adaptés à ces environnements. L'information sémantique n'a pratiquement pas non plus été utilisée pour résoudre des problèmes difficiles à traiter par des méthodes classiques, comme la mise à jour de la carte.

2.2 Limites des représentations actuelles

Le chapitre 1 a permis de souligner l'évolution des représentations spatiales de l'environnement depuis le début des recherches en robotique mobile, de présenter leurs limites et d'expliquer pourquoi ces modèles sont insuffisants pour envisager la navigation autonome de robots dans de grands environnements potentiellement dynamiques. L'ajout d'informations sémantiques dans ces représentations a ouvert de nouvelles perspectives pour la modélisation de l'environnement et la navigation mais elles restent largement sous-exploitées. L'objectif de cette thèse est donc de montrer comment l'information sémantique, en la combinant avec l'information spatiale, peut être utile à la navigation en élargissant son utilisation au delà des limites présentées dans l'état de l'art. En premier lieu sera présenté un nouveau modèle d'environnement résolument orienté navigation, s'appuyant sur la représentation métrique-topologique précédemment développée dans le laboratoire [Meilland 2011]. Ce modèle intégrera des informations de nature métrique, topologique et sémantique dans une structure hiérarchique adaptée spécifiquement à la modélisation de très larges environnements. Ensuite il sera montré comment l'exploitation de l'information sémantique permet la mise à jour de la carte et son extension au delà des limites perceptuelles du robot par un procédé nommé *extrapolation de carte*. Enfin des algorithmes utilisant spécifiquement l'information sémantique pour la navigation seront présentés, exploitant les spécificités de la représentation proposée pour améliorer les performances des robots et étendre leur domaine d'application à des environnements complexes arbitrairement grands.

Deuxième partie

Cartographie Hybride Métrique-Topologique-Sémantique

Préambule

La cartographie d'environnements dynamiques de grandes dimensions présente plusieurs défis pour les systèmes intelligents amenés à y évoluer de manière autonome. Outre les problèmes de dérive de l'odométrie, la vaste quantité de données que contiennent ces modèles est un problème majeur à la fois pour leur construction et leur utilisation. La localisation, la fermeture de boucle ou encore la planification de trajectoire sont autant de tâches qui demandent un accès rapide aux données de la carte. Plus celle-ci est grande, plus le temps nécessaire pour retrouver une information est long, ce qui dégrade les performances du robot. Une alternative à la construction de modèles 3D, bien adaptée aux grands environnements consiste à utiliser une série d'images mémorisées et référencées globalement comme modèle de l'environnement. Ces méthodes fournissent un maximum de précision locale car l'information est exprimée directement dans le référentiel du capteur d'acquisition et donc, est exempt d'erreurs liées aux algorithmes de reconstruction et de changements de repères. Les images peuvent alors être utilisées pour la localisation visuelle dans le voisinage de la base de données en recherchant celle qui est la plus proche de l'image courante. Il existe plusieurs méthodes de localisation à partir d'images. Dans [Royer 2005], une base de données est construite lors d'une phase d'apprentissage. Des points de Harris et leurs positions 3D sont extraits de toutes les images et sont utilisés pour localiser en ligne une caméra en utilisant la mise en correspondance des points d'intérêt. Une méthode similaire mais basée sur un graphe d'images sphériques est présentée dans [Courbon et al. 2009] où les images omnidirectionnelles de la base construite sont utilisées pour le suivi de trajectoire. Le problème des méthodes basées points d'intérêt est qu'elles sont en général sensibles au changement de point de vue, l'extraction de ces points étant souvent mal conditionnée et bruitée. D'autres méthodes utilisant une mémoire d'images pour réaliser un asservissement visuel ont aussi été proposées. Par exemple dans [Mezouar et Chaumette 2003] au lieu d'estimer la position 3D de points d'intérêt un asservissement visuel est réalisé en fonction de l'erreur de re-projection dans l'image. L'inconvénient majeur de ce genre de techniques est qu'elles ne permettent pas de s'éloigner de la trajectoire de référence réalisée lors de l'acquisition, ce qui les rend peu applicable pour la navigation. Une méthode de localisation visuelle directe basée sur la minimisation itérative de l'erreur d'intensité entre tous les pixels de l'image a été introduite dans [Meilland 2011]. Elle permet la localisation robuste à partir d'une base de données d'images sphériques préalablement construite. Il s'agit sans doute d'une des méthodes les plus aboutie qui sera utilisée et étendue dans ce travail de thèse.

Bien qu'adaptées à de grands environnements, les méthodes basées images souffrent cependant de plusieurs inconvénients. La localisation n'est possible que si l'image courante est proche d'une des images de référence. Il faut un moyen de rechercher dans la base de données l'image présentant le plus de similitudes avec l'image courante. Ceci est d'autant plus compliqué que la carte devient grande. De plus, la nature dynamique du monde

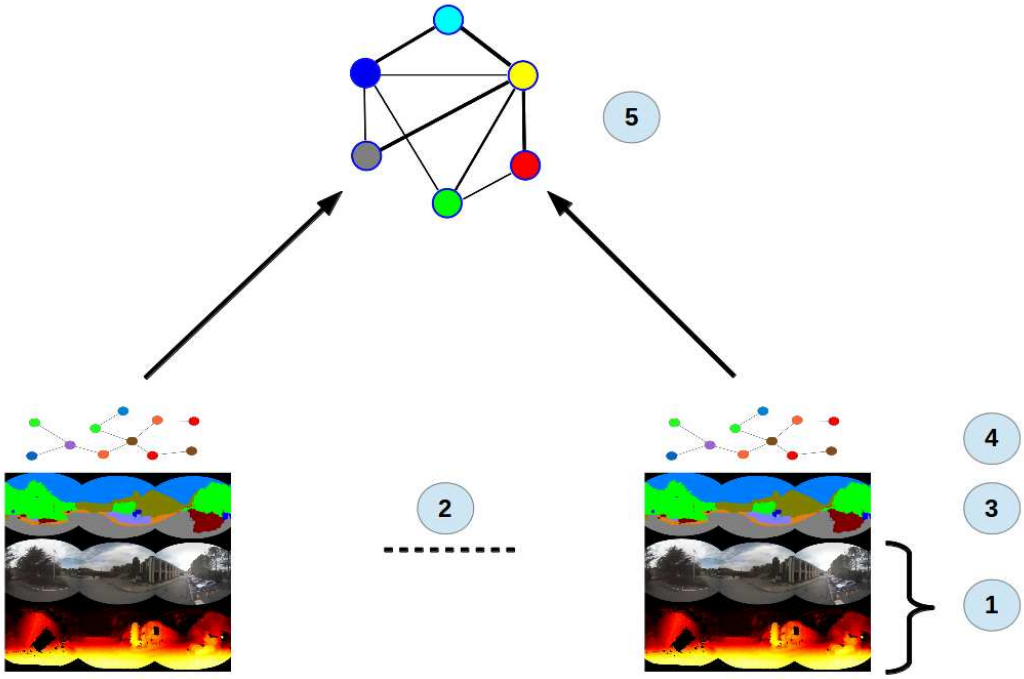


FIGURE 2.3 – *Principales fonctions de cartographie intervenant dans le processus de construction de la carte.*

réel conduit souvent à l'obsolescence rapide des représentations statiques basées sur des mémoires images. Les changements intervenant dans la scène tendent à faire diverger le modèle des observations ce qui peut conduire à l'échec des algorithmes de navigation. Un scénario réaliste de déploiement de robots dans l'environnement quotidien impose donc de développer un modèle qui permette de représenter l'information utile à la navigation de façon compacte et de concevoir des mécanismes de mise à jour des représentations internes du robot.

Cette partie présente une carte dont l'architecture est spécifiquement adaptée à la navigation dans de grands environnements dynamiques. La représentation hybride multi-niveaux proposée intègre des informations de nature photométrique, géométrique, topologique et sémantique qui sont exploitées à des niveaux différents pour permettre la réalisation des fonctions de navigation, y compris dans des environnement dynamiques. La construction de cette carte hybride s'appuie sur un jeu de fonctions de cartographie $M_l(.) : \mathbb{O}_I \rightarrow \mathbb{O}_O, l = 1, 2, ..., L$ dont chacune capture un aspect particulier de l'environnement au travers de l'analyse d'attributs spécifiques, pour lesquels $\mathbb{O}_I$ et $\mathbb{O}_O$ forment respectivement le domaine de définition et le domaine image. Dans le cadre de cette thèse, le nombre des processus intervenant dans la construction de la carte est $L = 8$, correspondant successivement à : (1) la construction de représentation sphériques locales de l'environnement, (2) la construction d'un graphe de ces cartes locales, (3) l'annotation des

images sphériques, (4) la construction d'une description sémantique locale compacte, nommée *graphe sémantique* et (5) la construction d'une représentation sémantique globale non métrique, nommée graphe conceptuel. A cela s'ajoutent trois processus exploitant l'information sémantique et contextuelle pour améliorer la cartographie de l'environnement (6) en étendant par le raisonnement la zone cartographiée au delà des limites perceptuelles du robot, (7) en mettant à jour les données et (8) en corrigeant automatiquement les erreurs dans les graphes sémantiques. Les chapitres suivant détaillent chacun de ces processus.

CHAPITRE 3

Le modèle des sphères ego-centrées

3.1 Introduction

La construction d'une carte 3D précise et facilement utilisable d'un environnement de grande dimension est un problème complexe, d'une part parce que les erreurs de reconstructions deviennent non négligeables à grande échelle et d'autre part parce que la réutilisation de la carte demande un accès rapide aux données. Un modèle basé sur un ensemble de représentations locales sous forme d'images sphériques augmentées de la profondeur a été proposé dans [Meilland 2011]. Cette thèse s'appuyant sur ce travail, ce modèle est rappelé dans ce chapitre. L'utilisation d'une représentation sphérique offre de nombreux avantages. Elle permet notamment de créer un modèle compact de l'environnement local puisqu'une seule image suffit à décrire un point de vue. Elle facilite également la fermeture de boucle du fait de l'invariance de la représentation par rotation.

L'idée d'utiliser une représentation à base d'images sphériques a été proposée à plusieurs reprises dans la littérature [Nayar 1997, Baker 2001, Meilland 2010]. Pour contourner l'incapacité de construire des capteurs réellement sphériques, plusieurs méthodes ont été développées pour l'acquisition d'images et de données de profondeur. Deux techniques sont principalement utilisées pour l'acquisition de données photométriques : les caméras catadioptriques omnidirectionnelles et les systèmes multicaméras. Les caméras catadioptriques omnidirectionnelles [Nayar 1997] sont en général des caméras perspectives ou orthographiques devant lesquelles sont ajoutés des miroirs convexes, le plus souvent hyperboliques ou paraboliques, alignés sur l'axe optique. Ce type de caméra peut être modélisé par une projection unifiée [Barreto 2006, Mei et Rives 2007] qui consiste à effectuer deux projections successives, d'abord sphérique puis perspective comme illustré à la figure 3.1. L'un des inconvénients de ces caméras omnidirectionnelles est que le champ visuel est projeté sur un capteur unique, la résolution est donc faible. De plus, la résolution spatiale est non uniforme, la qualité de l'image diminue donc en direction des bords du capteur. Enfin, le champ visuel est souvent limité verticalement à la demie sphère inférieure ce qui n'est pas idéal pour la modélisation d'environnement urbains où une part importante des zones saillantes et stables de l'environnement se trouve en hauteur. Une sphère visuelle peut aussi être construite à partir d'un système multi-caméras en assemblant plusieurs images. Dans [Baker 2001] des sphères sont construites à partir d'images capturées simultanément par plusieurs caméras reliées rigidement alors que [Lovegrove et Davison 2010] proposent de

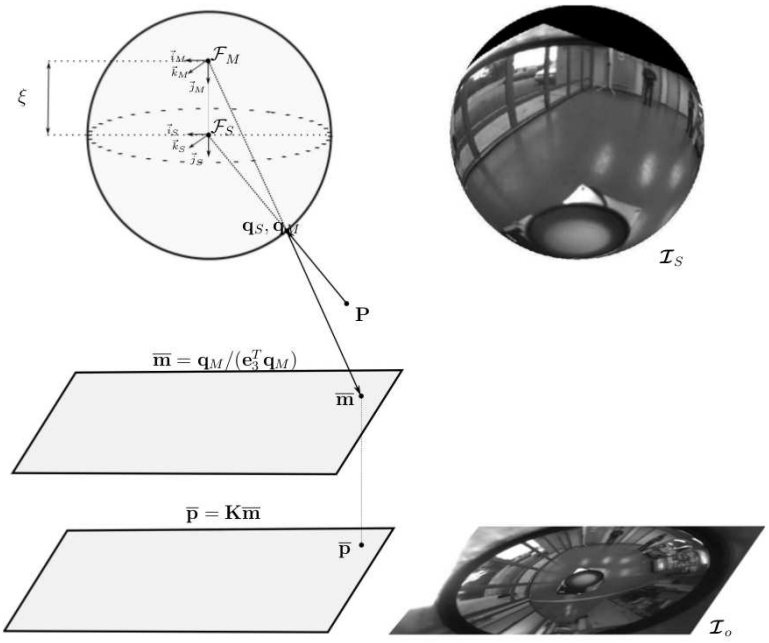


FIGURE 3.1 – *Modèle de projection unifié [Mei et Rives 2007]*

reconstruire ces sphères à partir d'une séquence d'images acquise par un capteur unique. Dans cette thèse, les images sphériques sont reconstruites à partir des données acquises par un capteur composé de 6 caméras disposées en deux anneaux de trois caméras, comme illustré à la figure 3.2.

Les systèmes d'acquisition des données de profondeur se distinguent en deux groupes : les systèmes actifs et passifs. Un exemple de système actif est donné dans [Gallegos et al. 2010] où l'image sphérique est construite avec une caméra catadioptrique surmontée d'un télémètre laser pour l'acquisition des données de profondeur. Pour propager l'information de profondeur dans l'image, les hypothèses d'un sol plan et d'un environnement structuré composé de murs verticaux sont nécessaires. Le problème de cette approche est, comme évoqué précédemment, la faible résolution des caméras omnidirectionnelles. Une approche similaire a été utilisée dans [Cobzas 2003] mais au lieu d'une caméra omnidirectionnelle c'est une camera perspective couplée avec un laser qui est montée sur une tourelle pan/tilt. Mais ce type de système n'est pas compatible avec un environnement dynamique ou un véhicule en mouvement du fait de sa lenteur. Une autre approche active pour l'acquisition de données de profondeur consiste à utiliser un projecteur de lumière structurée, comme dans les très populaires caméras RGBD *Kinect* ou *Xtion*. Ces systèmes permettent d'acquérir des données de profondeur en temps réel et sont utilisés pour le SLAM [Audras 2011]. Cependant, ces systèmes sont réservés aux usages intérieurs, leur utilisation étant rendu impossible en extérieur par l'éblouissement du capteur et dans une moindre mesure, la précision des mesures sur de longues distances. Les systèmes passifs quant à eux utilisent

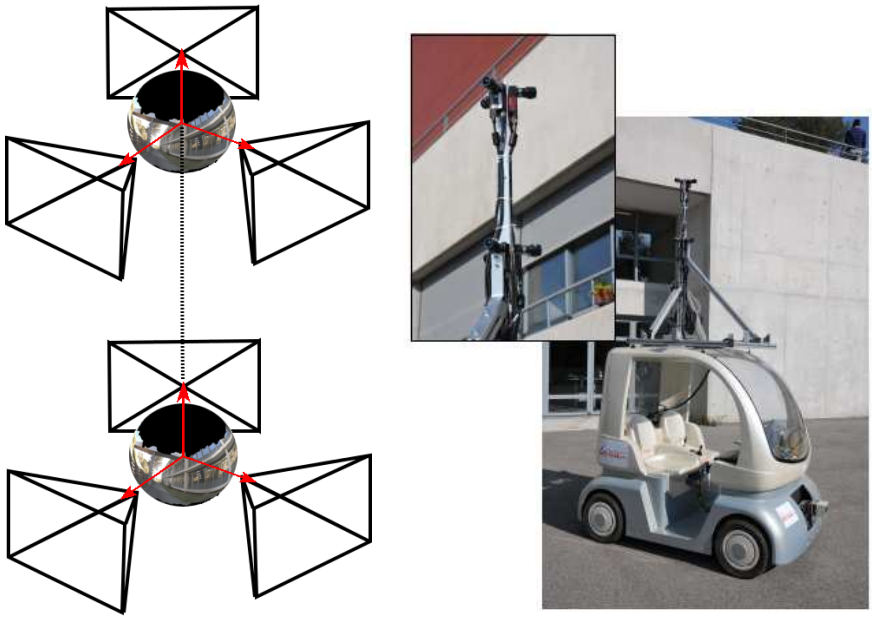


FIGURE 3.2 – *Capteur utilisé pour l'acquisition des principales données utilisées dans cette thèse.*

la vision stéréoscopique pour produire une estimation de la profondeur. Dans [Li 2006], des techniques de mise en correspondance dense stéréo sont appliquées à deux images sphériques pour reconstruire la profondeur. Plus récemment [Kim et Hilton 2009] a utilisé deux caméras mono-dimensionnelles pivotantes pour construire des images sphériques puis appliquer des méthodes de mise en correspondance dense pour obtenir la profondeur. Le problème est comme précédemment, que les capteurs mobiles ne sont pas adaptés aux scènes dynamiques ni aux véhicules en mouvement. Ils sont donc de peu d'utilité dans le cas présent.

3.2 Synthèse de nouvelle vue

L'un des problèmes des mémoires visuelles est que les images qui les constituent sont acquises depuis un point de vue particulier. Or le robot utilisant la carte est amené à se déplacer dans le voisinage de la base de données. Pour pouvoir estimer l'apparence de l'environnement à des positions différentes de celles où ont été acquises les images, la méthode de synthèse de nouvelle vue est utilisée. Elle consiste à déformer une image acquise à une position donnée de manière à simuler l'apparence de l'environnement depuis un autre point de vue. Ceci est particulièrement utile pour comparer deux images acquises depuis deux points de vue différents en ramenant l'une d'elles dans le référentiel de l'autre.

Soit $\mathcal{I}^*$ une image sphérique de référence de dimension $m \times n$ associée à une fonction d'intensité $\mathcal{I}^*(\boldsymbol{p}^*)$ et un référentiel $\mathcal{R}^*$. Un pixel dans l'image $\mathcal{I}^*$ est identifiée par sa

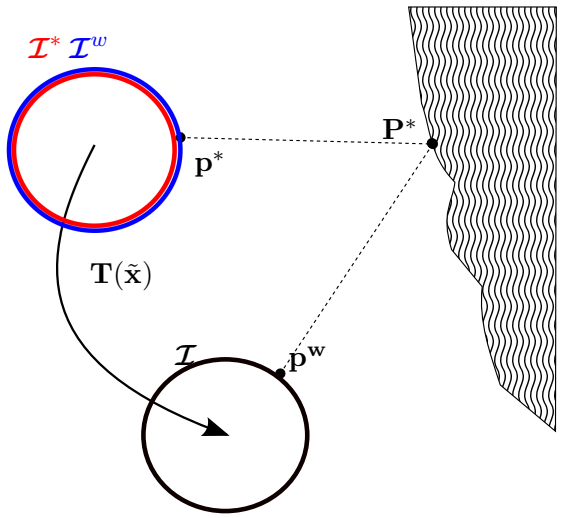


FIGURE 3.3 – *Synthèse de nouvelle vue. L'information photométrique de $\mathcal{I}$ est importée à la position de $\mathcal{I}^*$ en synthétisant une nouvelle image $\mathcal{I}^\omega$.*

position $\boldsymbol{p}^* = (u, v)$, avec $u \in [0, m[$ et $v \in [0, n[$. Un point 3D de l'espace Euclidien est défini par $\boldsymbol{P} = \{\boldsymbol{p}^*, Z^*\}$ où $Z^* \in \mathbb{R}^+$ est la profondeur exprimée dans le repère de $\mathcal{I}^*$. Une image sphérique augmentée de la profondeur est définie par $\mathcal{S}^* = \{\mathcal{I}^*, Z^*\}$.

Soit $\mathcal{I}$ une seconde image sphérique associée à un référentiel $\mathcal{R}$ dont la position exprimée dans $\mathcal{R}^*$ est $\boldsymbol{T}(\boldsymbol{x})$. Une nouvelle image $\mathcal{I}^\omega$ contenant l'information photométrique de $\mathcal{I}$ peut être synthétisée à la position de $\mathcal{I}^*$ en utilisant la fonction de warping :

$$p^\omega = \omega(\boldsymbol{T}(\boldsymbol{x}); Z^*, \boldsymbol{p}^*) \tag{3.1}$$

qui transfère le pixel $\boldsymbol{p}^*$ de l'image de référence dans le repère de la nouvelle image $\mathcal{I}^\omega$ par la transformation rigide $\boldsymbol{T}(\boldsymbol{x})$ suivie d'une projection sphérique. La fonction d'intensité associée à la nouvelle image $\mathcal{I}^\omega$ est alors donnée par :

$$\mathcal{I}^\omega(\boldsymbol{p}^*) = \mathcal{I}(\omega(\boldsymbol{T}(\boldsymbol{x}); Z^*, \boldsymbol{p}^*)) \tag{3.2}$$

Les positions des points p^ω ne correspondant pas exactement aux pixels de $\mathcal{I}$, l'interpolation bilinéaire est utilisée pour calculer l'intensité correspondante.

3.3 Construction d'images sphériques

Pour construire une image sphérique avec un système multi caméras, plusieurs images doivent être recalées, projetées et fusionnées sur une sphère virtuelle tangente aux capteurs. Pour une caméra i de matrice de paramètres intrinsèque $\boldsymbol{K_i}$, le point de la sphère unitaire

$q_s \in S^2$ est projeté dans l'image par projection perspective :

$$\overline{\boldsymbol{p}} = \frac{\boldsymbol{K_i R_i q_s}}{\boldsymbol{e_3^T K_i R q_s}} \tag{3.3}$$

où $\boldsymbol{R_i}$ est la matrice représentant la rotation de la caméra i par rapport à la sphère et où $\boldsymbol{e_3^T}$ permet d'extraire la troisième composante du vecteur au dénominateur. Les N images $\mathcal{I}_i$ sont alors projetées sur la sphère puis fusionnées via une fonction de warping des intensités des images perspectives vers l'image sphérique $\boldsymbol{\mathcal{I}}$:

$$\boldsymbol{\mathcal{I}}(\boldsymbol{q}_s) = \sum_{i=0}^{N} \alpha_i \boldsymbol{\mathcal{I}}_i(\omega(\boldsymbol{K_i}, \boldsymbol{R_i}, \boldsymbol{q}_s)) \tag{3.4}$$

où les paramètres α_i sont les coefficients de fusion des intensités. Il convient de noter que la translation entre les caméras n'est pas prise en compte car l'information de profondeur n'est pas disponible. Ceci revient à supposer que les centres de projection sont confondus et les caméras ne sont donc alignées qu'en rotation. Dans certains cas, par exemple lorsque des objets sont proches du capteurs, des artefacts peuvent apparaître dans l'image sphérique du fait de la parallaxe. Cependant dans le cas de scènes extérieurs, les translations entre les centres de projection optiques sont négligeables.

3.3.1 Systèmes multicameras multibaseline

Le système multi-caméras utilisé dans cette thèse est constitué de deux anneaux de trois caméras, alignées verticalement, chaque caméra couvrant un peu plus de 120^o du champ (voir figure 3.2). Chaque anneau permet de reconstruire une sphère visuelle et la profondeur est extraite par mise en correspondance dense des deux sphères. Seuls les principes de construction de cette image sphérique sont présentés ci-dessous et le lecteur est renvoyé à [Meilland 2012] pour plus de détails.

3.3.1.1 Calibration

Avant de pouvoir faire la mise en correspondance des sphères et donc reconstruire les sphères augmentées, il est nécessaire de calibrer le capteur pour déterminer les paramètres extrinsèques et intrinsèques des caméras. La première phase de l'étalonnage consiste à calibrer chaque caméra séparément. Le processus utilisé est standard si ce n'est qu'une mire spéciale de forme asymétrique est utilisée pour déterminer les paramètres intrinsèques de chaque caméra. Dans un second temps, ce sont les paramètres extrinsèques de chaque anneaux de caméras qui sont estimés. Chaque caméra de l'anneau supérieur observe la même zone que la caméra correspondante de l'anneau inférieur ce qui permet de créer des paires stéréo entre les deux anneaux. La méthode de calibration utilisée repose sur la fermeture de boucle séparée de chacun des anneaux. Pour trois images $\boldsymbol{\mathcal{I}}_1, \boldsymbol{\mathcal{I}}_2$ et $\boldsymbol{\mathcal{I}}_3$ d'un

même anneau, l'erreur à minimiser s'écrit :

$$e^0 = \begin{pmatrix} e_1 = \boldsymbol{\mathcal{I}}_2(\omega(\widehat{\boldsymbol{R}_2\boldsymbol{R}}(\boldsymbol{x}_2), \boldsymbol{K}_2), \boldsymbol{q}_S) - \boldsymbol{\mathcal{I}}_1(\omega(\boldsymbol{Id}, \boldsymbol{K}_1), \boldsymbol{q}_S) \\ e_2 = \boldsymbol{\mathcal{I}}_3(\omega(\widehat{\boldsymbol{R}_3\boldsymbol{R}}(\boldsymbol{x}_3), \boldsymbol{K}_3), \boldsymbol{q}_S) - \boldsymbol{\mathcal{I}}_1(\omega(\boldsymbol{Id}, \boldsymbol{K}_1), \boldsymbol{q}_S) \\ e_3 = \boldsymbol{\mathcal{I}}_3(\omega(\widehat{\boldsymbol{R}_3\boldsymbol{R}}(\boldsymbol{x}_3), \boldsymbol{K}_3), \boldsymbol{q}_S) - \boldsymbol{\mathcal{I}}_2(\omega(\widehat{\boldsymbol{R}_2\boldsymbol{R}}(\boldsymbol{x}_2), \boldsymbol{K}_2), \boldsymbol{q}_S) \end{pmatrix} \tag{3.5}$$

où les matrices $\boldsymbol{K}_i$ sont les paramètres intrinsèques de chaque caméras et les $\boldsymbol{R}(x_i)$ sont les rotations estimer. La première images est fixée arbitrairement comme référence. Les paramètres $\boldsymbol{x_2}, \boldsymbol{x_3}$ sont ensuite calculés en utilisant l'algorithme de Gauss-Newton. Lorsque les sphères supérieures et inférieures ont été reconstruites, les images sphériques sont rectifiées pour permettre la mise en correspondance dense entre le sphères.

3.3.1.2 Matching et triangulation sphérique

La mise en correspondance des sphères à été faite par l'algorithme Efiicient Large Scale Stereo Matching [Geiger et al. 2010] qui offre les meilleurs résultats parmi les méthodes testées. L'information de profondeur est obtenue par triangulation entre les deux sphères photométriques en utilisant la carte de disparité dense fournie par algorithme matching. Pour un point 3D $\boldsymbol{\mathcal{P}}$ de coordonnées $q_t = (\theta_t, \phi_t)$ et $q_b = (\theta_b, \phi_b)$ dans les sphères supérieur $\boldsymbol{\mathcal{S}}_t$ et inférieure $\boldsymbol{\mathcal{S}}_b$ respectivement, la disparité s'écrit $d_\phi = \phi_b - \phi_t$ (voir figure 3.4). La distance ρ_t, associée au point q_t s'écrit alors :

$$\rho_t = t_y \frac{cos(\phi_t)}{sin(d_\phi)} \tag{3.6}$$

où t_y est la baseline entre les sphères rectifiées. Le point $\boldsymbol{\mathcal{P}}$ est alors définie par :

$$\boldsymbol{\mathcal{P}} = \rho \begin{bmatrix} sin(\theta_t)cos(\phi_t) \\ sint(\phi_t) \\ cos(\theta_t)cos(\phi_t) \end{bmatrix} \tag{3.7}$$

La sphère visuelle augmentée de la profondeur est alors définie par :

$$\mathcal{S} = \{\boldsymbol{\mathcal{I}}, Z\} \tag{3.8}$$

où $\boldsymbol{\mathcal{I}}$ est la sphère photométrique et Z la carte de profondeur associée. La construction de cette représentation correspond à la première fonction de cartographie $M_1(.) : \mathbb{O}_{\mathcal{I}} \times \mathbb{O}_Z \rightarrow \mathbb{O}_{\mathcal{S}}$, où $\mathbb{O}_{\mathcal{I}}$ est l'ensemble des images RGB, $\mathbb{O}_Z$ est l'ensemble des images de profondeur et $\mathbb{O}_{\mathcal{S}}$ l'ensemble des sphères RGBD.

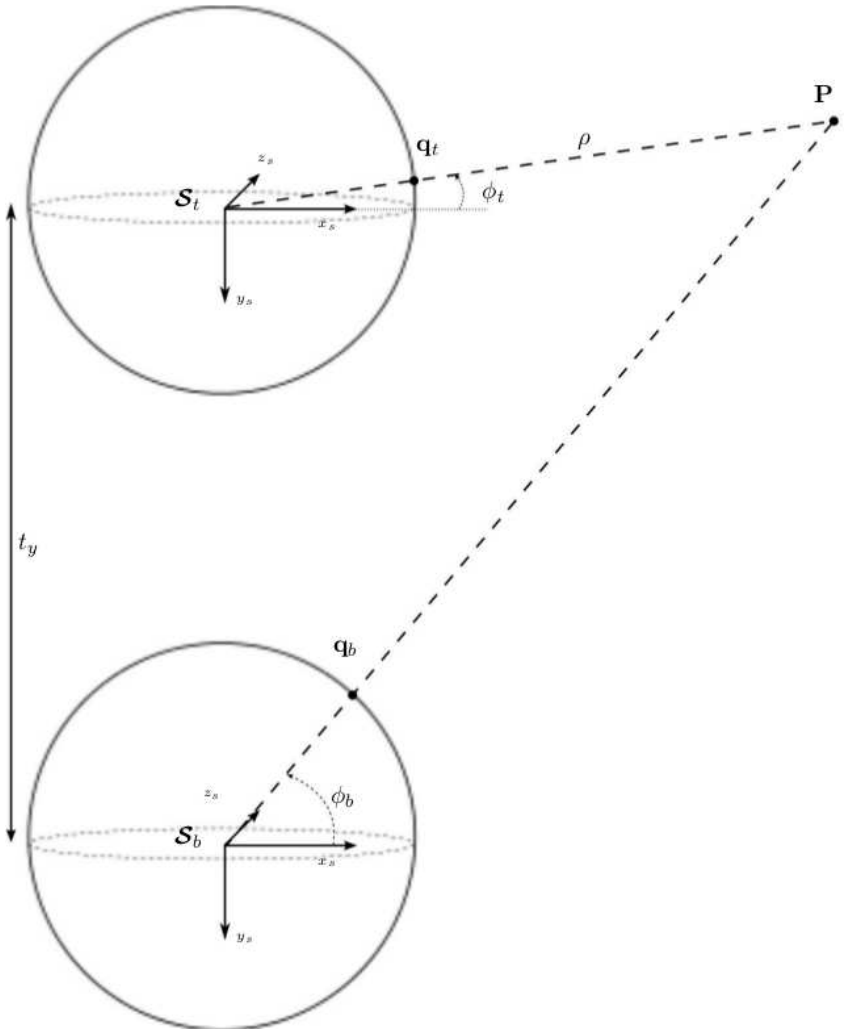


FIGURE 3.4 – *Triangulation sphérique*

3.4 Odométrie visuelle

Les sphères visuelles augmentées représentent l'environnement local du robot. Pour modéliser l'environnement à plus grande échelle, il faut construire un graphe global couvrant tout l'espace en repositionnant les sphères les unes par rapport aux autres. On note $\mathcal{G}$ ce graphe défini par :

$$\mathcal{G} = \{\mathcal{S}_1, ...\mathcal{S}_n; \mathcal{T}_1, ...\mathcal{T}_n\} \tag{3.9}$$

où les $\mathcal{S}_n$ sont les n sphères augmentées et les $\mathcal{T}_n$ leurs poses relatives. Il est nécessaire d'estimer la pose entre deux sphères successives. En considérant l'image sphérique courante $\mathcal{I}$ et l'image sphérique de référence $\mathcal{I}^*$, la pose relative est calculée en recalant les images sphériques l'une par rapport à l'autre. Pour cela une méthode robuste de minimisation d'erreur est employée. La fonction de coût permettant d'optimiser l'erreur d'intensité entre

deux sphères $\{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{I}}^*\}$ est donnée par :

$$\mathfrak{F}_I = \frac{1}{2}\sum_{i}^{k} \boldsymbol{\Psi}_{hub} \left\| \boldsymbol{\mathcal{I}}(\omega(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); P_i)) - \boldsymbol{\mathcal{I}}^*(\omega(\mathbf{I}d; P_i^*)) \right\|^2, \tag{3.10}$$

où $\omega(.)$ est une fonction de warping qui projette un point 3D P_i du repère associé à $\boldsymbol{\mathcal{I}}^*$ dans celui de $\boldsymbol{\mathcal{I}}$. La pose $\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})$ est une approximation de la vraie transformation $\mathbf{T}(\tilde{\mathbf{x}})$ et $\boldsymbol{\Psi}_{hub}$ est une fonction robuste de pondération de l'erreur donnée par [Huber 1981].

La construction de ce graphe correspond à la seconde fonction de cartographie $M_2(.)$: $\mathbb{O}_{\mathcal{S}} \rightarrow \mathbb{O}_{\mathcal{G}}$, où $\mathbb{O}_{\mathcal{G}}$ est l'ensemble des graphes de sphères.

3.5 Limites de la représentation RGBD

La représentation des sphères ego-centrées présente de nombreux avantages pour la modélisation de grands environnements par rapport aux autres modèles spatiaux, mais aussi certaines limites. Tout d'abord la sphère de référence utilisée pour la localisation est choisie en calculant la distance métrique entre la pose courante et la sphère de référence. Lorsque le nombre de sphères dans la base de données devient grand, il est très difficile de choisir la bonne sphère de référence en un temps raisonnable. Il faut donc disposer d'un moyen de sélection efficace. Ensuite, l'utilisation de données proches du capteurs pour la mise en correspondance des images, présente l'inconvénient d'être peu robuste aux variations des conditions d'observation de la scène, dues par exemple, à la présence d'objets dynamiques. La carte est donc difficilement réutilisable sur le long terme. Il a donc été décidé de faire évoluer la représentation en ajoutant une couche sémantique aux sphères. L'objectif de cette partie est de déterminer comment intégrer cette information dans une carte de sphère augmentée en vue de l'utiliser résoudre des problèmes liés à la navigation dans de grands espaces qui ne sont pas, ou mal traités par des méthodes classiques.

CHAPITRE 4

Extraction d'information sémantique

4.1 Introduction

Un des problèmes importants de la vision artificielle est de savoir comment identifier sans ambiguïté des objets dans une image. C'est un problème complexe qui fait appel à la fois à la perception et à l'apprentissage. On peut distinguer deux tâches liées à ce problème. La première consiste à identifier un objet préalablement observé. Elle fait appel à la capacité à se remémorer un ensemble de perceptions, on parle donc de *reconnaissance* d'objet. La seconde consiste à identifier la *nature* d'un objet à partir de la mémorisation de perceptions liées à un groupe d'objets similaires précédemment observés. Elle fait appel à la capacité à associer un stimulus visuel à des stimuli différents mais présentant des similarités. On parle de *classification*. Les deux tâches sont proches mais la seconde demande une capacité de généralisation supplémentaire puisque les instances observées ne sont pas les mêmes que celles ayant produit les stimuli mémorisés. Dans la continuité de l'approche dense utilisée jusqu'ici, on va chercher à *classer* le contenu d'une image sphérique en attribuant à chaque pixel un label. Cette fonction est réalisée par un processus de cartographie $M_3 : \mathbb{O}_{\mathcal{S}} \rightarrow \mathbb{O}_{\mathcal{L}}$ où $\mathbb{O}_{\mathcal{S}}$ matérialise l'espace des images sphériques et $\mathbb{O}_{\mathcal{L}}$ celui des images annotées. Le modèle des sphères augmentées des labels est alors défini par :

$$\mathcal{S} = \{\boldsymbol{\mathcal{I}}, Z, \mathcal{L}_S\} \tag{4.1}$$

où $\mathcal{L}_S$ correspond aux labels attribués à chaque pixel.

Le processus de classification est en général réalisé en deux étapes. Dans une première phase, une représentation de l'image est calculée pour extraire les textures, les couleurs et les formes. Puis, dans une seconde phase, cette représentation intermédiaire est analysée par un algorithme de classification à proprement dit, qui associe un label à chaque pixel. La représentation doit permettre un compromis entre précision et dimension : elle doit être suffisamment expressive pour permettre de discriminer les différentes classes tout en étant suffisamment compacte pour limiter la quantité de données à traiter et permettre de classer les images en un temps compatible avec leur utilisation pour la navigation. L'algorithme de classification doit, quant à lui, prendre en compte le contexte spatial pour l'attribution des labels aux pixels. Ceci permet en outre d'atténuer le bruit auquel sont sujets les pixels qui peut dégrader le résultat final. Il est de plus nécessaire que l'algorithme de classification permette une bonne généralisation. Dans des scènes réelles, la variance intra-classe peut

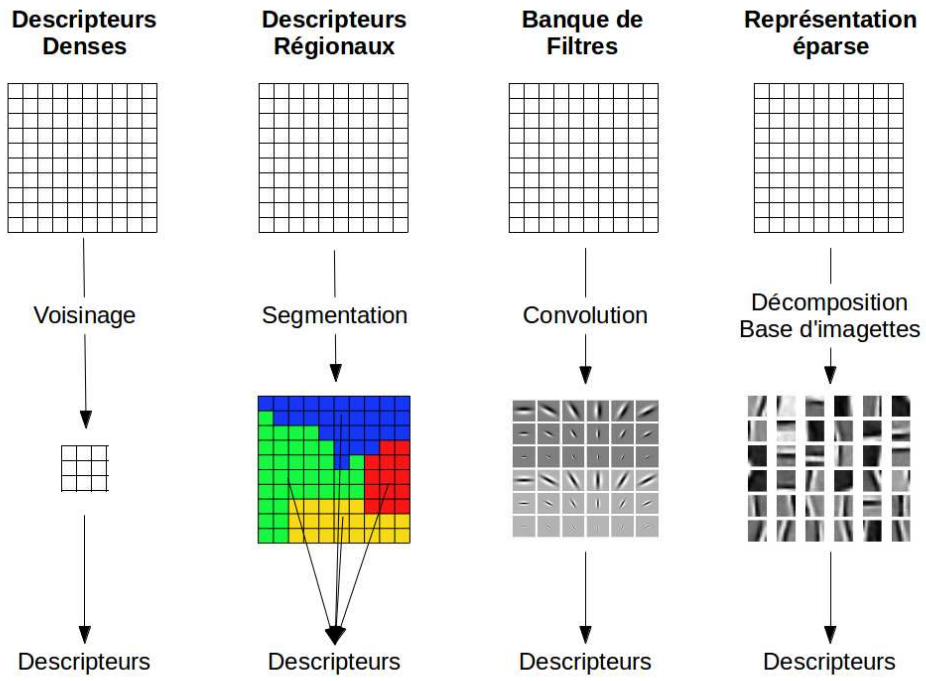


FIGURE 4.1 – *Les représentations de l'image les plus courantes pour la classification.*

être forte et la capacité d'adaptation de l'algorithme d'apprentissage est fondamentale pour garantir des résultats cohérents.

Dans la littérature beaucoup de solutions ont été proposées pour annoter automatiquement le contenu des images, à la fois pour la représentation et la classification de celles-ci. On distingue au moins quatre types de représentations différentes pour l'image, illustrées à la figure 4.1. La première consiste à calculer de manière dense un descripteur correspondant le plus souvent à un histogramme de gradients, calculé sur un voisinage du pixel. C'est le cas de SIFT, SURF ou encore HoG. Cette représentation est très riche mais la quantité de descripteurs, un par pixel, conduit généralement à des temps de classification relativement long. C'est pourquoi certaines approches ajoutent une première étape de segmentation de l'image et classent ensuite les régions au lieu des pixels [Cadena 2013]. L'attribution d'un label global pour chaque région suppose une segmentation parfaite, ce qui n'est évidement pas le cas dans la pratique. Les images sont donc volontairement sur-segmentées pour assurer la cohérence locale du label. Mais le gain de temps obtenu en réduisant le nombre d'entités à classer est d'autant plus faible que le temps nécessaire à la segmentation de l'image est grand. L'utilité de cette approche dépend donc fortement du reste de la chaine de traitement.

Une autre famille de méthode utilise un vecteur descripteur correspondant à la réponse d'une banque de filtres en chaque pixel. C'est le cas des textons [Leung 2001] qui donnent une description dense de l'occurrence des textures dans l'image. Un des avantages de ces méthodes est la possibilité d'utiliser des filtres de différentes tailles pour modéliser la texture locale à différentes échelles, ce qui permet de prendre en compte le contexte local implicitement. Enfin d'autres approches, inspirées de l'architecture du cerveau, apprennent automatique la forme du descripteur. L'image est décomposée en une combinaison linéaire d'imagettes de références, formant un dictionnaire [Bo 2011]. Chaque champ du descripteur représente alors l'importance de l'imagette correspondante dans la reconstruction de l'image locale. L'apprentissage automatique des descripteurs est particulièrement intéressant puisqu'il permet de résoudre le problème ardu de la conception des descripteurs. Mais il introduit aussi un nouveau problème qui est celui de la sensibilité des résultats par rapport à la base d'apprentissage. Les descripteurs comme SIFT ou HOG ont montré leur efficacité dans une très grande variété de situations mais ici, les imagettes du dictionnaire sont calculées pour une base d'apprentissage donnée. Si celle-ci change, le processus d'apprentissage doit être relancé. Ces approches sont donc très bien adaptées à des utilisations pour lesquelles de très grandes bases de données sont disponibles, sur internet notamment, mais plus difficile à envisager pour des robots pouvant être rapidement déployés dans des environnements variés.

Quelque soit la représentation calculée, il faut ensuite utiliser un algorithme de classification pour attribuer les labels aux pixels. Une des principales difficultés est alors de savoir comment prendre en compte l'aspect de l'image sur de grandes échelles. En effet les descripteurs sont le plus souvent calculés localement et ne tiennent pas compte du contexte. Une stratégie répandue consiste à utiliser un modèle graphique probabiliste (MGP) pour tenir compte du voisinage d'un pixel dans l'attribution d'un label (voir section 4.2.3.2). Un MGP est un modèle probabiliste qui représente une distribution de probabilité sur les classes possibles en se factorisant en fonction de la structure d'un graphe associé. Dans ces approches, les pixels sont considérés comme des observations X_i de variables aléatoires Y_i prenant valeur dans $\mathcal{L}$, l'ensemble des classes possibles. Deux formes principales de MGP sont utilisées en classification d'image : les Réseaux Bayésiens et les Champs Aléatoires Markoviens. Les Réseaux Bayésiens encodent la probabilité conjointe à toute les variables aléatoires et les observations $P(Y_1, ..., Y_n, X_1, ..., X_n)$ en utilisant un graphe des dépendances orienté et acyclique. Ces méthodes sont pour la plupart issues du domaine du traitement de la parole et plus généralement du signal 1D [Jia 1999]. Mais des problèmes se posent dans le cas des images car la dépendance d'un pixel en fonction de son voisinage ne peut pas être modélisée correctement par un graphe orienté acyclique. De manière plus importante encore, les modèles basés sur des graphes orientés doivent faire l'hypothèse de l'indépendance statistiques des variables aléatoires, ce qui n'est pas le cas en général, chaque pixel d'une image étant fortement corrélé au pixel voisin. Pour cette raison, les méthodes modernes [Cadena 2013, Byung-soo 2013, Xiong 2010] privilégient les Champs Aléatoires Markoviens (CAM) qui encodent naturellement la relation entre les

pixels en modélisant la probabilité conditionnelle $P(Y|X)$ sur un graphe non dirigé. L'inconvénient majeur de ces modèles est qu'il n'existe pas d'algorithmes d'inférence exacte adapté à la classification d'image. Bien que la plupart des modèles ne considèrent que le voisinage immédiat de chaque pixel, le graphe utilisé pour modéliser les relations entre eux contient des millions de connections. Il est alors impossible d'utiliser des algorithmes exacts pour propager les prédictions dans le graphe. Il faut donc recourir à des méthodes d'inférence approchée. Plusieurs voies sont explorées pour améliorer les performances des MGP. Un modèle graphique hybride entre la représentation Markovienne et Bayésienne, appelé *chain graph*, a été utilisé dans [Pronobis et Jensfelt 2012]. Il repose sur un graphe dont les arêtes peuvent être orientées ou non. L'avantage principal de ce modèle est qu'il permet de représenter des relations plus riches que les modèles markoviens usuels puisque des liens de dépendances directionnels peuvent être modélisés. Mais dans notre cas, l'importance des pixels étant la même, aucune direction de dépendance ne peut a priori être privilégiée. Des algorithmes plus performants pour l'inférence ont aussi été développés, permettant de modéliser les relations à une distance arbitrairement grande entre les pixels et à coût constant [Krähenbühl 2011]. Ceci permet de prendre en compte le contexte quelque soit l'échelle de l'objet, contrairement à la segmentation ou aux méthodes basées sur des filtres, pour lesquelles l'échelle est fixe. Cette méthode s'adapte donc potentiellement mieux aux changements de point de vue. Enfin, une structure comparable aux champs Markoviens a été présentée dans [Munoz 2013] mais au lieu de représenter une distribution de probabilité, la structure est organisée de manière hiérarchique et des interactions entre les différentes échelles de l'image sont possibles. Mais la méthode repose sur la segmentation de l'image pour définir des régions, on peut donc formuler les mêmes critiques que précédemment.

Enfin, il convient de noter qu'une classe importante de méthodes basées sur des architectures profondes inspirées du cerveau humain a été utilisée pour la labelisation avec des performances comparables aux MGP. Il s'agit par exemple des *Deep Boltzman Machines* introduites dans [Salakhutdinov 2009] ou des *Convolutional Neural Network*. Ces approches de type réseaux de neurones (RN) sont souvent, à tort, présentées comme concurrentes des MGP. Il s'agit en fait de concepts orthogonaux, parfaitement compatibles et qu'il serait sans doute avantageux de combiner, comme cela a été fait dans d'autres domaines [Mirowski 2009]. Les travaux futurs, notamment ceux portant sur le cerveau humain [1], permettront sans doute de mieux comprendre le fonctionnement des RN et les liens entre MGP et RN et d'envisager des architectures hybrides, mais cela sort du cadre de cette thèse.

Les raisons de privilégier les méthodes actuelles utilisant les MGP plutôt que les RN tiennent essentiellement en deux points. D'une part, les MGP fonctionnent mieux avec un jeu limité de données et généralisent plus facilement à des cas non observés lors de l'apprentissage. Cette caractéristique est appréciable dans le cas de la mise en œuvre de robot dans des environnements nouveaux pour lesquels peu de données sont disponibles pour

1. Il est fait référence notamment au projet Human Brain Project, https ://www.humanbrainproject.eu/fr

l'apprentissage. D'autres part, et dans une moindre mesure, les MGP encodent l'information d'une manière qui est naturelle et accessible alors que le fonctionnement des réseaux neuronaux, notamment le parcours de l'information, est moins bien comprise. Les méthodes basées sur les MGP ont donc été privilégiées dans cette thèse, comme expliqué dans la section suivante qui détaille la stratégie employée pour la classification des images.

4.2 Pipeline visuel utilisé

Plusieurs contraintes s'appliquent sur notre système de classification d'images. Premièrement, étant destiné à un robot mobile, le système doit permettre d'accéder à l'information sémantique en un temps compatible avec le déplacement du robot. Il est donc exclu de mettre plusieurs minutes pour classer les images. Cependant, la sémantique d'une scène change relativement peu pour un déplacement de quelques mètres, aussi il est inutile pour un robot se déplaçant à une vitesse comparable à l'homme, de classer les images à une fréquence supérieur au hertz. Le temps alloué à la labelisation a donc été fixé à quelques secondes par image. Deuxièmement, dans la continuité de l'approche dense utilisée jusqu'ici, on souhaite attribuer des labels aux objets de l'image avec une précision pixellique. Ceci permet d'envisager d'utiliser les labels pour des tâches de bas niveau comme la mise en correspondance des images ou la reconstruction 3D qui, bien que n'ayant pas été abordées dans cette thèse, le sont dans d'autres travaux du laboratoire. Troisièmement, aucune hypothèse n'est faite quant à la position des objets dans l'image. Ceci permet d'envisager des déplacements en 3D du robot et de mieux généraliser aux cas réels où les objets peuvent potentiellement occuper des positions arbitraires. Enfin, bien que le système de stéréovision permette l'acquisition de données de profondeur, la géométrie n'est pas utilisée dans le processus de labellisation pour plusieurs raisons. D'une part, l'odométrie nécessite l'estimation dense de la profondeur dans l'image. Or les algorithmes qui le permettent ont tendance à produire des données lissées, difficilement exploitable pour la classification. L'information géométrique est dès lors peu intéressante dans la mesure où elle ne permet pas l'identification de structures particulières. D'autre part, dans le cadre de la navigation en extérieur qui nous intéresse principalement ici, les objets qui forment une grande partie de l'image que l'on souhaite labelliser se situent relativement loin du capteur. L'information géométrique n'est donc pas une source d'information fiable dans ce cas.

4.2.1 Architecture globale

L'architecture globale de la chaine de traitement est illustrée à la figure 4.2. Parmi les diverses représentations possibles de l'image, celle utilisée ici repose sur l'extraction dense de descripteurs. Ce choix se justifie pour plusieurs raisons. Premièrement, la redondance de l'information contenue dans les descripteurs voisins, participe à rendre le système robuste à la fois au bruit et aux changements des conditions d'observation. Deuxièmement,

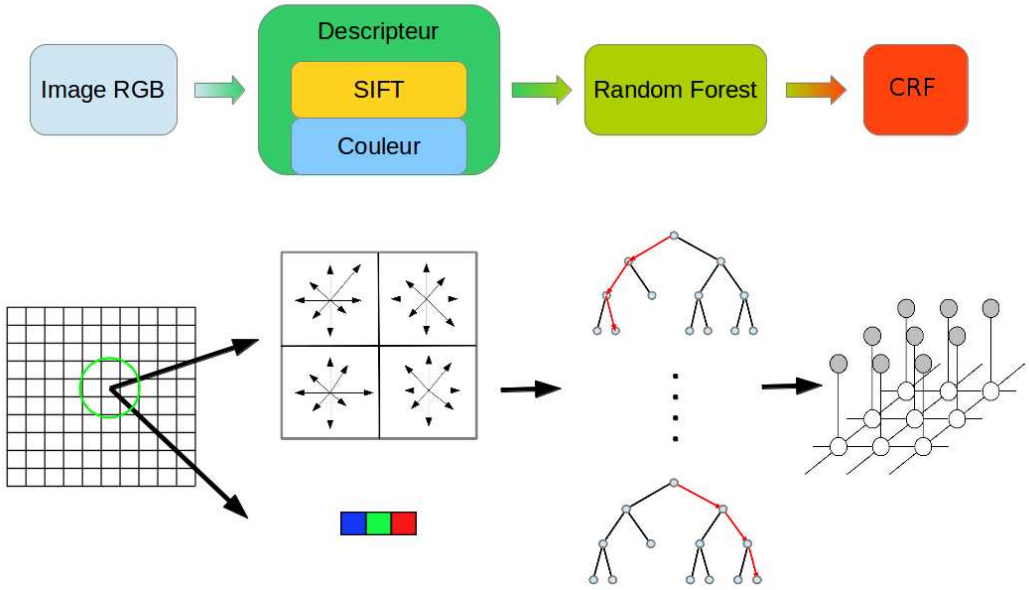


FIGURE 4.2 – *Les trois étapes de la classification. Des descripteurs sont extraits de manière dense de l'image, formés de vecteur SIFT plus de statistiques sur la couleur. Ces descripteurs sont classés par une forêt d'arbres aléatoires qui produisent une distribution de probabilité des labels pour chaque pixel. Enfin un MGP est utilisé pour modéliser les relations entre les pixels et lisser les prédictions.*

des indices théoriques [Bengio 2007] suggèrent que des descripteurs denses ont tendance à améliorer les performances de catégorisation. Parmi la multitude de descripteurs existant, SIFT a largement montré sa robustesse et son pouvoir descriptif. Par ailleurs, le principal obstacle qui s'oppose à l'utilisation d'approches denses, le coût du calcul d'un grand nombre de descripteurs, peut être partiellement surmonté dans le cas de descripteurs denses (voir 4.2.2). Le descripteur utilisé pour la labelisation des images sphériques consiste donc en la concaténation d'un descripteur SIFT et de statistiques concernant la couleur de l'image.

Les scènes réelles comportent un nombre arbitrairement grand d'instances d'une même classe et la base d'apprentissage ne peut évidement pas contenir un exemple de chaque. Un des critères fondamentaux pour évaluer la qualité d'un algorithme de classification est donc sa capacité de généralisation. Elle correspond à la possibilité de modéliser des fonctions complexes, c'est à dire présentant plus de variations que d'exemples disponibles dans la base d'apprentissage. Un faisceau d'indices théoriques et pratiques [Bengio 2007] montrent que la capacité de généralisation est liée à la profondeur de l'architecture employée,

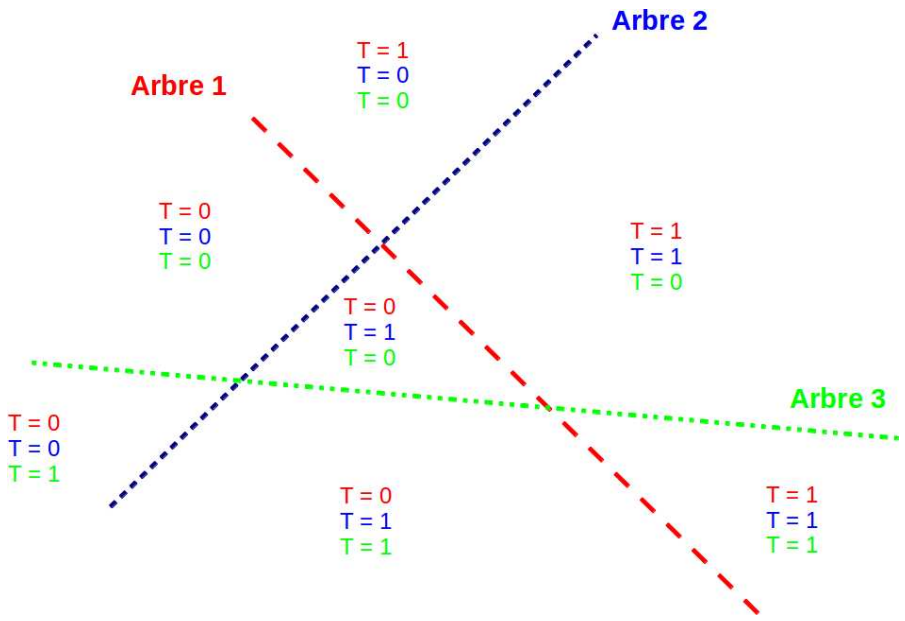


FIGURE 4.3 – *Illustration de la partition de l'espace des solutions par une forêt d'arbres aléatoires. Alors qu'un arbre unique (ici représenté par une simple partition binaire de l'espace pour la clarté du schéma) partitionne l'espace linéairement par rapport au nombre de paramètres, une forêt d'arbres aléatoires peut discriminer l'espace en un nombre de régions exponentiel par rapport au nombre de paramètres.*

c'est à dire au nombre de couches représentant des opérations non linéaires successives. Plus l'architecture est profonde, plus elle est à même de représenter des fonctions complexes et donc de s'adapter aux variations de l'environnement. Pour cette raison, l'étage de classification utilise les *Random Forest* (RF) pour estimer la distribution de probabilité des labels pour chaque pixel. En effet les RF ont une architecture plus profonde que la plupart des algorithmes de classification du fait de l'utilisation d'un grand nombre d'arbre pour produire un vote. Sa profondeur est d'ordre 3. Par ailleurs, les RF permettent de discriminer des régions de l'espace des solutions de manière exponentielle par rapport au nombre de paramètres (figure 4.3), contrairement à beaucoup d'algorithmes pour lesquels le nombre de régions discriminées est linéaire par rapport au nombre de paramètres. Ceci permet notamment de diminuer le temps d'apprentissage. En outre les RF sont très rapides, à la fois dans la phase d'apprentissage et de prédiction ce qui les rend attractives dans le cas du déploiement de robots mobiles dans des environnements nouveaux. Dans une seconde étape, les résultats des RF sont utilisés dans un MGP qui utilise le contexte pour fournir des prédictions corrigées. Le MGP est un Champ Conditionnel Aléatoire (CRF) dont la structure est détaillée à la section 4.2.3.2.

4.2.2 Calcul des descripteurs

Le descripteur SIFT décrit l'apparence locale de l'image par la concaténation d'histogrammes calculés sur 16 fenêtres locales définies autour d'un point d'intérêt. Chacun des histogrammes modélise l'orientation du gradient à l'intérieur de la fenêtre suivant 8 directions. Les valeurs sont pondérées par la norme du gradient. Normalement, SIFT utilise une fenêtre gaussienne autour du point où est calculé le descripteur pour pondérer les valeurs en fonction de leur distance au point d'intérêt. Ici, cette fonction est remplacée par une pondération homogène de tous les gradients. Ce sont les histogrammes qui sont pondérés par une approximation de la gaussienne sur la fenêtre donnée. Cette approximation permet un gain significatif en temps de calcul pour une diminution négligeable du pouvoir discriminant. Par ailleurs, l'orientation et l'échelle de chaque descripteur sont fixées au départ et alignées sur le repère de l'image ce qui permet un gain de temps supplémentaire.

En plus du descripteur SIFT, plusieurs propriétés de la couleur sont extraites du voisinage du point d'intérêt :

- La couleur du pixel notée C_i
- La couleur moyennée sur le voisinage, C_m
- La variation locale de la couleur, exprimée comme :

$$\sigma = \frac{1}{N}\sqrt{\sum_{N}(C_i - C_m)^2} \tag{4.2}$$

où N est le nombre de pixel dans le voisinage

La couleur est normalisée pour rendre le descripteur plus robuste aux changements d'illumination qui peuvent affecter la scène.

4.2.3 Classification

4.2.3.1 Forêts d'arbres de décision

Les forêts d'arbres aléatoires sont des méthodes d'apprentissage utilisant un ensemble d'arbres de décision pour classer des données. Chaque arbre de décision fournit comme résultat la classe la plus probable correspondant au vecteur d'entrée. Le résultat final est obtenu en prenant le mode des résultats de tous les arbres, c'est à dire la classe qui a reçu le plus grand nombre de votes. La distribution de probabilité finale donnée par tous les arbres, peut être estimée approximativement en calculant :

$$P(L|d) = \frac{1}{T}\sum_{i=1}^{T} P_\delta(L|d) \tag{4.3}$$

où d est le descripteur fourni en entrée, L le label, T le nombre d'arbres utilisés et P_δ la distribution de probabilité fournit par chaque arbre qui est une distribution de Dirac

donnant 1 si c est la classe choisie par l'arbre, 0 sinon.

Tous les arbres sont entraînés avec les mêmes paramètres mais sur des jeux de données différents. Soit N l'ensemble des données disponibles pour l'apprentissage, représentant n descripteurs. Chaque sous-ensemble N_i est constitué suivant la procédure de bootstrap, c'est à dire que n vecteurs sont choisis de manière aléatoire dans N avec remplacement. Le même vecteur peut donc être tiré plusieurs fois et certains vecteurs ne sont pas utilisés et forment ce que l'on appelle les données *out of bag*. Le caractère aléatoire du choix des vecteurs dans chaque sous ensemble permet de réduire la variance des prédictions. Le hasard est aussi introduit à un deuxième niveau, lors de la construction des arbres. Pour chaque nœud, un sous-ensemble des variables (champs du descripteurs) choisi aléatoirement, est créé pour déterminer la meilleure séparation des données. Le nombre de variables utilisées pour chaque nœud est typiquement de l'ordre de la racine carrée du nombre total de variables. Les prédictions sont ensuite réalisées en sélectionnant itérativement la branche droite ou gauche à chaque nœud jusqu'à atteindre un nœud terminal.

L'occurrence plus importante de certaines classes peut biaiser les résultats en faveur de la classe la plus probable. En effet si une classe est présente 90% du temps, labelliser l'ensemble des pixels comme étant de cette classe permet d'atteindre de très bon résultats alors que ce n'est pas ce qui est souhaité. Pour palier ce problème, chaque données utilisée pour l'apprentissage est pondérée par un poids inversement proportionnel à la fréquence de la classe dans les données. Ceci permet de pénaliser plus fortement les erreurs sur les classes les moins fréquentes et donc d'obtenir des prédictions équilibrées pour l'ensemble des classes (voir section 4.4).

4.2.3.2 Champ Conditionnel Aléatoire

L'inconvénient majeur de le plupart des algorithmes de classification comme les arbres de décision, est qu'ils ne tiennent compte que des données locales, à savoir le descripteur courant, pour associer un label au pixel. Il en résulte une labelisation bruitée qui souffre des variations locales de luminosité, des éventuelles occlusions etc. Une manière efficace d'améliorer les prédictions consiste à prendre en compte le contexte autour de chaque pixel lors du processus de labelisation. Pour cela beaucoup de méthodes ont recours à des modèles graphiques probabilistes qui permettent de forcer la cohérence globale dans l'attribution des labels aux pixels. Parmi les différents modèles graphiques probabilistes, les champs conditionnels aléatoires (CRF) sont aujourd'hui les plus utilisés. Ils ont été introduit dans [Lafferty 2001] pour s'affranchir des limitations des modèles graphiques précédents, comme les HMM.

Pour définir un CRF, on construit $\boldsymbol{X}$ un champ aléatoire défini sur l'ensemble des variables $(\boldsymbol{X}_1, ...\boldsymbol{X}_N)$ correspondant aux données d'observations et $\boldsymbol{Y}$ un champ aléatoire défini sur l'ensemble des variables $(\boldsymbol{Y}_1, ...\boldsymbol{Y}_N)$ modélisant les classes correspondant aux observations. $\boldsymbol{Y}$ a valeur dans $\mathcal{L} = (l_1, ...l_k)$, l'ensemble des k classes possibles. Un CRF

est alors défini sur $\boldsymbol{X}$ et $\boldsymbol{Y}$ comme suit :

Definition Soit $\mathcal{G} = (V, E)$ un graphe avec V l'ensemble des noeuds et E l'ensemble des arêtes. $\mathcal{G}$ est tel que $\boldsymbol{Y} = (\boldsymbol{Y}_\nu)_{\nu \in V}$ et $\boldsymbol{Y}$ est indexé par les nœuds V du graphe. Alors $(\boldsymbol{X}, \boldsymbol{Y})$ est un champ conditionnel aléatoire si, lorsque conditionné sur $\boldsymbol{X}$, les variables aléatoires $\boldsymbol{Y}_v$ obéissent à la propriété de Markov par rapport au graphe : $P(\boldsymbol{Y}_\nu | \boldsymbol{X}, \boldsymbol{Y}_\omega, \omega \neq \nu) = P(\boldsymbol{Y}_\nu | \boldsymbol{X}, \boldsymbol{Y}_\omega, \omega \sim \nu)$ où $\sim$ signifie que ν et ω sont voisin dans $\mathcal{G}$.

Autrement dit, un CRF est un graphe pour lequel les labels associés aux noeuds dépendent d'une part des observations $\boldsymbol{X}$ et d'autre part du contexte, modélisé par le voisinage directe du noeud courant.

La probabilité conditionnelle d'avoir une labelisation $\boldsymbol{Y}$ compte tenu des observations $\boldsymbol{X}$ peut être écrite sous forme d'une distribution de Gibbs :

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{\boldsymbol{Z}(\boldsymbol{X})} exp(-\sum_{c \in \mathcal{C}_\mathcal{G}} \phi_c(\boldsymbol{Y}_c|\boldsymbol{X})) \tag{4.4}$$

où $C_\mathcal{G}$ est l'ensemble des cliques c, ϕ_c le potentiel associé et $\boldsymbol{Z}(\boldsymbol{X})$ un facteur de normalisation qui garanti une probabilité dans l'intervalle $[0, 1]$.

Pour différentes raisons, notamment pour éviter les erreurs de calcul numérique lorsque les probabilités sont faibles, il est d'usage de travailler avec le logarithme des probabilités plutôt que les probabilités elles-mêmes. En effet si les probabilités varient sur $[0, 1]$, leur logarithme, lui, varie sur $]-\infty, 0]$. On définit alors l'énergie associée à la labellisation $\boldsymbol{y} \in \mathcal{L}^N$ est :

$$\boldsymbol{E}(\boldsymbol{y}|\boldsymbol{X}) = \sum_{c \in \mathcal{C}_\mathcal{G}} \phi_c(\boldsymbol{y}_c|\boldsymbol{X}) \tag{4.5}$$

Le problème de la classification multi-classes avec un CRF est souvent formulé comme la recherche du maximum de probabilité a posteriori (MAP), donné par :

$$\boldsymbol{y}^* = \underset{\boldsymbol{y} \in \mathcal{L}^N}{\operatorname{argmax}} P(\boldsymbol{y}|\boldsymbol{X}) \tag{4.6}$$

La recherche du maximum de probabilité est donc équivalente à la recherche du minimum d'énergie.

Les premiers modèles de CRF, appelés *adjacency CRF*, utilisaient uniquement les pixels immédiatement adjacents comme voisinage d'un pixel. De ce fait ces modèles échouaient à représenter les relations à longue distance entre les pixels et produisaient des résultats très lissés.

Ici on utilise le modèle présenté dans [Krähenbühl 2011]. Il s'agit d'un CRF dont le graphe associé est complet, c'est à dire que chaque pixel est connecté à tous les autres.

Dans ce contexte, l'énergie de Gibbs s'écrit :

$$E(\boldsymbol{y}|\boldsymbol{X}) = \sum_{i}^{N} \phi_u(\boldsymbol{y}_i|\boldsymbol{X}) + \sum_{i,j\neq i} \phi_p(\boldsymbol{y}_i, \boldsymbol{y}_j|\boldsymbol{X}_i, \boldsymbol{X}_j) \tag{4.7}$$

où $\phi_u(\boldsymbol{y}_i|\boldsymbol{X})$ est le potentiel unitaire de chaque pixel, c'est à dire une mesure de la probabilité (log) de lui associer un label donné. Ce potentiel est calculé indépendamment pour chaque pixel à partir des résultats obtenu par les RF.

Le potentiel associé aux paires de pixels, qui lui tient compte du contexte, est donné par :

$$\phi_p(\boldsymbol{y}_i, \boldsymbol{y}_j|\boldsymbol{X}_i, \boldsymbol{X}_j) = \mu(\boldsymbol{y}_i, \boldsymbol{y}_j) \sum_{m} \omega_m k_m(\boldsymbol{f}_i^{(m)}, \boldsymbol{f}_j^{(m)}) \tag{4.8}$$

où $\mu(.,.)$ est une fonction de compatibilité des labels qui pénalise l'attribution de labels différents pour des pixels voisins et où :

$$k_m(\boldsymbol{f}_i^{(m)}, \boldsymbol{f}_j^{(m)}) = exp(-\frac{1}{2}[\boldsymbol{f}_i^{(m)} - \boldsymbol{f}_j^{(m)}]^T \Lambda_m [\boldsymbol{f}_i^{(m)} - \boldsymbol{f}_j^{(m)}]) \tag{4.9}$$

est un noyau gaussien et où $\boldsymbol{f}_i$ et $\boldsymbol{f}_i$ sont des vecteurs représentant les caractéristiques des pixels i et j et $\omega^{(m)}$ une pondération de l'importance de chacune des caractéristiques.

Inference L'inférence est basée sur l'approximation du champ moyen de $P(\boldsymbol{Y}|\boldsymbol{X})$ par une distribution $Q(\boldsymbol{Y})$ qui minimise la distance de Kullback-Leibler :

$$Q(\boldsymbol{Y}) = \underset{Q\in D}{\operatorname{argmin}} D_{KL}(Q||P) \tag{4.10}$$

où D_{KL} est la distance de Kullback-Leibler et D l'ensemble des distributions qui peuvent être exprimées comme le produit des probabilités marginales

$$Q(\boldsymbol{Y}) = \prod_{i} Q_i(\boldsymbol{Y}_i) \tag{4.11}$$

Avec la structure présentée précédement, l'étape d'échange de messages de la méthode du champ moyen consiste simplement en un filtrage par une gaussienne pour lequel une approximation efficace existe. Ceci permet de réduire la complexité du problème de quadratique à linéaire par rapport au nombre de nœud dans le graphe et sublinéaire par rapport au nombre d'arêtes. Pour plus de détails, le lecteur est renvoyé à l'article de référence [Krähenbühl 2011].

4.3 Consistance spatio-temporelle

Les RF et les CRF permettent d'obtenir de bons résultats de labelisation (voir les résultats à la section 4.4). Cependant classer des images acquises successivement de façon indépendante peut conduire à des résultats bruités. Pour rendre plus robuste la classification, il peut être intéressant d'utiliser un filtrage, en exploitant les prédictions de plusieurs images sphériques voisines. Le forçage de la consistance spatio-temporelle des résultats a été envisagée à plusieurs reprises dans la littérature. Une première approche consiste à intégrer dans le modèle du CRF, le voisinage temporel de chaque pixel [Wojek 2008, Xiao 2009]. Le problème est qu'avec un graphe complètement connecté, le nombre de connexions du graphe explose et avec lui la complexité de l'inférence. Une autre approche utilise le flux optique pour propager les labels d'une image à l'autre [Miksik 2013] mais nécessite un apprentissage préalable d'une fonction de similarité entre les pixels.

Dans le contexte de la représentation des sphères égocentrées, puisque les informations de profondeur sont disponibles, une autre stratégie a été adoptée, illustrée à la figure 4.4. Il s'agit de reprojeter sur la sphère à filtrer, les images labellisées des sphères voisines au moyen d'une fonction de warping.

On note $\boldsymbol{\mathcal{I}}^*$ l'image de référence de dimension $m \times n$ pour laquelle on veut filtrer les résultats de la prédiction. Un pixel dans l'image $\boldsymbol{\mathcal{I}}^*$ est identifiée par sa position $\boldsymbol{p}^* = (u, v)$, avec $u \in [0, m[$ et $v \in [0, n[$. Un point 3D de l'espace Euclidien est défini par $\boldsymbol{P} = \{\boldsymbol{p}^*, Z, l\}$ où $Z \in \mathbb{R}^+$ est la profondeur exprimée dans le repère image et $l \in \mathcal{L}_S$ le label associé. Soit $\boldsymbol{\mathcal{I}}_i$ une image du voisinage $\mathcal{N}$ de $\boldsymbol{\mathcal{I}}^*$ ayant une position $\boldsymbol{T}(x)_i$ exprimée dans le repère de l'image $\boldsymbol{\mathcal{I}}^*$. On peut synthétiser une nouvelle image à partir des labels $L(p)$ de l'image $\boldsymbol{\mathcal{I}}_i$ à la position de l'image de référence par une fonction de warping :

$$p^{\omega} = \omega(\boldsymbol{T}(x)_i; Z, l, \boldsymbol{p}^*) \tag{4.12}$$

La fonction $\omega(.)$ transfère le pixel $\boldsymbol{p}^*$ de l'image de référence dans l'image $\boldsymbol{\mathcal{I}}_i$ par la transformation rigide $\boldsymbol{T}(x)_i$ suivie d'une projection sphérique. Les positions des points p^{ω} ne correspondant pas exactement aux pixels des images $\boldsymbol{\mathcal{I}}_i$, l'interpolation au plus proche voisin est utilisée pour sélectionner le label adéquat. On note $\boldsymbol{\mathcal{I}}_i^{\omega}$ l'image ainsi créée.

L'accumulation des N images de $\mathcal{N}$ permet de définir une distribution de probabilité des classes en chaque pixel :

$$P(L(\boldsymbol{p}^*)) = \frac{1}{N} \sum_{i \in \mathcal{N}} \omega_i P(L(\boldsymbol{p}_i^{\omega})) \tag{4.13}$$

où $L(\boldsymbol{p})$ est le label associé au point $\boldsymbol{p}$ et ω_i une pondération associée à chaque image, inversement proportionnelle à la distance de celle-ci à l'image de référence.

Outre le fait de filtrer les labels et donc de rendre les résultats plus robustes, cette approche permet aussi d'accéder à une mesure de la confiance en un label donné. Dans le

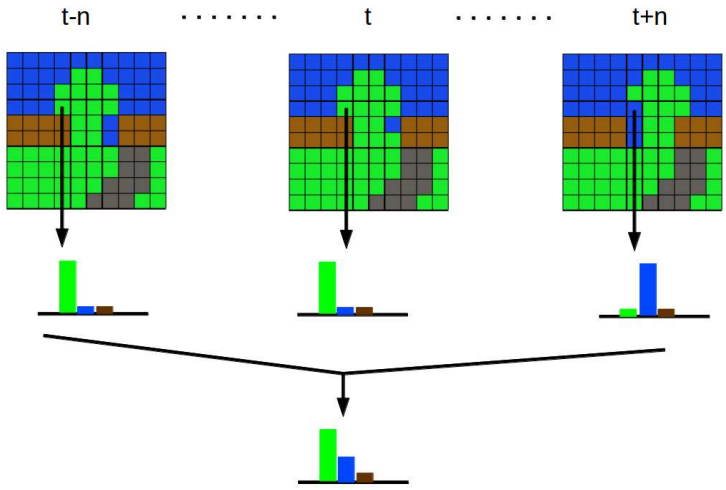


FIGURE 4.4 – *Illustration du principe de filtrage : en utilisant une fonction de warping, les labels associés à un même pixel depuis plusieurs points de vu sont accumulés pour produire une nouvelle distribution de probabilité.*

contexte du déploiement de robots dans un environnement réaliste ceci est d'autant plus intéressant que l'inférence MAP ne fournit que le label le plus probable sans estimation de la confiance. Par ailleurs, ceci ajoute un degré de profondeur à l'ensemble du processus de classification et améliore ainsi la capacité de généralisation du processus, comme expliqué précédemment.

4.4 Résultats de la labelisation

La qualité du processus de classification a été évaluée en deux temps. Dans un premier temps, la labelisation individuelle des images a été testée sur plusieurs bases de données, certaines acquises à l'INRIA et composées d'images sphériques, d'autres disponibles publiquement et composées d'images panoramiques ou de vidéos. Nos résultats sont comparés à des travaux de références sur chacune des bases de données, exceptée celle acquise à l'INRIA qui n'est pour l'heure, pas rendue publique. Dans un second temps, la consistance spatio-temporelle a été évaluée uniquement sur la base de donnée de l'INRIA du fait de la nécessité de tenir compte du formalisme de la représentation sphérique.

4.4.1 Bases de données

INRIA La base de données acquise sur le campus de l'INRIA se compose de plus de 13000 images sphériques de dimension 2048 x 665 pixels. Elle a été réalisée le long d'un

parcours de 1.6km dans un environnement semi-urbain, c'est à dire composé de bâtiments et de parking, espacés de zones boisées. Les 9 classes suivantes sont labelisées dans cette base de données : ciel, arbre, route, panneau de signalisation, trottoir, bâtiment, voiture, signe au sol (route), autres. Un sous ensemble d'images choisies aléatoirement a été labellisé à la main pour servir de base d'apprentissage.

KITTI La base de données KITTI a été créée avec un véhicule doté d'un GPS, d'un scanner laser Velodyne HDL-64E, de deux caméras noir et blanc, deux caméras couleurs. Les images sont acquises à 10Hz et de dimension 1382 x 512 pixels. La base de données n'est pas labelisée et ne fournit pas, pour l'heure, de benchmark pour la labelisation d'image. Mais plusieurs personnes ont annotés des images de KITTI, les résultats sont regroupés ici [2]. La base de données utilisée est celle produite par [German Ros 2015] avec pour modification la fusion des classes "voiture" et "velo" du fait du faible nombre d'exemple pour cette dernière et la modification de certains labels qui semblaient mal attribués. La classe piéton n'a pas non plus été prise en compte, comme dans l'article de référence, du fait du faible nombre d'exemples disponibles. Les classes utilisées sont donc bâtiment, arbre, ciel route, véhicule, trottoir, poteau, panneau et barrière.

CamVid est une base de données composée de plusieurs vidéos au format 960x720 acquise dans un environnement urbain hautement dynamique [3]. Les images sont labelisées à 1hz avec 32 classes différentes qui sont données à la figure 4.6. Deux séquences sont utilisées pour les tests : 01TP qui dure 2 :04minutes et 06R0 qui dure 1 :41minutes. La première moitié de chaque séquence est utilisée pour l'apprentissage et la seconde pour les tests. Les résultats sur cette base de données sont comparés avec ceux de [Miksik 2013].

4.4.2 Mesures

Plusieurs mesures standards sont rapportées pour évaluer la qualité de la labelisation multi-classes. Étant donné l'objectif de labelliser correctement le plus grand nombre de pixels possible, la première mesure rapportée est l'exactitude globale, c'est à dire le nombre total de pixels correctement labellisés, noté g :

$$g = \frac{1}{M} \sum_{i=0}^{M} \delta_{l_i}^{l_{true}} \tag{4.14}$$

où M est le nombre total de pixels, δ est le symbole de Kronecker, l_i le label attribué au pixel et l_{true} le label réel. Cependant sa valeur peut être biaisée par la proportion de l'image occupée par chaque classe. Par exemple, une scène où prédominent les classes "arbres" et "route" obtiendra un très bon score en labellisant toute l'image uniquement avec ces deux

2. www.cvlibs.net/datasets/kitti/eval_semantics.php
3. http ://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/#ClassLabels

labels et en ignorant les autres classes, ce qui n'est pas souhaitable. L'exactitude moyenne par classe, notée m, est donc aussi rapportée de manière à s'affranchir du biais introduit par la taille relative des instances. Deux autres mesures sont aussi données pour chaque classe individuellement. La première est la précision, notée p_r.

$$p_r = \frac{1}{N} \sum_{k=1}^{N} \frac{t_k}{n_k} \tag{4.15}$$

où N est le nombre de classes, t_k le nombre de vrais positifs pour la classe k et n_k le nombre de pixels labellisés comme appartenant à cette classe, c'est à dire le nombre de vrais positifs et de faux positifs. La seconde mesure proposée est le rappel, noté r_p.

$$r_p = \frac{1}{N} \sum_{k=1}^{N} \frac{t_k}{m_k} \tag{4.16}$$

où m_k est le nombre de pixel appartenant à la classe k, c'est à dire l'ensemble des vrais positifs et des faux négatifs.

4.4.3 Mise en oeuvre

L'ensemble des expériences a été mené sur un ordinateur doté d'un processeur Intel® Core™ i7-3840QM cadencé à 2.80GHz. Les programmes sont mono-threadés. Plusieurs bibliothèques en C++ ont été utilisées pour la conduite des expériences. Les descripteurs denses sont calculés grâce à l'implémentation dense de SIFT disponible dans *VLFEAT*[4]. Pour l'inférence dans le CRF, la bibliothèque utilisée est celle fournie par les auteurs de l'article [Krähenbühl 2011] disponible ici.[5] L'apprentissage des paramètres du CRF a été réalisé par validation croisée.

4.4.4 Labelisation unitaire

KITTI Les résultats de la labelisation pour la base de données KITTI sont donnés à la table 4.1 et comparés avec ceux de l'article [German Ros 2015] à la table 4.2. Des exemples de résultats sont donnés à la figure 4.5. Le temps nécessaire à la labelisation d'une image pleine résolution (1241x376) est de 9.2 secondes se décomposant en : 1.2 seconde pour extraire les descripteurs, 4.6 secondes pour les RF et 3.8 secondes pour l'inférence avec le CRF. La majorité des classes présentent des résultats satisfaisants. Les classes "panneau" et "barrière" présentent les résultats les plus faibles pour la précision et le rappel. Ceci s'explique pour la classe "panneau" par la faible taille des instances et le petit nombre d'exemples de la classe. De plus la grande variance intra-classe comparée au faible nombre d'exemples disponibles dans la base de données rend plus délicate la détection

4. http ://www.vlfeat.org/
5. http ://graphics.stanford.edu/projects/densecrf/

de cette classe. Pour la classe "barrière", les instances se superposent souvent à des bâtiments ou de la végétation, d'où le fait que la classe soit mal estimée. Par ailleurs, cette classe présente un faible nombre d'exemple dans la base d'apprentissage ce qui renforce encore le phénomène. La classe "poteau" présente une précision correcte, ce qui signifie que peu de pixels labellisés comme tels sont en fait une autre classe. Cependant le r_p est faible. Ceci provient probablement du fait de la taille réduite des instances de cette classes qui sont difficile à détecter. En effet les descripteurs étant calculés sur un voisinage, ils incluent forcément, pour de petits objets, des pixels qui n'appartiennent pas à la classe correspondante, d'où l'ajout de "bruit" qui rend difficile leur classification correcte.

La comparaison des résultats de notre algorithme avec ceux de [German Ros 2015] montre des performances significativement meilleures dans notre cas. Ceci provient probablement de l'utilisation d'un modèle 3D dans [German Ros 2015] qui introduit un biais de reprojection et qui diminue significativement la qualité des résultats. Par ailleurs, les labels de certaines images utilisées pour l'apprentissage ou les tests ont été corrigés suite à la détection de ce qui est probablement des erreurs dans l'étape de labelisation manuelle. Ceci peut sensiblement améliorer les résultats, notamment pour les classes peu représentées.

TABLE 4.1 – *Résultats labellisation sur la base de données KITTI (%)*

Classe	p_r	r_p
bâtiment	87	87
arbres	82	86
ciel	97	93
route	72	78
voiture	87	86
trottoirs	69	73
poteaux	73	32
panneau	44	32
barrière	51	34

TABLE 4.2 – *Comparaison avec les résultats de [German Ros 2015] sur la base de données KITTI (%)*

exactitude	[German Ros 2015] (TSS)	Notre Algorithme
globale	61.6	82.1
moyenne	51.2	72.2

CamVid La comparaison des résultats de [Miksik 2013] avec notre algorithme sur la base de données CamVid sont donnés à la table 4.3 et des exemples de résultats sont donnés à la figure 4.6. Le temps moyen nécessaire à la labelisation d'une image pleine résolution

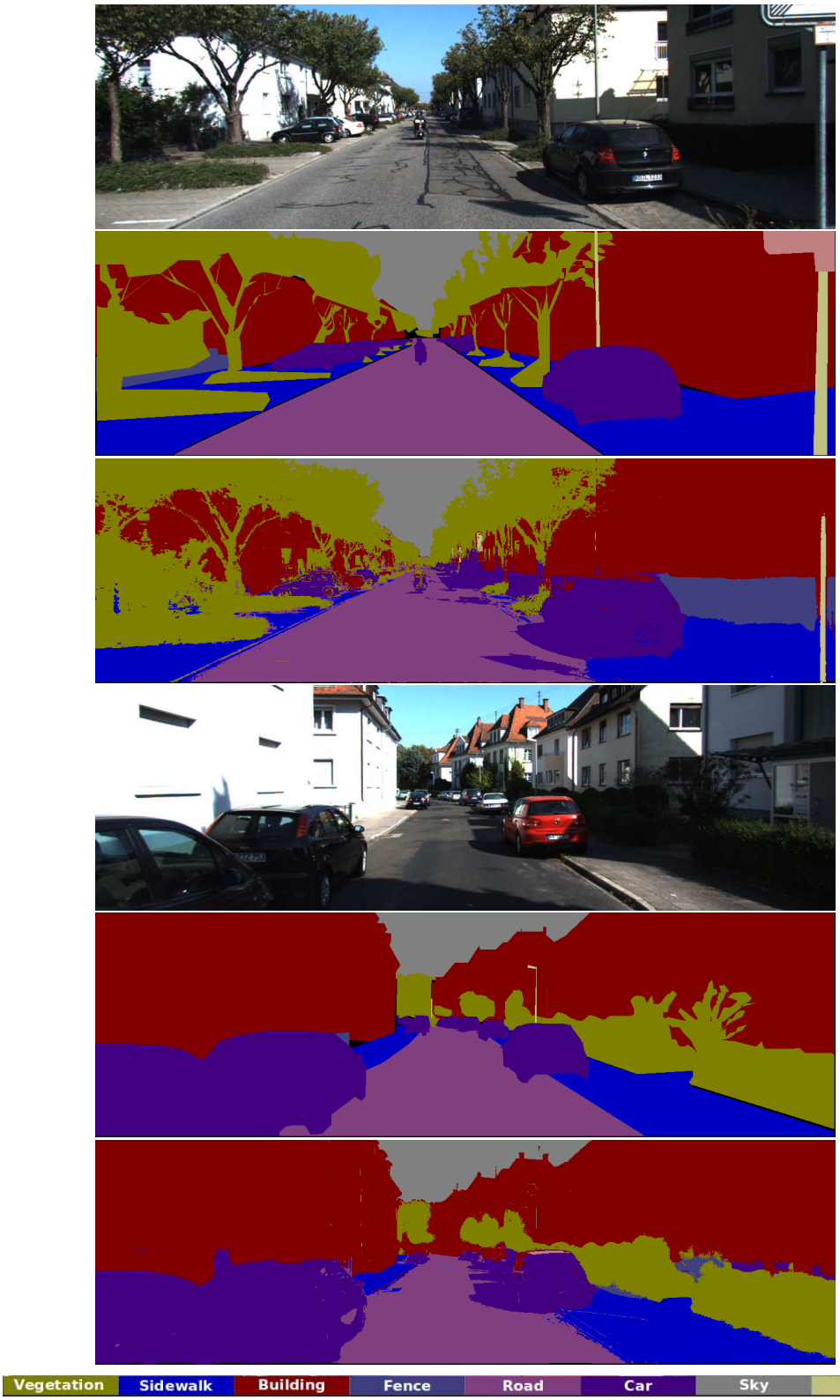


FIGURE 4.5 – *Exemple de résultats pour la base de données KITTI. En haut image couleur, au milieu vérité terrain, en bas image labellisée.*

est de 11.2 secondes se décomposant en : 2.0 secondes pour extraire les descripteurs, 4.9 secondes pour les RF et 4.3 secondes pour l'inférence avec le CRF. L'exactitude par pixel est inférieur que celle de [Miksik 2013] mais l'exactitude moyenne est meilleure dans notre cas. Ceci signifie que notre algorithme a tendance à mieux détecter chaque objet, au prix d'une moins bonne estimation de sa forme générale. Bien que l'exactitude globale la plus haute soit souhaitable, notre objectif étant de naviguer, on peut raisonnablement considérer que l'exactitude par classe est plus importante dès lors que ce qui caractérise un lieu est davantage la présence de certaine classe que leur forme exacte.

TABLE 4.3 – *Comparaison avec les résultats de [Miksik 2013] sur la base de données CamVid (%)*

exactitude	[Miksik 2013]	Notre Algorithme
globale	84.2	80.1
moyenne	59.5	73.2

INRIA Les résultats de la labelisation pour la base de données INRIA sont donnés à la table 4.4 et des exemples de résultats sont donnés à la figure 4.7. Sur cette base de données, l'exactitude globale est de 82.0% et l'exactitude par classe de 79.6%. Le temps nécessaire à la labelisation d'une image à la résolution 1024*333 est de 6.8 secondes se décomposant en : 0.9 seconde pour extraire les descripteurs, 2.9 secondes pour les RF et 3.1 secondes pour l'inférence avec le CRF. Les classes "ciel", "arbre" et "route" présentent les meilleurs résultats à la fois en terme de précision et de rappel. La classe bâtiment présente des résultats relativement faibles qui s'expliquent par la grande variabilité intra-classe relativement au nombre d'exemples dans la base d'apprentissage. Les couleurs et textures des bâtiments sur le site de l'INRIA changent considérablement et l'algorithme de classification généralise difficilement. Par ailleurs, les bâtiments, tout comme les voitures, présentent de larges surfaces réfléchissantes qui rendent difficiles l'attribution de labels correctes.

TABLE 4.4 – *Résultats labellisation sur la base de données INRIA*

Classe	p_r	r_p
Ciel	90	99
Arbre	80	87
Bâtiment	61	72
Signalisation	40	35
Voiture	71	70
Route	83	91
Marquage au sol	75	49
Troittoir	73	83

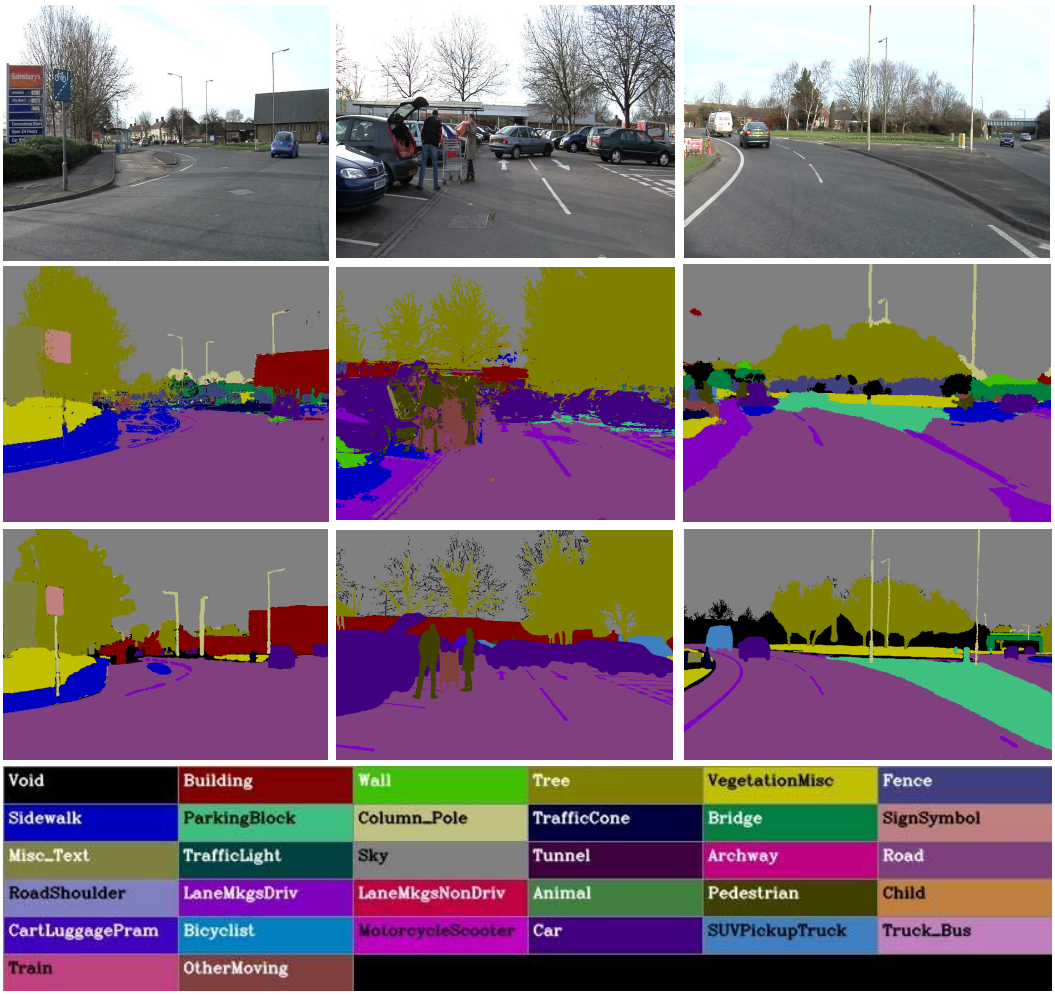


FIGURE 4.6 – *Exemple de résultats pour la base de données CamVid.*

4.4.5 Consistance temporelle

Pour évaluer l'amélioration de la labelisation permise par le forçage de la consistance temporelle, plusieurs mesures ont été réalisées avec une taille de voisinage croissante. Les résultats sont donnés à la table 4.5. Le voisinage s'entend, pour une séquence d'image sphérique, comme la largeur d'une fenêtre centrée sur l'image courante. On observe que pour un voisinage croissant la qualité de la labelisation croit avant de diminuer. L'amélioration de la classification sur un voisinage restreint s'explique par le fait que la scène change peu et que l'accumulation des observations permet d'être plus robuste au bruit et de mieux différencier des classes semblables. La diminution de la qualité au delà d'une certaine distance s'explique par le fait que, pour des images sphériques éloignées, la projection des labels sur l'image centrale ajoute des erreurs du fait des occlusions qui ne sont pas prises en compte dans le modèle et des erreurs dans l'estimation de la profondeur qui conduisent

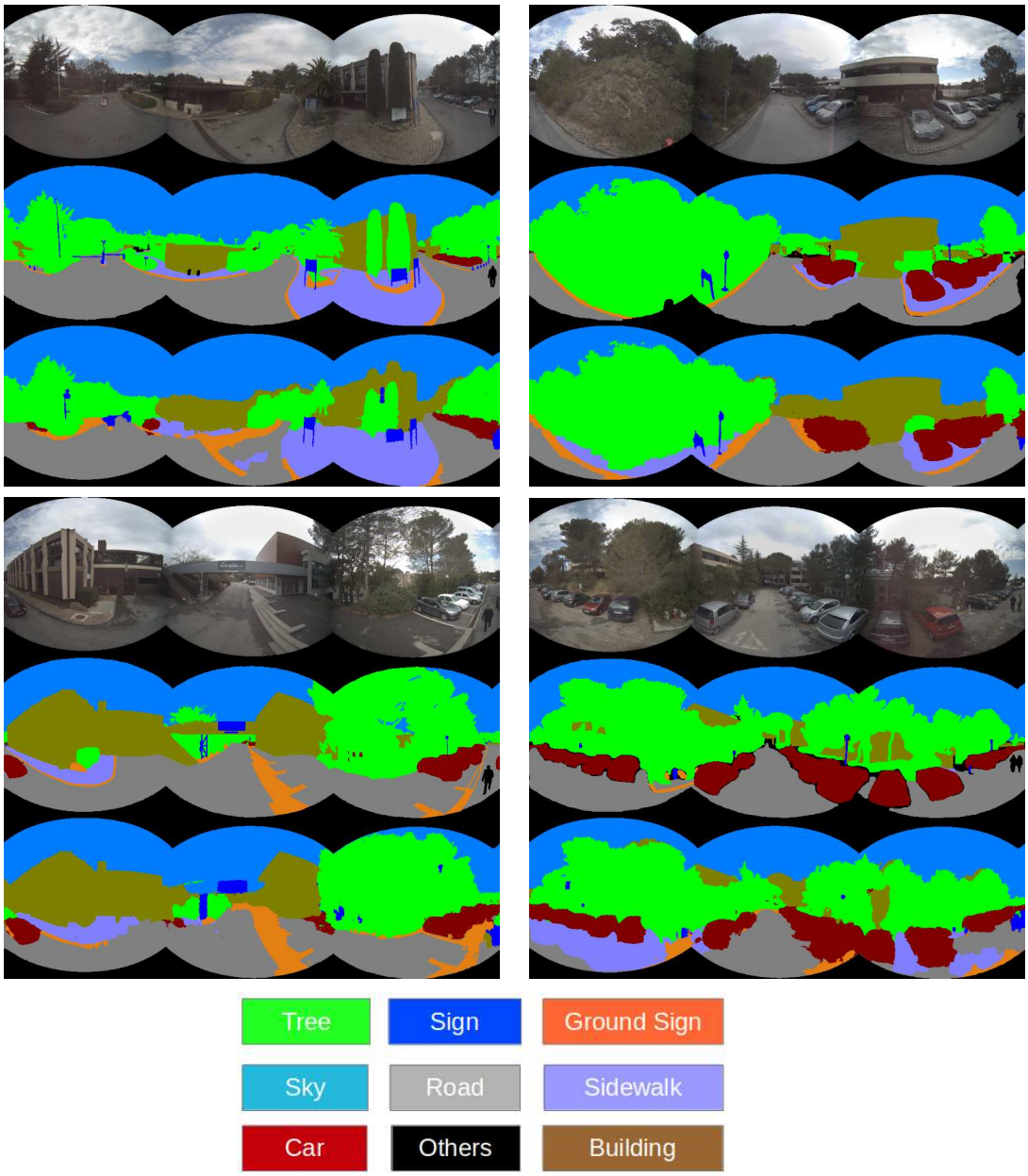


FIGURE 4.7 – *Exemple de résultats pour la base de données INRIA. En haut image couleur, au milieu vérité terrain, en bas image labellisée.*

a projeter les labels au mauvais endroit.

L'incrément de qualité de 3% en moyenne est relativement faible et tient au fait que seule la partie la plus proche de l'image, pour laquelle l'estimation de profondeur est disponible et fiable, est utilisée pour le warping. Cette zone est déjà la mieux estimée donc l'amélioration est limitée. Cependant le filtrage des labels joue un rôle important dans cer-

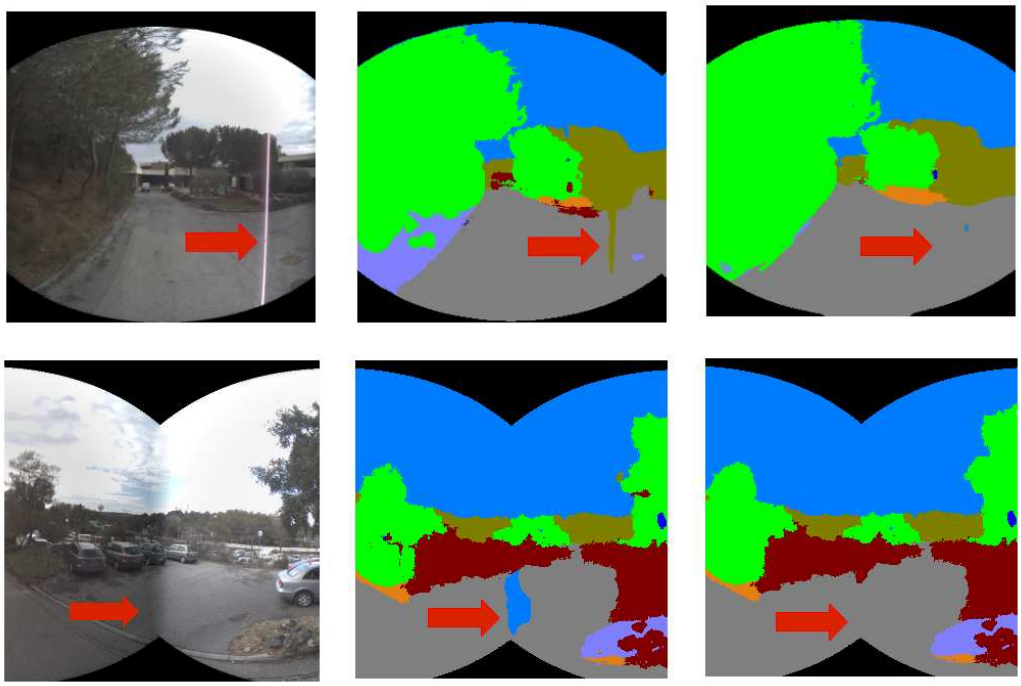


FIGURE 4.8 – *Efficacité du filtrage lorsque la luminosité de l'image est modifiée. A gauche : image RGB ; au centre : image labelisée sans filtrage ; A droite : image filtrée. L'exemple du haut montre un cas où la camera est éblouie par le soleil. L'image du bas montre un cas où le stitching a fortement modifié l'intensité lumineuse dans l'image, conduisant à une confusion des labels*

TABLE 4.5 – *Résultats de la labellisation en prenant en compte la consistance temporelle sur la base de données INRIA*

Voisinage	$g(\%)$
$N = 1$	81.1
$N = 3$	82.7
$N = 7$	84.2
$N = 11$	82.3

taines zones. En effet les images sphériques de la base de données INRIA souffrent de deux défauts : premièrement l'opération de stitching utilisée pour assembler les images sphériques des différentes caméras formant une vue sphérique, créer un assombrissement locale de l'image au niveau de la jonction. Ceci a tendance à induire l'algorithme de classification en erreur et à produire du bruit de labelisation à ce niveau. Le deuxième défaut est dû aux conditions d'ensoleillement pendant l'acquisition. Dans certaines images les caméras sont éblouies par le soleil ce qui génère du blooming sous forme d'un trait vertical dans l'image. Ces deux défauts apparaissent toujours au même endroit dans l'image alors

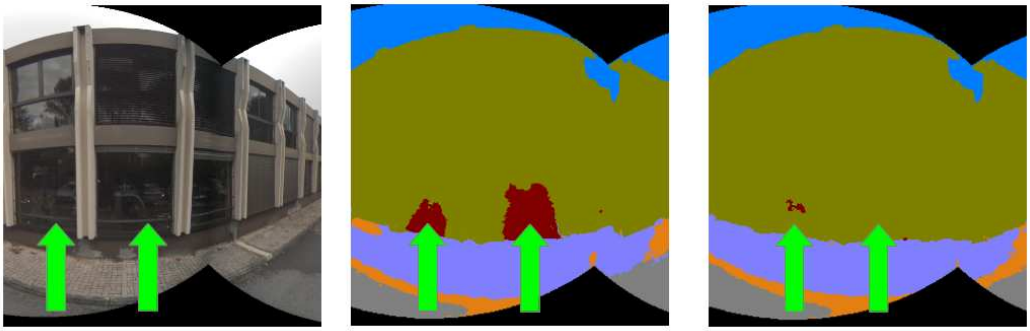


FIGURE 4.9 – *Exemple de traitement des reflets. A gauche : image RGB, on distingue le reflet des voitures dans les vitres ; au centre : image labellisée sans filtrage, les reflets des voitures sont labellisés avec la classe voiture ; A droite : image filtrée, les reflets sont très atténués et la zone est labellisée avec la classe "bâtiment" comme il se doit.*

que le robot se déplace et que la scène évolue dans l'image. Le forçage de la consistance temporelle permet donc d'atténuer fortement ce phénomène comme illustré à la figure 4.8.

Un autre problème difficile à traiter et partiellement résolu par le forçage de la consistance temporelle est celui des surfaces spéculaires. En réfléchissant l'environnement alentour, une zone spéculaire induit très facilement les algorithmes de classification en erreur puisque l'aspect local ne correspond pas à l'objet qui supporte le reflet mais à celui reflété. Ce phénomène est illustré à la figure 4.9. En prenant en compte les labels attribués à la même zones sur plusieurs images successives, l'algorithme peut corriger au moins partiellement le phénomène. En effet le reflet se déplace à mesure que le robot évolue et l'erreur est de ce fait atténuée par la prise en compte du voisinage temporel.

4.5 Limites des mesures de performances des algorithmes de classification

Dans un but de normalisation des résultats et pour offrir une comparaison avec d'autres algorithmes, les mesures rapportées ici correspondent aux mesures standards utilisées dans le domaine de la labelisation d'images. Cependant il est intéressant de noter les limites de ces métriques et donc les réserves qu'il convient d'adopter quant à l'analyse des résultats, ceci à travers trois exemples concrets issus des deux premières bases de données utilisées pour les tests. Le premier exemple est illustré à la figure 4.10 où sont représentés une image RGB de la base KITTI et la vérité terrain associée, visible par transparence. On remarque que les labels sont associés de manière grossière aux pixels, notamment pour les arbres dont une part significative n'est pas labellisée correctement dans la vérité terrain. Si des pixels associés à un arbre par l'algorithme de classification ne sont pas correctement labellisés dans la vérité terrain, ils seront comptabilisés comme des erreurs, à tord. Plus grave encore,

ce type d'image, utilisé pour l'apprentissage, introduit des erreurs dans le modèle appris par l'algorithme de classification. Le premier problème de ces mesures est donc leur sensibilité à la précision de la labellisation. Le second problème est illustré à la figure 4.9 par les reflets des voitures dans les vitres. L'objet situé à cette position est bien un bâtiment, comme indiqué par la vérité terrain, mais l'image observée est celle d'un véhicule. Dès lors faut-il considérer que l'algorithme de classification se trompe lorsqu'il attribue à ces pixels le label "voiture" ? Là encore comptabiliser ce résultat comme une erreur n'est probablement pas opportun. Enfin le dernier problème que soulève ces mesures est illustré à la figure 4.11. L'ensemble de la végétation a été associé à une seul label "arbre" (en vert) lors de la phase d'apprentissage. Or dans cet exemple, l'algorithme de classification associe une partie des tronc des arbres à la classe "poteau" (en bleu). En s'approchant plus près des arbres que lors de la phase d'apprentissage, l'algorithme a estimé qu'il y avait plus de ressemblance entre un tronc d'arbre et un poteau qu'entre un tronc et du feuillage. Est-ce une erreur ? Ce problème souligne une autre limite qui tient au nombre de classes définies pour l'apprentissage qui n'est pas forcément adapté à toutes les situations.

Plus que la limite de ces mesures, ces exemples soulignent plutôt les limites de la supervision humaine qui, si elle est performante sur des exemples simples, est plus limitée dans des cas réels où la diversité des situations rendent les règles d'apprentissages fournies par l'Homme valable uniquement statistiquement. La mesure de performances à l'échelle des pixels est donc à considérer comme une estimation des performances plutôt qu'une mesure précise de celles-ci et les variations d'un algorithme à l'autre à considérer avec une distance d'autant plus raisonnable qu'elles sont faibles. Il serait probablement plus judicieux de mesurer le nombre d'objets détectés par rapport au nombre d'objets effectivement présents dans l'image mais là encore des problèmes se posent pour la définition de la vérité terrain. Par exemple comment dénombrer des arbres aux feuillages entrelacés ? Cette définition ne résoudrait probablement pas non plus le problème des reflets. La définition d'une mesure idéale reste donc un problème ouvert.

4.6 Conclusion

Dans ce chapitre une méthode de labellisation des images sphériques a été introduite. Elle exploite l'aspect local des images à travers l'utilisation de descripteurs robustes et le contexte spatial en ayant recours à un modèle graphique probabiliste. Le filtrage temporel des prédictions permet d'obtenir des résultats stables et dont la sensibilité aux changements d'apparence de la scène est réduite. Les expériences menées sur trois bases de données composées d'images de haute résolution acquises dans des environnements réels ont permis de démontrer l'efficacité et la robustesse de l'approche proposée, générant des images labellisées avec un haut niveau de confiance. Cependant l'information sémantique sous cette forme n'est pas bien adaptée à un usage pour la navigation. En effet la quantité de données que représente une image annotée est importante et les objets ne sont pas di-

FIGURE 4.10 – *Illustration du problème posé par la mesure de performances au niveau des pixels. La précision de la vérité terrain impacte les résultats.*

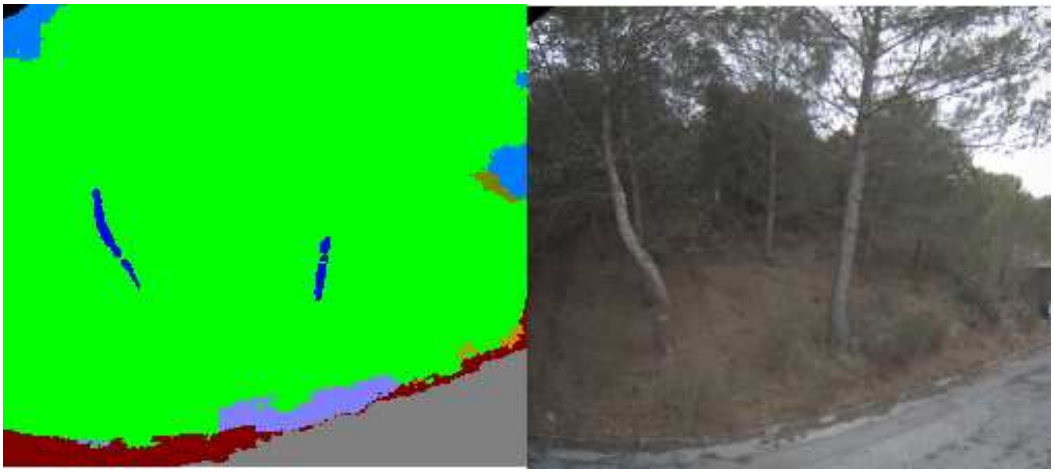


FIGURE 4.11 – *Exemple de cas où la vérité terrain n'est pas forcément adaptée à la situation du fait d'un nombre de classes définies insuffisant.*

rectement identifiés en tant que tels. Il serait préférable de disposer d'une description qui permette de modéliser précisément l'information sémantique contenue dans la scène tout en étant plus compacte. La construction de cette représentation est précisément le sujet du prochain chapitre.

CHAPITRE 5

Construction de la couche sémantique

5.1 Introduction

Le chapitre précédent a permis de montrer comment extraire les labels d'une image en prenant en compte le contexte spatial et temporel. L'étape de classification fournit des images annotées de manière dense et stable, ce qui permet d'envisager l'emploi de la sémantique à bas niveau, par exemple pour guider la reconstruction 3D. Le modèle local de l'environnement est une sphère RGBDL, c'est à dire une image sphérique augmentée des données de profondeur et labelisée. Cependant la compréhension de la scène utile à la navigation, requiert une analyse globale de l'image et la labellisation dense de celle-ci n'est pour cela pas suffisante. Comme évoqué à la section 3.5, la représentation des sphères RGBD souffre de certains défauts que l'utilisation de la sémantique doit permettre de corriger. Notamment, la représentation n'offre aucun moyen de référencer les sphères dans le graphe global, ce qui pose problème pour la localisation. La mise à jour des données, fondamentale dans un environnement dynamique, n'a pas non plus été envisagée. Il est donc nécessaire de construire une représentation sémantique qui permette de discriminer les images sphériques et de mettre à jour leur contenu. C'est précisément ce qui est proposé dans ce chapitre à travers une nouvelle description sémantique des images.

5.2 Graphe Sémantique

Pour atteindre les objectifs présentées en introduction, la représentation sémantique doit avoir trois propriétés essentielles :

- **Discriminante** : la représentation doit permettre de modéliser un très grand nombre d'images de manière unique. Pour cela elle doit tenir compte à la fois de la structure de l'image et de son contenu.
- **Compact** : le modèle doit être compact pour permettre une recherche rapide d'une sphère de référence donnée dans une vaste base de données.
- **Actualisable** : la représentation doit pouvoir être mise à jour simplement pour s'adapter aux changements de l'environnement.

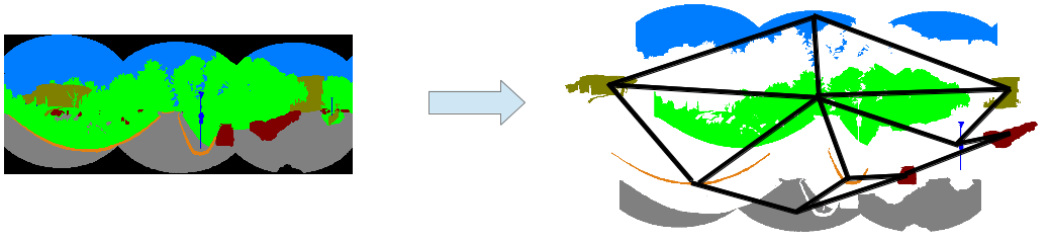


FIGURE 5.1 – *Vue schématique d'un graphe sémantique. Une fois l'image annotée, chaque zone sémantique, formée de pixels contiguës de même label, est extraite et forme un nœud du graphe. Les arêtes ne relient que les zones connexes.*

Pour répondre à ces exigences, le concept de *Graphe Sémantique* sur une image sphérique, illustré à la figure 5.1, a été introduit.

Soit $\mathcal{I}_{\mathcal{L}}$ une image labellisée, c'est à dire dont les pixels prennent leurs valeurs dans $\mathcal{L}$. Une *zone sémantique* dans l'image $\mathcal{I}_{\mathcal{L}}$, notée A_i, est un ensemble de pixels contiguës $\boldsymbol{p_i}$ ayant la même valeur. On définit un *graphe sémantique* pour l'image $\mathcal{I}_{\mathcal{L}}$, noté $\mathcal{G}_s$, par $\mathcal{G}_s = (A, E)$ où A est l'ensemble des *zones sémantiques* de l'image et E les arêtes du graphe. A chaque élément $A_i \in A$ est associé une enveloppe elliptique f_i et un jeu de paramètres :

- L_i, le label
- $q_i = (u, v)$, la position du barycentre de l'ellipse englobante de A_i dans l'image
- $s_i = (h, w)$, la taille de A où h et w sont les demie-axes de l'ellipse englobante.
- α_i l'orientation cette l'ellipse.

Les arêtes du graphe encodent les relations entre les zones sémantiques *connexes*, qui sont modélisées par le nombre de pixels à la frontière. Les zones non-connexes ne sont pas connectées dans le graphe.

Il convient de noter que cette représentation est subjective puisqu'elle dépend du point de vue. Ainsi une zone A_i peut recouvrir plusieurs objets de même nature qui apparaissent distincts depuis un autre point de vue. Ceci est conforme à ce que l'on souhaite puisque dans le cadre de la représentation des sphères egocentrées, c'est bien un point de vue particulier qui est mémorisé sous la forme d'une image sphérique et que l'on souhaite caractériser.

5.2.1 Caractéristiques des Graphes Sémantiques

Compacité : La représentation de l'image sous forme de *graphe sémantique* permet une forte compression de la carte. Pour une image contenant 20 zones sémantiques, chacune possédant 6 propriétés codées avec un entier de 32 bits, l'image sémantique peut être entièrement décrite par 3840 bits. L'image couleur originelle contient approximativement 24.10^6 bits, la représentation sous forme de graphe sémantique représente donc une com-

pression d'un facteur 6250. Cette représentation est donc particulièrement bien adaptée aux très grands environnements. Par comparaison, les méthodes basées sur des sacs de mots requièrent 4096bits pour coder **un seul** descripteur SIFT avec des flottants 32 bits.

Pouvoir discriminant de $\mathcal{G}_s$: Pour référencer les images sphériques dans la base de données, la représentation doit être suffisamment riche pour permettre de différencier toutes les images les unes des autres. En notant $\overline{\mathcal{L}}$ le cardinal de $\mathcal{L}$, l'ensemble des classes annotées, C_q le nombre de positions possibles dans l'image, C_s les tailles possibles des objets et n le nombre d'objets, le nombre d'images modélisables N_R est de l'ordre de :

$$N_R = (\overline{\mathcal{L}} \times C_q \times C_s)^n \tag{5.1}$$

En posant $\overline{\mathcal{L}} = 9$, $C_q = 8$ positions correspondant aux quatre quadrants de l'image et pour chacun la distinction haut/bas, $C_s = 4$ correspondant à la distinction qualitative grand/petit pour chaque dimension et en fixant $n = 10$, le nombre de possibilité de codage est supérieur à 10^{24}. Bien que seul une fraction de ces possibilités soit réaliste, le pouvoir représentatif des graphes sémantiques est très élevé.

Invariance par rotation Le graphe extrait d'une image dépend de la position d'où est observée la scène mais est invariant par rotation autour de l'axe vertical. En effet la position relative des objets ne change pas entre deux images prises suivant des orientations différentes. Pour cette raison, la représentation sous forme de graphe sémantique s'adapte particulièrement bien aux images sphériques.

Robustesse En plus de son extrême compacité, la représentation sous forme de graphe sémantique présente aussi l'avantage de se baser sur l'intégralité de l'image et de modéliser sa structure, elle est donc moins sensible aux problèmes d'occlusion que les méthodes basées sac de mots.

5.3 Architecture de la carte

L'architecture de la carte globale est illustrée à la figure 5.2. Elle est constituée de sphères de références RGBD-LG, c'est à dire de sphères photométriques (RGB) augmentées des données de profondeur (D) et annotées (L). A chaque sphère de référence est associé un graphe sémantique (G) décrivant sa structure et son contenu. Toutes les sphères sont connectées au niveau de la couche métrique par l'estimation de leurs poses relatives dans le graphe global $\mathcal{G}$. Pour structurer la recherche d'information dans la carte et permettre l'accès rapide à une carte locale, les sphères sont référencées dans un arbre T, en fonction de leur contenu sémantique. L'arbre T est composé de deux couches dont les nœuds sont successivement caractérisés par

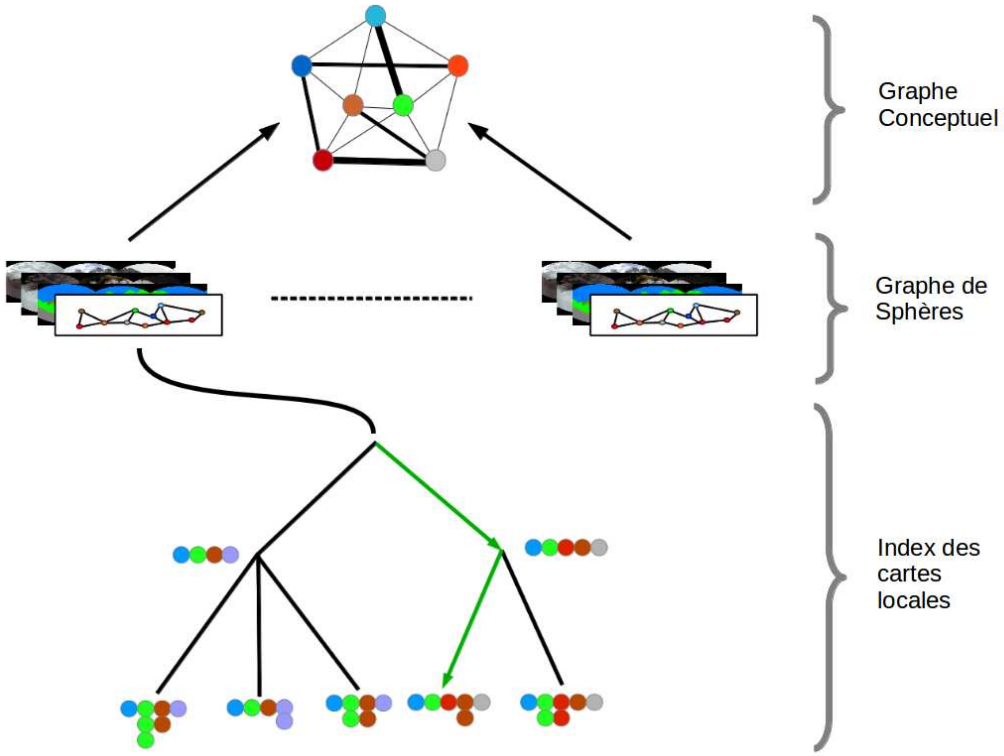


FIGURE 5.2 – *Vue schématique de l'architecture complète d'une carte. Chaque sphère de référence est composée de 2 couches principales : la couche métrique composée des images photométriques et de profondeur ; la couche sémantique composée de l'image annotée et du graphe sémantique. L'ensemble des graphes sémantiques est alors utilisé pour construire le graphe conceptuel qui traduit les relations statistiques de connexité entre les classes dans l'ensemble de la carte. Ils sont aussi utilisés pour référencer les cartes locales dans un arbre à deux niveaux. Le premier niveau sépare les cartes en fonction des classes présentes dans l'image, le second niveau en fonction du nombre d'instances de chacune des classes.*

- Le groupe des classes C_i apparaissant dans le graphe sémantique
- Le nombre d'instance de chacune des classes C_i dans ce graphe

Les graphes sémantiques sont donc regroupés successivement suivant les classes qu'ils contiennent puis le nombre d'instances de chacune de ces classes.

L'ensemble des graphes sémantiques est utilisé pour établir des statistiques quant à la connexité spatiale des différentes classes dans l'ensemble de la carte. Ces statistiques prennent la forme d'un *graphe conceptuel* $\mathcal{G}_\mathcal{C} = \{V_C, E_C\}$ où V_C est l'ensemble des nœuds correspondant aux différentes classes possibles et E_C l'ensemble des arêtes qui encodent la fréquence de la connexité de deux classes. L'intérêt de $\mathcal{G}_\mathcal{C}$ est de créer un modèle qui permette au robot de connaître ce qu'il est courant d'observer et ce qui ne l'est pas, ceci dans le but de détecter des incohérences dans sa perception du monde extérieur. Par exemple, le graphe conceptuel encode le fait que la route soit fréquemment bordée d'un

trottoir ou qu'on observe jamais de voiture dans le ciel.

Construction du graphe sémantique : Le graphe sémantique est construit par un processus de cartographie $M_4(.) : \mathbb{O}_{\mathcal{L}} \rightarrow \mathbb{O}_{\mathcal{G}_S}$ où $\mathbb{O}_{\mathcal{G}_S}$ est l'espace des graphes sémantiques. A partir de l'image labellisée $\mathcal{I}_{\mathcal{L}}$, les zones sémantiques sont extraites par croissance de région. Un pixel racine est choisi au hasard dans l'image. Les pixels voisins et similaires, c'est à dire de même label, sont accumulés jusqu'à ce qu'il n'y ait plus aucun voisin à ajouter. Le processus est reconduit itérativement jusqu'à ce que toute l'image soit couverte. Une fois les index des pixels appartenant à chacune des zones sémantiques extraits, la forme globale de chacune est estimée en lui associant l'ellipse correspondante, au sens des moindres carrés. Pour cela l'algorithme introduit dans [Fitzgibbon 1995] est utilisé.

Construction du graphe conceptuel : La graphe conceptuel $\mathcal{G}_{\mathcal{C}}$ est construit à partir des N images annotées correspondant à la vérité terrain par un processus de cartographie $M_5(.) : \mathbb{O}_{\mathcal{G}_S} \rightarrow \mathbb{O}_{\mathcal{G}_C}$ où $\mathbb{O}_{\mathcal{G}_C}$ est l'espace des graphes conceptuels. $\mathcal{G}_{\mathcal{C}}$ est initialisé comme un graphe complètement connecté. Chaque arête de $\mathcal{G}_{\mathcal{C}}$ est pondérée par la probabilité de trouver un lien entre deux classes (m, n) dans une image. Si la probabilité associée à une arête est nulle, celle-ci est retirée du graphe.

5.4 Détection d'erreur dans un graphe sémantique

Le forçage de la consistance temporelle des labels a permis d'améliorer sensiblement la qualité de la classification, notamment dans des zones où les variations de luminosité induisent les algorithmes de classification en erreur. Malgré ces améliorations, il subsiste des erreurs dans l'image annotée, dues, par exemple, à la perception d'objets distants qui apparaissent moins contrastés et moins bien résolus et sont donc plus difficiles à reconnaître. Ces objets ont la même apparence sur de longue distances, le filtrage temporel n'est donc pas efficace dans ce cas. Ces erreurs doivent être détectées et éliminées pour ne pas les intégrer dans le graphe sémantique. La détection de certaines anomalies dans la classification des pixels est possible à cette échelle par un raisonnement de haut niveau sur les zones sémantique. En effet en s'appuyant sur le graphe conceptuel, il est possible de détecter des zones incohérentes et de les éliminer du graphe. Ce processus est décrit par l'algorithme 1. Pour chaque arête du graphe sémantique on évalue la probabilité $P_{ij}^{(m,n)}$ d'avoir un lien entre les deux classes m et n associées aux nœuds i et j. Si cette probabilité est nulle dans le graphe conceptuel, c'est à dire que ces deux classes n'ont jamais été observées côte à côte dans une image, alors l'un des deux nœuds n'est pas cohérent. Pour décider du nœud à supprimer, on calcule pour chacun une probabilité $P_c(i)$ que le nœud i soit correct. Cette probabilité est évaluée à partir de la moyenne des probabilités de ses arêtes, pondérée par

la taille de la zone sémantique associée au nœud :

$$P_c(i) = f(x) * \sum_j P_{ij}^{(m,n)} \tag{5.2}$$

où x est le pourcentage de la surface image occupée par la zone sémantique associée au nœud et $f(x)$ est une pondération qui prend la forme d'une sigmoïde :

$$f(x) = \frac{1 + e^{-0.5}}{1 + e^{-x}} \tag{5.3}$$

La sigmoïde privilégie les zones de grande dimension au détriment des zones plus petites en associant des pondérations faibles à ces dernières. La forme de la probabilité vient du fait qu'une classe détectée dans un endroit inhabituel aura tendance à avoir plusieurs voisins incohérents et une surface plus petite.

La correction des graphes sémantiques correspond à la fonction de cartographie $M_8(.)$: $\mathbb{O}_{\mathcal{G}_\mathcal{C}} \times \mathbb{O}_{\mathcal{G}_\mathcal{S}} \rightarrow \mathbb{O}_{\mathcal{G}_\mathcal{S}}$ qui étant donné un graphe sémantique et un graphe conceptuel, produit un graphe sémantique corrigé.

Algorithme 1 Detection des nœuds incohérents

ENTRÉE : $\mathcal{G}_s$, $\mathcal{G}_c$: le graphe sémantique courant et le graphe conceptuel
SORTIE : $\mathcal{G}_s^*$: le graphe sémantique corrigé
pour tout Arêtes $V_{ij} \in \mathcal{G}_s$ **faire**
  Calculer $P(V_{ij})$
  si $P(V_{ij}) < 5\%$ **alors**
    Calculer $P_c(i)$ et $P_c(j)$
    si $P(_c(i) > P_c(j)$ **alors**
      Supprimer nœud j
    sinon
      Supprimer nœud i
    finsi
  finsi
fin pour

5.4.1 Résultats

Pour évaluer l'efficacité de l'algorithme, des tests ont été menés avec les données acquises sur le campus de l'INRIA. Un sous ensemble de 100 images, choisies aléatoirement parmi celles annotées automatiquement, a été utilisé pour le test. La raison d'utiliser un sous-ensemble vient de la nécessité de générer une vérité terrain manuellement pour évaluer l'efficacité de l'algorithme.

Sur l'ensemble des objets incohérents, 35% ont pu être détectés par ce procédé, ce qui correspond à une réduction de 6% des nœuds des graphes sémantiques. Ces zones retirées étant peu fiables, la robustesse de la représentation est donc améliorée par leur retrait. Cependant cette approche ne permet pas de détecter des erreurs correspondant à des situa-

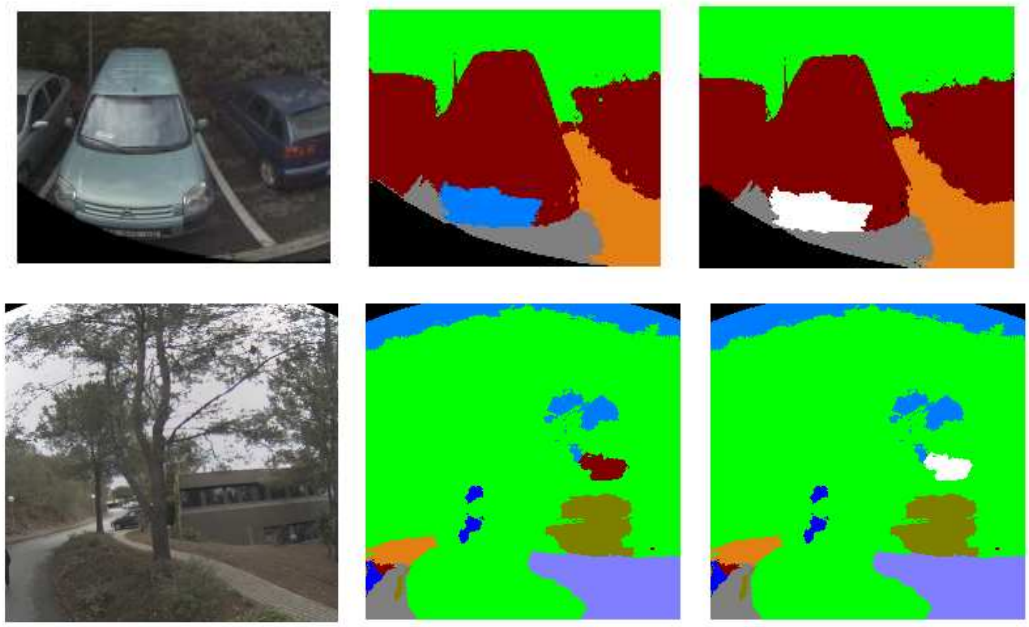


FIGURE 5.3 – *Exemple de cas où la prise en compte du contexte dans le graphe sémantique permet de détecter des erreurs matérialisées en blanc. A gauche l'image RGB ; Au centre l'image labellisée par l'algorithme de classification ; à droite : l'image corrigée par l'algorithme.*

tions jugées plausibles. Deux exemples sont présentés à la figure 5.3. La séquence du haut montre un cas où, malgré le filtrage temporel, le reflet sur la voiture induit l'algorithme de classification en erreur. Une partie de la voiture est labellisée comme "ciel". En analysant le voisinage et à partir des observations faites pendant la phase d'apprentissage, il est possible de déduire que le ciel ne peut pas être visible entre une voiture et la route et qu'il s'agit donc d'une erreur, matérialisée en blanc dans l'image. Dans le second cas, la ligne du bas, l'algorithme de classification détecte une zone labellisée comme "voiture" entre un arbre et le ciel. Ceci n'est pas cohérent et l'algorithme le détecte. Cependant dans ce même exemple, des zones lointaines qui apparaissent entre les arbres sont labellisées à tord comme "signalisation". L'algorithme ne peut pas le détecter puisque dans ce cas, il n'est pas incohérent d'observer la classe "signalisation" à coté de la classe "arbre".

Le procédé basé sur ce qu'il est courant d'observer dans des images permet donc de détecter des erreurs difficiles à traiter par d'autres moyens dès lors que celles-ci correspondent à des événements peu probables.

5.4.2 Conclusion

Ce chapitre a permis d'introduire un nouveau modèle permettant de décrire le contenu sémantique d'une image sphérique de manière compacte, discriminante et robuste, le

graphe sémantique. Ce modèle permet une analyse à haut niveau de la scène courante et peut être utilisé pour construire un modèle de connaissance à partir de l'observation statistique de la co-occurrence des classes dans les images, le graphe conceptuel. Celui-ci permet à son tour une supervision du processus d'annotation et, le cas échéant, une détection des erreurs de labelisation efficace. Outre sa capacité à décrire la scène de manière compacte, l'utilisation de l'information sémantique permet donc aussi un contrôle, au moins partiel, de l'interprétation des données, ce qui est un avantage certain dans un environnement réel où l'estimation du degré de confiance dans les perceptions est fondamental. Le modèle des graphes sémantiques sera particulièrement utile à toutes les étapes de la navigation, comme il sera montré dans la partie III de cette thèse.

CHAPITRE 6

Mise à jour de la représentation

6.1 Introduction

La cartographie d'environnements dynamiques est l'un des principaux défis qui s'opposent encore à la réalisation de modèles fiables de l'environnement. Un robot évoluant dans le monde réel doit être capable de faire évoluer sa représentation du monde sans quoi le modèle peut diverger de la réalité et engendrer des erreurs de localisation ou de planification de trajectoire. Pour mettre à jour le modèle, il faut, d'une part, être capable de détecter les changements survenus dans l'environnement et d'autre part, décider de l'information pertinente à conserver dans la représentation. La détection des changements n'est pas triviale dès lors qu'aucune information n'est disponible a priori sur le comportement dynamique des objets. Ils peuvent survenir de façon arbitraire, régulièrement ou ponctuellement, avec une dynamique rapide ou lente. Pour les détecter, le robot doit mémoriser plusieurs perceptions du même lieu ce qui accroît la taille de la représentation, qui est déjà un problème dans le cas de grands environnements. Le choix de l'information à conserver, quant à lui, résulte d'un compromis connu sous le nom de dilemme stabilité-plasticité. Il n'existe a priori pas de solution optimale et ce choix dépend de l'utilisation finale.

Plusieurs approches ont été proposées dans la littérature. La première consiste à ne conserver que l'information statique, considérée implicitement comme la plus probable [Burgard 2003]. Par exemple [Wolf 2005] propose une carte composée de deux grilles d'occupation, l'une servant à la cartographie des objets dynamiques, l'autre à la modélisation de l'espace considéré comme statique. Ces méthodes sont bien adaptées à la détection d'objets fortement dynamiques mais elles demandent un grand nombre d'observations pour couvrir toute la carte ce qui ramène au problème de la taille des représentations dans le cas de grands environnements. La seconde stratégie consiste à mettre à jour en permanence la carte. La cartographie est donc considérée comme un processus sans fin. L'algorithme bio-inspiré RATSLAM est utilisé dans [Milford 2010] pour cartographier en continu son environnement au prix d'une carte de taille croissante. Une approche similaire est proposée dans [Dayoub 2011] où un modèle de mémoire inspiré de la mémoire humaine est utilisé pour mettre à jour la description d'images sphériques basée sur des caractéristiques locales. Le processus de mise à jour consiste à retirer les vecteurs caractéristiques qui ne sont plus visible dans la sphère courante et à conserver ceux qui sont jugés stables. La mise à

jour d'une représentation en continue, outre le risque d'introduire des objets dynamiques dans le modèle qui seront retirés par la suite, pose le problème de savoir comment maintenir à jour en permanence la totalité de la carte. Il est difficile d'envisager que Google, en dépits des moyens considérables dont dispose l'entreprise, mette à jour quotidiennement *streetview* ! Une troisième approche proposée dans [Biber 2008] ne nécessite pas d'identifier explicitement les objets dynamiques. Plusieurs cartes acquises à des instants différents sont maintenues en parallèle et la meilleure en terme de similitude avec la scène observée à l'instant courant est choisie au moment de la navigation. Outre le problème, là encore, de la taille de la représentation qui nécessite plusieurs cartes entières, cette approche suppose que les changements surviennent de façon homogène dans l'environnement, c'est à dire que la perception locale d'une scène informe sur l'aspect de tout l'environnement. Or si la scène courante ressemble à celle observée à un instant précédent t, rien ne dit que le reste de l'environnement n'ait pas changé significativement. Enfin la dernière famille de méthodes proposées repose sur la transposition du problème dans un autre espace. Un exemple est donné dans [Krajník 2014] qui utilise l'analyse spectrale pour modéliser la dynamique de l'environnement. Les événements dynamiques sont représentés directement par leur fréquence ce qui simplifie la représentations. Cependant les inconvénients majeurs de cette approche tiennent en ce qu'elle nécessite un grand nombre d'observations pour fonctionner et qu'elle ne permet de modéliser que les changements périodiques.

Les contraintes intrinsèques aux grands environnements ne sont donc que peu ou pas prises en compte par les méthodes existantes, notamment le fait qu'il est souvent impossible de réaliser un grand nombre d'observations pour l'intégralité de l'environnement. Ce chapitre a donc pour objectif de présenter une méthode permettant de mettre à jour le modèle de carte hybride métrique-topologique-sémantique présenté dans les chapitres précédents en tenant compte des contraintes liées aux grands environnements.

6.2 Principe

Dans l'approche présentée ici, on propose de mettre à jour la carte directement au niveau de la couche sémantique puisque l'ensemble des processus de navigation s'appuient sur cette couche (voir partie III de cette thèse). Cette approche présente au moins trois avantages. Premièrement, la détection des changements est très simple à ce niveau puisqu'elle se traduit par un changement de label associé à une région. Deuxièmement, la compréhension de scène peut être utilisée pour guider la mise à jour, ce qui n'est pas possible en intervenant à bas niveau. Enfin, l'utilisation de l'information sémantique pour la mise à jour de la carte est d'autant plus intéressante qu'elle permet de distinguer de manière très naturelle les changements temporaires, dus aux déplacements des objets dynamiques, des changements permanent dus à la disparition ou l'apparition d'instances de classes statiques. Elle solutionne donc au moins partiellement le problème du choix de l'information à conserver dans la carte. En effet une modification des labels survenue entre deux images peut avoir

plusieurs origines. Il peut s'agir d'une occlusion temporaire due à la présence d'un objet dynamique, dans ce cas l'observation est rejetée. Il peut s'agir de la disparition d'un objet dynamique, dans ce cas l'observation est acceptée. Enfin il peut s'agir d'une modification structurelle de l'environnement qui se manifeste par le remplacement d'un objet statique par un autre objet statique. Dans ce cas, l'observation la plus récente peut naturellement être privilégiée.

Il existe deux origines principales aux changements qui peuvent survenir dans une scène :

- Les changements d'apparence, qui sont dus principalement à l'alternance jour/nuit, aux conditions météorologiques et aux changements de saisons.
- Les changements dus au déplacement des objets.

Dans le cas des cartes sémantiques, les changements d'apparence doivent être pris en charge par les algorithmes de labelisation puisqu'ils impactent directement la capacité à classer un objet. La méthode présentée se concentre donc sur le problème des changements qui surviennent du fait du déplacement des objets.

Parmi les classes identifiées lors de la phase de classification, on peut distinguer celles qui sont purement statiques, comme les bâtiments, la route ou les arbres, de celles qui sont potentiellement dynamiques comme les voitures. On note C_D les classes dynamiques et C_S les classes statiques. Les classes dynamiques modifient l'aspect de l'environnement principalement en occultant d'autres objets. Pour obtenir une représentation stable, l'objectif est donc de reconstruire l'environnement sans les objets dynamiques qui s'y trouvent au moment de l'acquisition.

Pour cela, un faible nombre d'observations est réalisé et permet de couvrir une partie de l'environnement. Puis la compréhension de scène est utilisée pour généraliser à toute la carte la mise à jour.

6.3 Mise à jour par l'observation

Lorsque le robot parcours son environnement à deux instants différents, il peut observer des changements d'apparence dus au déplacement d'objets dynamiques et certaines zones précédemment occultées peuvent être observées. Ces observations peuvent être utilisées pour construire une représentation plus stable de l'environnement en remplaçant là où c'est possible, les labels d'objets dynamiques par ceux des classes statiques sous-jacentes. Pour ce faire, la transformation entre les images prises à deux instants différents est calculée par une méthode de mise en correspondance dense. Puis une fonction de warping est utilisée pour projeter les labels de la nouvelle observation vers l'image de référence et pouvoir comparer les labels. Le processus est illustré à la figure 6.1.

Soit $\mathcal{I}_s^*$ l'image annotée de référence de dimension $m \times n$ à partir de laquelle on veut calculer une représentation stable. Un pixel de $\mathcal{I}_s^*$ est identifié par sa position $p^* = (u, v)$,

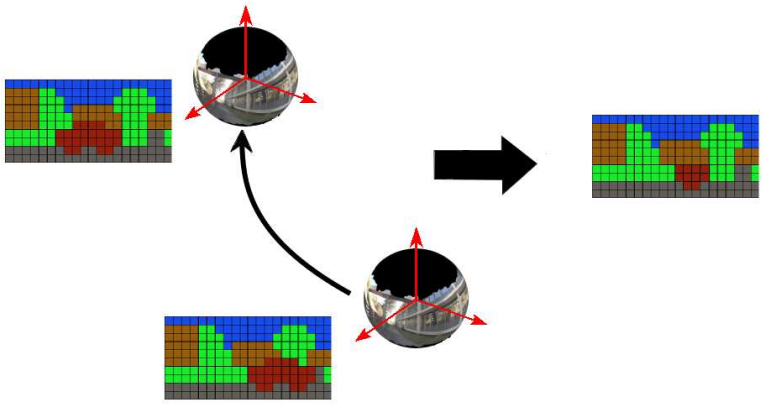


FIGURE 6.1 – *Illustration du processus de warping sémantique. L'image annotée courante est alignée dans le repère de l'image de référence. Les labels sont comparés et l'image de référence est mise à jour à partir des nouvelles observations. Les pixels rouges matérialisent les objets dynamiques.*

où $u \in [0, m[$ et $v \in [0, n[$. Un point 3D de l'espace euclidien est défini par $P^* = \{p^*, Z, l\}$ où $Z \in \mathbb{R}^+$ est la profondeur exprimée dans le repère de l'image de référence et $l \in \mathcal{L}_S$ le label associé. Soit $\mathcal{I}_i$ une autre observation de la même scène à un autre instant et depuis une position légèrement différente $\mathbf{T}(x)_i$, où $\mathbf{T}(x)_i$ est exprimé dans le référentiel de $\mathcal{I}_s^*$.

Il est possible de synthétiser une nouvelle image annotée, notée vI_i', à partir des labels $L(p)$ de $\mathcal{I}_i$ à la position de l'image de référence en utilisant la fonction de warping :

$$p^\omega = \omega(\mathbf{T}(x)_i; Z, l, p^*) \tag{6.1}$$

où $\omega(.)$ transpose le pixel p^* du repère de l'image de référence vers celui de la nouvelle image en utilisant la transformation rigide $T(x)_i$ suivie d'une projection sphérique. Le point projeté ne correspondant pas nécessairement à un pixel, l'interpolation au plus proche voisin est utilisée pour choisir le nouveau label. Finalement, les deux images, de référence et synthétisée, sont comparées. Si un pixel donné p_i est associé à la classe $C_j \in C_D$ dans $\mathcal{I}_s^*$ et à la classe $C_k \in C_S$ dans $\mathcal{I}_i'$, le label choisi est C_k et inversement. Si $C_i \in C_D$ et $C_j \in C_D$, l'image n'est pas mise à jour.

6.4 Généralisation

L'utilisation d'une fonction de warping permet de mettre à jour la carte dans des zones où les nouvelles observations autorisent la découverte des classes occultées par les objets dynamiques. Mais certaines zones restent non observables avec un faible nombre de

séquences d'acquisition. Pour obtenir un modèle stable de l'environnement, ces zones sont considérées comme des trous qu'il faut compléter avec des classes statiques. Pour cela, un modèle d'occlusion est calculé à partir des observations et utilisé pour inférer les classes occultées. Il repose sur l'hypothèse ergodique qui stipule que le comportement moyen des objets dynamiques dans le temps est le même que le comportement moyen des objets dynamiques dans l'espace. Autrement dit, cette hypothèse permet de généraliser les occlusions observées à un endroit donné de la carte à d'autres endroits de l'espace. Cette hypothèse est raisonnable dès lors que dans le cas de grands environnements les cartes ont une taille significative et permettent d'estimer un modèle fiable. L'utilisation de cette hypothèse permet de tirer partie de la grande étendue spatiale du modèle pour compenser le faible nombre d'observation dans le temps. Concrètement, le modèle d'occlusion décrit quelle est la probabilité qu'une classe statique donnée soit occultée par une classe dynamique. Par example, ce modèle encode le fait qu'une classe dynamique "piéton" occulte plus probablement la classe "trottoir" que la classe "ciel" tout comme la classe voiture occulte avec plus de probabilité la classe "route". Ce modèle prend la forme d'une distribution de probabilité qui est calculée à partir des occlusions observées en utilisant deux observations de la même scène. Elle modélise le probabilité de trouver une classe statique C_i étant donnée l'observation d'une classe dynamique C_j :

$$P(C_i|C_j) = \frac{O(C_i, C_j)}{N} \tag{6.2}$$

où N est le nombre total de pixels initialement associés à une classe dynamique et $O(C_i, C_j)$ le nombre de pixels initialement associés à C_j et corrigés avec le label C_i en utilisant la fonction de warping. L'avantage de ce modèle est qu'il n'est pas nécessaire de mémoriser des observations spécifiques qui y sont intégrés d'une manière très compacte.

Mais ce modèle n'est pas suffisant pour estimer correctement les labels des classes occultées parce qu'il ne prend en compte que la probabilité d'observer certaines occlusions au cours du temps. Pour obtenir une prédiction fiable, il faut en plus tenir compte du contexte local qui peut fortement nuancer les prédictions. Ceci se fait via l'utilisation du graphe sémantique associé à l'image annotée. Le graphe encode la connexité des zones sémantiques de l'image. Pour une zone correspondant à une classe dynamique, il donne donc les zones adjacentes qui sont partiellement occultées et qui forment le voisinage, noté $\mathcal{N}$. Chaque nœud du graphe sémantique est caractérisé par une ellipse englobante f_i pour décrire sa forme. Cette ellipse est définie par sa position $q_i = (u_i, v_i)$, la taille des demi-axes de l'ellipse englobante $s_i = (h_i, w_i)$ et son orientation α_i. Pour modéliser la probabilité d'associer un label donné à un pixel $p = (u, v)$, une fonction gaussienne est associé à chaque nœud voisin dans le graphe $n_i \in \mathcal{N}$. Elle prend la forme générale

$$F_i(u, v) = A_i exp^{(-a(u-u_i)^2+2b(u-u_i)(v-v_i)+c(v-v_i)^2)} \tag{6.3}$$

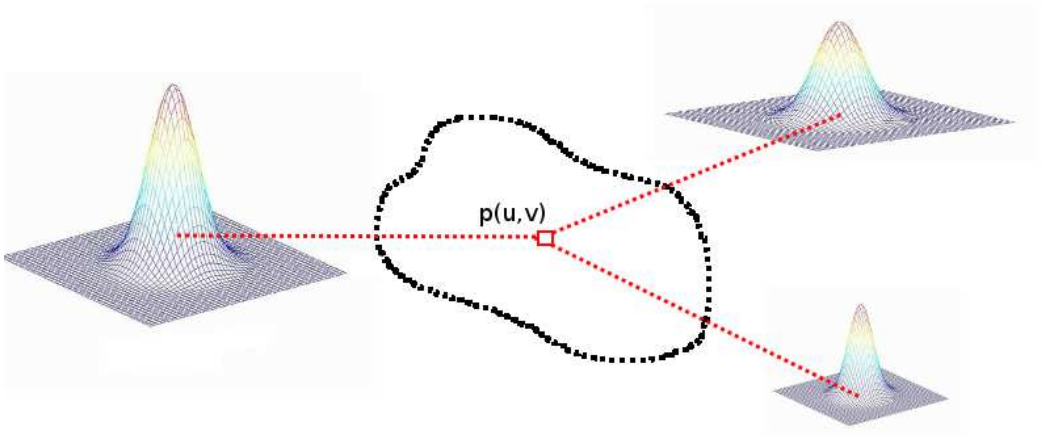


FIGURE 6.2 – *Illustration du processus de d'inférence. A trois objets voisins d'une zone à compléter sont associés trois gaussiennes dont les propriétés dépendent de la forme et de l'orientation de l'ellipse englobante associée. Le label associé au pixel $p(u, v)$ est donné par l'équation 6.7 en fonction des valeurs relatives de ces gaussiennes.*

où A_i est l'amplitude de la gaussienne, donnée par $P(C_i|C_j)$ et où :

$$a = \frac{cos^2\theta}{2\sigma_u^2} + \frac{sin^2\theta}{2\sigma_v^2} \tag{6.4}$$

$$b = \frac{sin(2\theta)}{4\sigma_u^2} - \frac{sin(2\theta)}{4\sigma_v^2} \tag{6.5}$$

$$c = \frac{sin^2\theta}{2\sigma_u^2} + \frac{cos^2\theta}{2\sigma_v^2} \tag{6.6}$$

avec $\sigma_u = h_i$,$\sigma_v = w_i$ et $\theta = \alpha_i$.

Pour chacun des pixels de la zone à compléter, le label le plus probable est donné par :

$$C(p) = \max_{i \in \mathcal{N}}(F_i(u, v)) \tag{6.7}$$

où C est la nouvelle classe associée au pixel p.

Avec cette approche, il est possible de mettre à jour la carte en exploitant à la fois le contexte local et l'expérience acquise par le robot au long de son exploration de l'environnement. Ceci permet de créer une représentation robuste et stable de l'environnement qui contrairement à beaucoup d'autres approches, ne nécessite pas la mémorisation d'une grande quantité d'observations mais d'un simple modèle compact. L'efficacité de la méthode est démontrée dans la section suivante et son utilité pour la navigation dans le chapitre 8.

6.5 Expériences

6.5.1 Méthode

Les expériences sont réalisées en utilisant une base de données d'images sphériques acquises sur le campus de l'INRIA de Sophia-Antipolis composée de deux séquences d'acquisition réalisées à trois ans d'intervalle. Chaque séquence est composée de plusieurs milliers d'images (voir chapitre 4) annotée avec les 9 classes suivantes : ciel, arbre, route, panneau de signalisation, trottoir, bâtiment, voiture, signe au sol (route), autres. La classe voiture représente les objets dynamiques. L'algorithme de classification produit des résultats imparfaits. Il est par ailleurs important de noter que l'apprentissage est fait uniquement sur la première séquence de la base de données, les résultats de la labelisation pour la deuxième séquence gèrent donc intégralement les variations de luminosité entre les deux acquisitions.

Pour évaluer les performances de l'algorithme proposé, deux mesures sont rapportées :
- La précision globale, donnée par le nombre de pixels correctement associés à la classe C_i par rapport au nombre total de pixels labellisés comme appartenant à C_i.
- La précision par classes non pondérée

La première mesure donne une vision globales des performances de l'algorithme mais peut donner une vision erronée lorsqu'une classe est très majoritaire. Dans ce cas, labelliser l'intégralité des pixels avec cette classe n'est que faiblement pénalisé. La seconde mesure permet de donner le score relatif de chaque classe, indépendamment de sa taille relative. Pour disposer d'une vérité terrain permettant l'évaluation, les prédictions sont réalisées pour les tests en inférant les classes occultées dans des zones où la seconde séquence d'observation permet d'avoir accès aux labels. Le modèle d'occlusion calculé à partir de l'expérience acquise lors des expérimentations est donné à la table 6.1. Les résultats du processus d'inférence sont donnés à la table 6.2 et illustrés à la figure 6.3.

Les expériences ont été réalisés sur un ordinateur doté d'un processuer Intel i7-3840QM CPU cadensé à 2.80GHz. Tous les programmes sont mono-threadés.

6.5.2 Résultats et analyse

Comme attendu le modèle d'occlusion encode le fait que les voitures sont le plus souvent sur la route et qu'elles cachent partiellement les arbres ou les immeubles devant lesquels elles passent. Les résultats présentés à la table 6.2 montrent pour les classes ayant une probabilité non nulle, que le processus d'inférence est très efficace pour découvrir les classes occultées avec un très grand nombre de pixel associés à la classe correct. Ces bons résultats sont dus à plusieurs causes. Premièrement, le fait de prendre en compte à la fois le contexte spatial local via le graphe sémantique et les probabilités déduites des observations faites sur de longues distances, permet de construire un modèle précis des occlusions et donc de faire des prédictions avec un bon niveau de performances. La deuxième raison

TABLE 6.1 – *Modèle d'occlusion*

Classe	Probabilité associée
ciel	0
batîment	0.04
route	0.76
trottoir	0.08
arbre	0.11
Panneau	0
Signe au sol	0

TABLE 6.2 – *Résultats du processus d'inférence*

Classe	Score
bâtiment	0.96
route	0.98
trottoir	0.99
arbre	0.97
total	0.98

tient au fait que dans la plupart des cas, les classes statiques ne sont que partiellement occultées par les objets dynamiques, la prise en compte du contexte permet donc d'évaluer correctement la forme que devrait avoir les objets occultés et de remplir correctement le trou laissé dans l'image par l'objet dynamique. Si une classe était complètement occultée, elle ne pourrait bien évidement pas être prédite avec cette méthode. Cependant il y a une faible la probabilité que cela se produise. En effet les zones qui restent occultées après la superposition de plusieurs séquences d'acquisition via la fonction de warping, sont de petite dimension. Il est donc peu probable qu'un objet soit totalement occulté derrière cette zone. Si c'est effectivement le cas, alors sa taille modeste influence peu le résultat global. Enfin, il convient de noter un autre point intéressant qui est illustré à la figure 6.3. Les quatre images du haut montrent deux images RGB et deux images labellisées du même endroit, issu des deux séquences d'acquisition. Sur la paire d'image de droite, l'algorithme de classification génère une erreur en associant à une partie de la route le label voiture. Le processus de mise à jour permet de corriger cette erreur. De ce fait, la mise à jour de la carte peut améliorer la qualité de l'annotation des images et donc rendre la représentation plus robuste.

Les résultats montrent que même avec un faible nombre d'observations, qui est compatible avec la mise en œuvre réaliste de robots dans des environnements réels, il est possible d'obtenir une représentation stable de laquelle sont retirés les objets dynamiques et où l'information manquante est complétée. Ceci est permis par un raisonnement à haut niveau

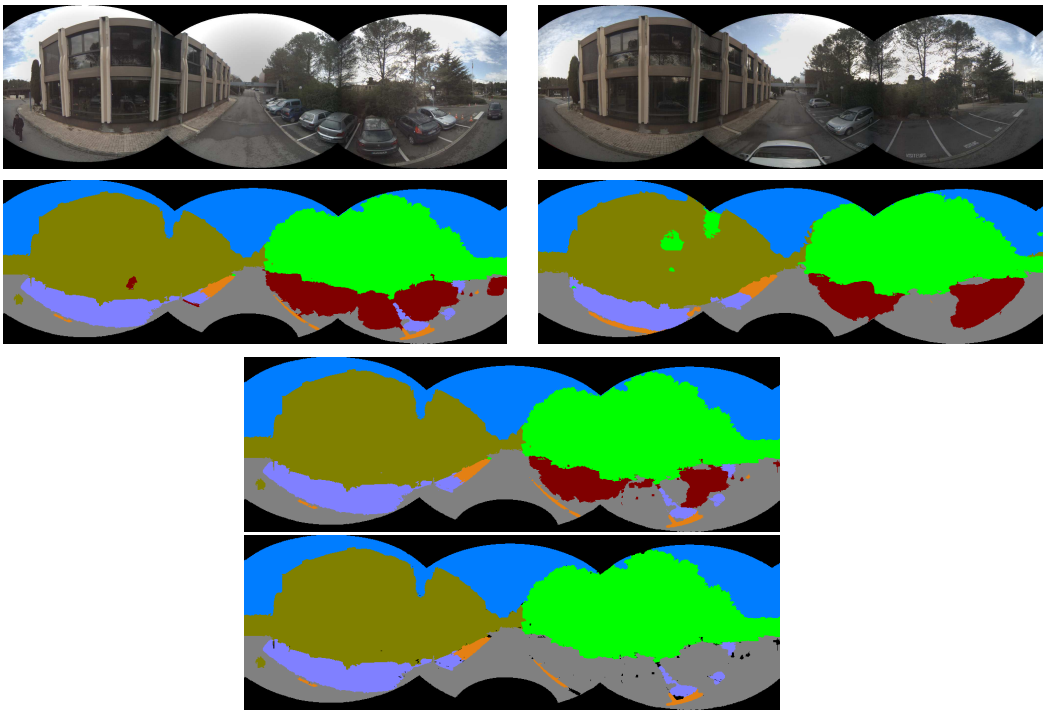


FIGURE 6.3 – *Exemples de résultats pour le processus complet de mise à jour. En haut à gauche image RGB de la première séquence d'acquisition*

rendu possible par l'utilisation de l'information sémantique et l'exploitation du contexte.

CHAPITRE 7

Extrapolation de Cartes d'Environnements Dynamiques

7.1 Introduction

L'information contextuelle est de plus en plus utilisée comme un moyen d'augmenter les limites perceptuelles des robots, comme cela a été montré dans le cas de l'annotation d'images. Ceci est largement motivé par leur intégration dans des environnements de complexité croissante pour lesquels les connaissances a priori ne suffisent plus à rendre les robots autonomes. L'exploitation des connaissances explicites mais aussi implicites, joue un rôle déterminant dans l'adaptation des robots à des environnements réels pour lesquels l'information disponible est souvent partielle et doit être complétée. La capacité à cartographier l'environnement est depuis longtemps considérée comme un prérequis au développement d'algorithmes de planification intelligents qui permettent au robot d'évoluer en sécurité tout en menant à bien ses missions. Initialement, la plus grosse partie des efforts de recherche dans ce domaine a porté sur la manière de générer des modèles de plus en plus précis de l'environnement qui se sont révélés insuffisants pour modéliser les environnements réels. Les stratégies de cartographie ont donc connu un développement vertical, avec l'intégration dans les modèles générés, de couches de connaissances correspondant à des concepts de plus en plus abstraits, comme cela a été fait dans le chapitre 5. Mais l'exploitation du contexte spatial comme un moyen de compléter la cartographie a été peu étudié.

Dans le domaine des environnements structurés, [O'Callaghan 2012] ont développés un mécanisme pour inférer l'espace occupé dans une zone non observée via l'utilisation d'un processus gaussien entraîné sur des cartes construites préalablement et d'une fonction de covariance qui s'adapte au voisinage du point donné. Dans le même esprit mais à un plus haut niveau, [Pronobis et Jensfelt 2012] ont développé une approche qui permet d'inférer la nature d'un environnement partiellement observé à partir de celle des lieux déjà visités. Dans ces deux approches, le contexte spatial, géométrique ou sémantique, est exploité pour ajouter de manière explicite des informations contenues implicitement dans la même couche de la carte. L'information contextuelle est donc utilisée uniquement horizontalement. Par ailleurs, le contexte utilisé dans ces approches est purement statique et aucun moyen n'est envisagé pour utiliser le contenu dynamique de la scène, qui est pourtant une

riche source d'informations.

L'utilisation d'objets dynamiques pour aider la cartographie a été abordée par [Grzonka 2010] qui présente une méthode pour construire des cartes approximatives en utilisant le suivi de la trajectoire des humains pour évaluer la structure de l'environnement ainsi que l'ouverture et la fermeture de portes comme candidats pour la fermeture de boucle et la délimitation des pièces. Mais l'intérêt de ce travail est limité par la nécessité pour les humains de porter des vêtements spécialement conçus pour être détecté par le système de capture de mouvement. Une approche intéressante a été présentée par [Jiang 2013] où le contexte de la scène est exploité via l'identification d'un jeu de poses (statiques) adoptées par les humains et la co-occurrence de ces poses avec des objets particuliers, pour déduire la sémantique de la scène. Mais cette approche bottom-up vise uniquement à labelliser la scène sans chercher à dépasser les limites perceptuelles du robot.

Dans la suite, on propose d'utiliser le contexte à la fois sémantique et dynamique pour étendre la carte de l'environnement à des zones non observées mais dont l'existence peut être déduite des données acquises. Un mécanisme permettant la circulation de l'information à la fois verticalement et horizontalement dans la carte est présenté, utilisant les indices contextuels d'une couche de la carte pour compléter l'information contenue dans une autre. Contrairement à la plupart des chapitres, les méthodes développées dans celui-ci seront appliquées aux environnement intérieurs, sans perte de généralité. Deux raisons justifient ce choix : d'une part le contexte est un vecteur d'information davantage significatif dans des environnements structurés ; d'autre part, la nécessité d'obtenir une vérité terrain fiable pour mesurer quantitativement les résultats de la méthode proposée ont naturellement conduit a choisir un environnement intérieur pour la conduite des expériences. Bien que le modèle de carte reste le même, des ajustement mineurs ont été introduits et seront précisés dans la suite.

7.2 Approche

Lorsqu'un robot cartographie un environnement réel, il subsiste toujours des zones qui n'ont pu être observées, produisant une carte incomplète. Les raisons de cette incapacité tiennent essentiellement en la présence de barrières physiques entre le robot et la zone à observer qui peuvent être le fait de phénomènes statiques, quasi-statiques ou dynamiques. L'objectif ici est d'identifier ces éléments et de les utiliser comme des indices contextuels permettant d'étendre le champ de perception du robot par le raisonnement. Parmi l'ensemble des classes identifiables dans un environnement, un groupe spécifique présente un intérêt tout particulier pour la navigation. Il s'agit des objets qui connectent des espaces entre eux, comme les portes, les escaliers ou les ascenseurs. En effet leur identification permet au robot d'inférer l'existence d'autres lieux qu'il n'a pas visités lors de la phase de cartographie. La présence d'instances de ces classes dans la scène est donc une source très riche d'information à la fois sur la topologie mais aussi sur la géométrie de l'environnement. De

même, l'observation d'objets dynamiques, et tout particulièrement d'êtres humains, permet de comprendre l'existence de zones navigables au delà des limites perceptuelles du robot. En extrapolant leurs trajectoires en dehors de la zone visible, il est en effet possible d'évaluer la présence d'espace libre. Par exemple un humain qui disparaît derrière un mur ou qui passe à travers une porte permet au robot d'inférer l'existence d'espace navigable de l'autre coté, même s'il n'est pas observable. Ainsi la détection et l'exploitation de ces éléments permet de donner une nouvelle dimension à la cartographie au travers du processus d'extrapolation détaillée dans la section suivante.

Ce travail s'appuie en parti sur le modèle présenté dans [Papadakis 2014] pour représenter l'espace libre autour d'un être humain, qui est généralisé à des objets statiques et dynamiques. Compte tenu des spécificités des environnements intérieurs, à savoir la nature structurée des scènes, les méthodes de labellisation employées dans la cadre des environnements extérieurs ont été remplacé ici par des méthodes de détection spécifiques exploitant la géométrie. Ces méthodes sont détaillées dans la section suivante. Par ailleurs le capteur a été remplacé par une Asus Xtion pro qui permet l'acquisition simultanée d'images et de données de profondeur.

7.3 Modèle

L'extrapolation s'intègre dans le processus de cartographie en s'appuyant sur les cartes locales et une *fonction d'extrapolation* dont les paramètres dépendent de la classe considérée. Ici, on dénote une carte locale par $S = \{P; L; U\}$ où P est l'ensemble des points 3D produit par le capteur et correspondant à la couche photométrique et géométrique, L un ensemble de groupes de points indexant les régions correspondant aux objets identifiés et formant la couche sémantique et U un ensemble de cellules 2D d'une grille d'occupation modélisant l'espace libre dans la carte locale, utilisée pour la navigation. Étant donnée une carte locale S_i dans laquelle une instance est identifiée, cette fonction propage l'information soit à l'intérieur de cette même carte, soit dans le graphe global au travers de la création d'une nouvelle carte locale S_j. La forme générale de cette fonction est :

$$\mathcal{M}(S_i) = \delta_{ij} S'_i + (1 - \delta_{ij}) S_j \tag{7.1}$$

où δ_{ij} est le symbole de Kronecker, i dénote l'index de la carte locale dans laquelle se trouve le robot et j l'index de la carte locale nouvellement créée par le processus d'extrapolation.

L'extrapolation de carte s'intègre dans le processus global de cartographie comme un processus $M_6(.) : \mathbb{O}_{\mathcal{L}} \rightarrow \mathbb{O}_{\mathcal{G}} \times \mathbb{O}_{\mathcal{S}}$ où $\mathbb{O}_{\mathcal{L}}$ est l'espace des images sphériques annotées, $\mathbb{O}_{\mathcal{S}}$ celui des sphères RGBD et $\mathbb{O}_{\mathcal{G}}$ celui des graphes de sphères.

7.3.1 Extrapolation basée objet statique

Les objets statiques sont ici représentés par la classe "escalier". La détection des instances de cette classe se fait en plusieurs étapes. Dans un premier temps, le nuage de point P_i de la scène S_i est filtré par des voxels de grande dimension pour ne conserver que la structure principale de la scène. Puis les surfaces planes sont extraites par l'algorithme RANSAC et analysées. Si l'un des plans forme un angle autour de 45^o avec l'axe vertical et que sa surface est supérieure à un seuil fixé, l'objet est labellisé comme un escalier. La position de l'escalier $\mathbf{t}$ est identifiée au centroïde du plan. L'orientation $\mathbf{o}$ de l'escalier est calculée à partir de la normale du plan $\mathbf{n}$ et du vecteur $\mathbf{w} = (\mathbf{n} \times \mathbf{z}) \times \mathbf{n}$ où $\mathbf{z}$ est l'axe vertical et $\times$ le produit vectoriel. La direction correspondant au sens ascendant de l'escalier est donnée par $\mathbf{o} = \mathbf{w}.sgn(\mathbf{w^T z})$. Les escaliers sont par ailleurs caractérisés par une largeur w et une hauteur h correspondant à la hauteur attendue de l'étage suivant. Le vecteur des paramètres associés à une instance prend la forme $\mathbf{a_s} = (\mathbf{t^T}, \mathbf{o^T}, (h, w)^T)$. Un exemple tiré des expériences est donné à la figure 7.1.

Une fois que l'escalier est détecté, une nouvelle carte locale S_j est générée et ajoutée au graphe global. Elle est connectée à S_i par une arête E_{ij} correspondant à la position attendue du haut des escaliers exprimé dans le référentiel de S_i. S_j est initialisée en copiant les données relatives à l'instance faisant le lien avec S_i , à savoir l'escalier, qui sont considérés comme partagées entre les deux cartes locales de telle façon que $S_j = \{P_S, R_S, \emptyset\}$ où P_S correspond à l'ensemble des points formant l'escalier et R_S le groupe correspondant. Puis la carte nouvellement créée est augmentée de telle façon que $S_j = \{P_S, R_S, U_j\}$ où U_j est obtenue par extrapolation de l'espace libre suivant la règle :

$$U_j = \{\mathbf{u} \in \mathbb{R}^2 | f_s(\mathbf{u}) > p_{th}\} \tag{7.2}$$

où f_s est identifiée à une loi normale 2D $\mathcal{N}(\mathbf{0}, \Omega)$ avec la matrice de covariance $\Omega = I\sigma^2$, σ étant proportionnel à la largeur w de l'escalier et $\mathbf{u} \in U_j$ un point arbitraire de la carte d'occupation locale. La fonction f_s encode la confiance en l'existence d'espace libre jusqu'à une certaine distance de l'origine $\mathbf{0}$ de S_j, contrôlé par la valeur de p_{th} le seuil de confiance fixé, comme illustré à la figure 7.2 pour le cas d'objets quasi-statiques.

7.3.2 Extrapolation basée objet quasi-statique

Le terme quasi-statique fait référence à des objets qui peuvent se présenter sous différents états lors de l'exploration. Par exemple, les portes, qui forment la classe utilisée ici, peuvent être ouvertes ou fermées mais ne constituent pas pour autant des objets dynamiques. Elles sont détectées de la manière suivante. Si un plan parmi ceux détectés précédemment est perpendiculaire au sol, alors il est analysé plus en détails. Des points saillants sont extraits dans l'image 2D correspondant aux points du plan et sont repartis en deux groupes : (i) ceux proches du sol et qui sont candidat pour les bords inférieurs de la porte, (ii) ceux qui sont proches à la hauteur de la partie supérieure porte. Ensuite on

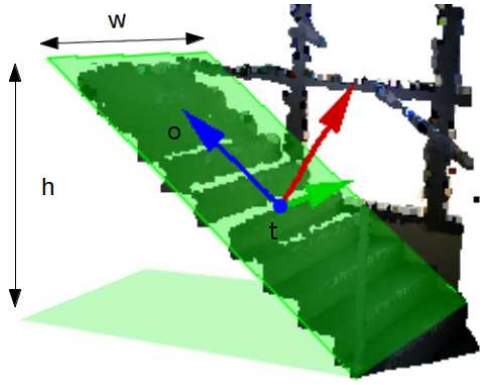


FIGURE 7.1 – *Exemple d'escalier identifié dans les expériences.*

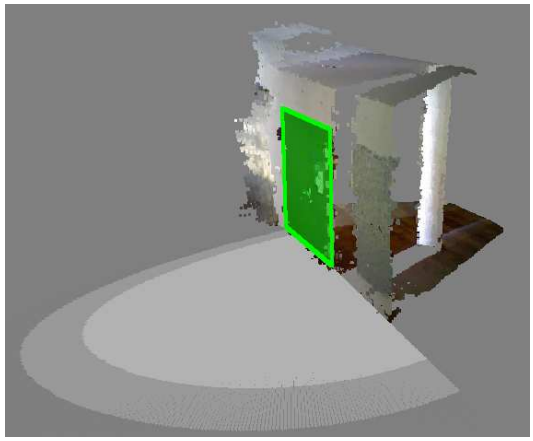


FIGURE 7.2 – *Exemple d'espace libre modélisé par la fonction f_i dans le cas d'une porte : l'estimation de l'espace libre est matérialisée en gris clair pour une probabilité supérieure à 0,6 et en gris foncé si supérieure à 0,5.*

recherche des couples de points dont un élément appartient à chaque ensemble et qui sont verticalement alignés. Pour chaque paire, on recherche alors s'il existe une autre paire à la distance correspondant à la largeur attendue d'une porte. Finalement, la position de l'axe de rotation de la porte est identifiée en cherchant la poignée de porte. Pour cela on utilise l'hypothèse raisonnable que des points d'intérêt ont tendance à se concentrer autour de la poignée de porte et on compte simplement le nombre de ces points aux deux positions possibles, à droite et à gauche de la porte. Le coté contenant le plus grand nombre de point est alors utilisé pour identifier l'axe de rotation de la porte (voir figure 7.3).

Une fois la porte détectée, sa position $\mathbf{t}$ est fixée comme le centroïde du plan correspondant. L'orientation $\mathbf{o}$ est calculée à partir de la normale du plan $\mathbf{n}$ et $\mathbf{z}$ l'orientation du robot de telle façon que $\mathbf{o} = \mathbf{n}.sgn(\mathbf{n}^T\mathbf{z})$. Comme dans le cas des objets statiques une nouvelle carte locale S_j est ajoutée au graphe global. L'arête E_{ij} qui relie la carte courante S_i à S_j, correspond à la position relative de S_j par rapport à S_i exprimée dans le référen-

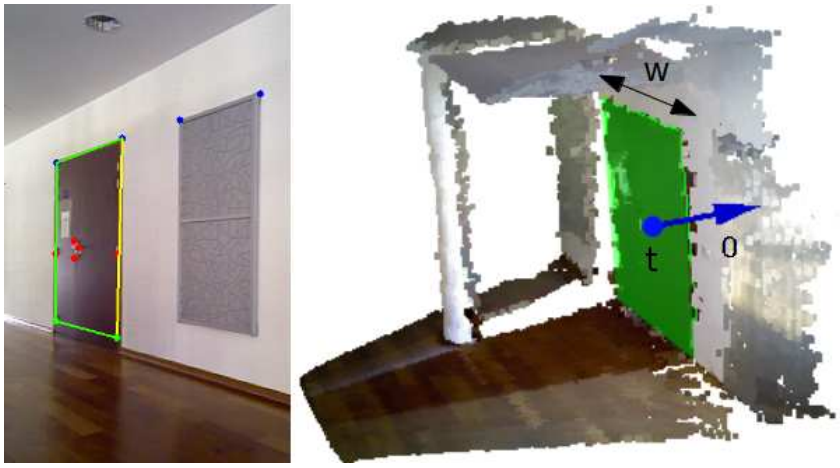


FIGURE 7.3 – *Exemple de porte identifiée dans les expériences. A gauche : les point rouge matérialisent les points d'intérêts détectés. Comme attendu ils se concentrent autour de la poignée et permettent d'identifier l'axe de rotation de la porte, représentée en jaune. A droite : exemple de porte détectée avec les paramètres caractéristiques.*

tiel de S_i. La nouvelle carte locale est positionnée à la distance d derrière la porte, dans la direction donnée par $\mathbf{o}$. Elle est initialisée par $S_j = \{P_D, R_D, \emptyset\}$ où P_D et R_D correspondent respectivement aux points 3D et au groupe de la porte. Puis la carte est mise à jour par $S_j = \{P_D, R_D, U_j\}$ où U_j est obtenue par extrapolation de l'espace libre suivant la règle :

$$U_j = \{\mathbf{u} \in \mathbb{R}^2 | f_d(\mathbf{u}) > p_{th}\} \tag{7.3}$$

où $f_d(.)$ est une loi normale asymétrique 2D denoté par $\overline{\mathcal{N}}$ et centrée en $\mathbf{t}'$, la position de l'axe de rotation de la porte dans le référentiel de S_j. L'espace libre prédit est illustré à la figure 7.2. La famille des lois de distribution normales asymétriques généralisent la distribution normale en ajoutant un paramètre dit de forme qui contrôle le degré d'asymétrie de la fonction. Ceci est réalisé en donnant à f_d la forme de la distribution de probabilité $\tilde{\mathcal{N}}(\mathbf{t}', \Omega, \alpha)$ définie comme :

$$f_d(\mathbf{u}) = 2\phi(\mathbf{u})\Phi(\alpha^T \mathbf{u}) \tag{7.4}$$

où $\mathbf{u} \in U_j$ est une cellule de la carte d'occupation locale, ϕ dénote la fonction de probabilité $\mathcal{N}(\mathbf{0}, \Omega)$ de matrice de covariance $\Omega = I\sigma^2$, Φ est la fonction de répartition de ϕ et $\alpha = (\alpha_1, \alpha_2)^T$ est le vecteur de paramètres de la loi normale asymétrique qui contrôle le degré d'asymétrie. $\mathbf{0}$ est l'origine dans le référentiel de S_j.

L'utilisation d'une loi normale asymétrique (NA) se justifie par le fait qu'une porte, contrairement à un escalier, n'est pas symétrique du fait de son axe de rotation. La loi NA permet de modéliser cette asymétrie en modulant le vecteur des paramètres de la fonction α suivant l'orientation de la porte et sa largeur w. Sur cette base, l'asymétrie est introduite

en posant $\alpha = (0, c \cdot w)^T$, le vecteur de paramètres de la porte s'écrit donc finalement $\mathbf{a}_d = (\mathbf{t}^T, \mathbf{o}^T, \alpha)^T$. Un exemple de porte détectée pendant les expériences est donné à la figure 7.3.

7.3.3 Extrapolation basée objets dynamiques

A travers la détection et l'analyse du déplacement des personnes, il est possible d'extrapoler leur trajectoire dans des zones non observables. Il est notamment possible d'étendre l'estimation de l'espace libre en faisant l'hypothèse, raisonnable en environnement intérieur, que l'espace traversé par des personnes est toujours un espace libre pour le robot. La detection d'êtres humains est réalisée grâce à la librairie OpenNI [Shotton 2011]. Dans le cas présent l'extrapolation d'information est réalisée dans la sphère courante S_i puisque la position des humains est arbitraire et évolue constamment. Deux scénarios sont envisagés. Dans le premier, un humain entre ou sort de la scène, soit par une porte soit par le bord du champ de perception. Dans ce cas, l'extrapolation de la trajectoire permet d'extrapoler l'existence d'espace libre avant ou après que l'être humain soit apparu. Dans le second scénario, un humain est partiellement observable, par exemple le haut du corps est visible alors que le bas est caché par un obstacle. Dans ce cas, l'extrapolation de l'espace libre est effectuée en attribuant de l'espace libre autour de l'être humain tout au long de sa trajectoire. Pour réaliser ce scénario, on associe à la position de l'être humain un noyau non stationnaire qui quantifie la notion d'espace personnel tel qu'introduit par [Hall 1966] dans la théorie de la proxémie.

Le première étape consiste à identifier l'être humain et à identifier les paramètres attribués à cette classe. Notamment, on collecte : (i) la position 2D de l'être humain $\mathbf{t} = (t_x, t_y)^\mathbf{T} \in \mathbb{R}^2$, (ii) l'orientation $\theta \in [0, 2\pi[$ et (iii) son coté dominant $d \in \{-1, +1\}$ ou $-1, +1$ correspondent respectivement au coté gauche et droit. Le coté dominant peut être identifié implicitement à partir de l'espace latéral minimum entre l'humain et l'espace occupé dans la carte métrique, en faisant l'hypothèse que l'être humain conserve un distance plus petite du coté dominant. Le vecteur des paramètres associé à l'être humain est alors $\mathbf{a}_h = (\mathbf{t}^T, \theta, d)^T$.

Le processus d'extrapolation est alors conduit en mettant à jour la carte locale S_i de telle manière que $S_i' = \{P; R; U'\}$ où $U' = U \cup U_i$, U_i étant obtenue par extrapolation de l'espace libre par la formule :

$$U_i = \{\mathbf{u} \in \mathbb{R}^2 | f_h(\mathbf{u}) > p_{th}\} \tag{7.5}$$

où la fonction $f_h(.)$ est une loi de probabilité asymétrique 2D, notée $\tilde{\mathcal{N}}(\mathbf{t}, \Omega, \alpha)$ et $\mathbf{u} \in U'$ une cellule de la carte d'occupation locale. La fonction $f_h(.)$ permet de quantifier l'espace personnel associé à une personne que l'on exploite pour extrapoler l'espace libre le long de la trajectoire de l'être humain. Un exemple est donné à la figure 7.4 à gauche, où l'on visualise l'espace personnel exprimé par la fonction $f_h(.)$.

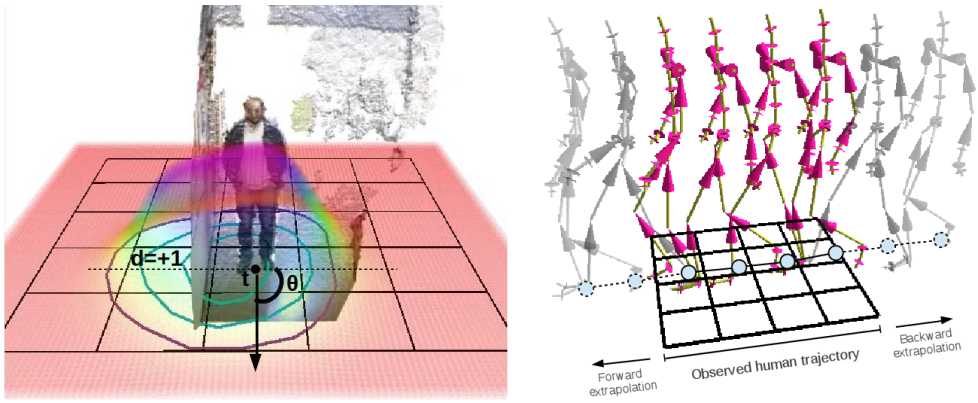


FIGURE 7.4 – *À gauche : illustration de l'espace libre attribué à une personne. À droite : trajectoire prédite lorsqu'une personne entre et sort du champ de vision du robot.*

L'extrapolation de la trajectoire de l'être humain dans le scénarios où il apparaît ou disparaît de la scène, est faite par une prédiction linéaire, s'appuyant sur l'hypothèse que la trajectoire typique de l'être humain se compose essentiellement de segments droits. Ceci est d'autant plus raisonnable que la prédiction de la trajectoire est faite sur une courte fenêtre temporelle. Le formule pour obtenir le vecteur de paramètres extrapolé est $\mathbf{a}_h(t+k) = \mathbf{a}_h(t) + k \cdot \dot{\mathbf{a}}_h(t)$ où t est la date à laquelle l'être humain a été observé pour la dernière fois et k la date de la prédiction souhaitée (positive ou négative). La figure 7.4 à droite donne un exemple de trajectoire extrapolée en dehors de la zone visible.

7.4 Expériences

Pour évaluer l'utilité de l'approche proposée, une série d'expériences en environnement intérieur a été conduite avec un robot Neobotix MP-500. Le robot est équipé d'un laser 2D frontal qui a été utilisé comme un moyen d'obtenir une estimation précise de l'espace 2D occupé. L'acquisition des données RGB-D a été faite avec une Assus XtionPro Live qui permet d'enregistrer des nuages de points texturés à 30 Hz.

Les expériences se sont déroulées dans un environnement varié contenant des bureaux, des couloirs, des escaliers, un hall etc. Elles ont été conduites d'après le schéma suivant : le robot navigue dans son environnement piloté par un opérateur et rencontre un certain nombre d'instances de classes d'intérêt (statique, quasi-statique et dynamique) le long de sa trajectoire. Pour faciliter le traitement des informations, lorsqu'un être humain est détecté, le robot est stoppé pour minimiser les interférences entre son mouvement propre et celui de l'être humain. La détection d'une classe donnée déclenche l'appel à la fonction d'extrapolation correspondante. Pour évaluer la qualité de l'extrapolation, une carte de l'intégralité de l'environnement a été réalisée au préalable, faisant office de vérité terrain.

7.4.1 Extrapolation de graphe

Pour estimer la capacité de l'algorithme à ajouter de nouveaux nœuds dans le graphe global lorsque nécessaire, les performances du détecteur d'objets statiques et quasi-statiques ont été évaluées. Deux mesures sont rapportées : le rappel par objet, qui se définit comme le ratio du nombre d'objets effectivement détectés sur le nombre total d'objets effectivement présents pour toute la séquence d'acquisition, le rappel par image, qui se définit comme le ratio du nombre d'objets détectés sur le nombre d'images dans lesquelles ils sont visibles. Les résultats sont présentés à la table 7.1.

TABLE 7.1 – *Résultats de la détection*

Classe	Rappel Image	Rappel Objet
Stairs	0.95	1
Doors	0.67	1

Dans le cas des escaliers, on atteint un très haut niveau de détection, à la fois pour le rappel objet et pour le rappel image. Ceci s'explique par le fait que la structure d'un escalier est très particulière et est relativement peu sensible au bruit de mesure du fait de sa taille. La détection des portes quant à elle, donne un bon score pour le rappel objet puisque toutes les portes sont détectées mais un score plus faible pour le rappel image. Ceci vient du fait que pour détecter une porte, celle-ci doit être visible intégralement. Il faut en effet pouvoir identifier les quatre coins pour la détecter. Dans les images pour lesquelles seule une partie de la porte est visible, la détection échoue.

La figure 7.5 démontre clairement l'intérêt de l'utilisation du contexte sémantique pour compenser le manque d'information par l'extrapolation d'informations de nature spatiales. Le graphe extrapolé est visible en bas à droite et la vérité terrain en bas à gauche. On constate que pour chacune des pièces non visitées ainsi que l'étage supérieur correspond une carte locale extrapolée. Ceci démontre clairement la capacité à capturer la topologie de l'environnement au delà des limites perceptuelles du robot. Dans le cas présent, au lieu de considérer que l'environnement est un simple hall, le robot est à même de percevoir l'existence de trois autres pièces et d'un autre étage, augmentant significativement le degré de connaissance qu'a le robot de son environnement.

7.4.2 Extrapolation d'espace libre

En plus d'augmenter la connaissance de la topologie du lieu exploré, l'approche proposée ici permet aussi d'étendre la perception des zones navigables. Pour évaluer cette capacité, espaces libres prédit et effectif ont été comparés pour 3 niveaux de confiance différents, à savoir $p_{th} \in \{0.5, 0.6, 0.7\}$. Pour mesurer les performances de l'algorithme, on mesure la précision P, définie comme le ratio du nombre de vrais positifs (VP), c'est

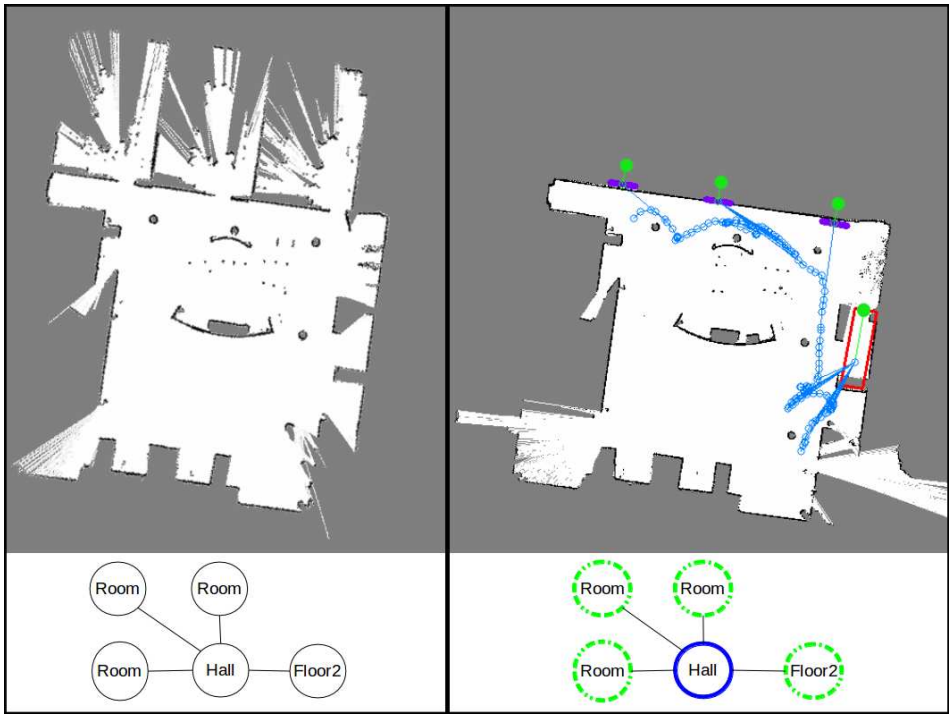


FIGURE 7.5 – *A gauche : vérité terrain pour l'espace occupé (en haut) et la topologie de l'environnement (en bas). A droite, en haut : carte de l'environnement de test avec la trajectoire du robot superposée. Les noeuds bleus correspondent aux cartes locales observées, les noeuds verts aux cartes locales extrapolées. Le rectangle rouge matérialise l'escalier et les lignes violettes les portes détectées. A droite, en bas : graphe de l'environnement de test. En bleu foncé la topologie sans extrapolation, en vert les nœuds ajoutés par le processus d'extrapolation.*

à dire de cellules labellisées comme étant navigables et qui le sont effectivement et le nombre total de cellules labellisées comme libres, qui inclut les faux positifs (FP) : $P = VP/(VP + FP)$. Les performances globales et pour chaque classe sont présentées à la table 7.2 et illustrées pour le cas des objets quasi-statique à la figure 7.6.

Ces résultats montrent clairement que l'on atteint un haut niveau de précision en prédisant l'espace libre pour chacune des classes et pour les trois niveaux de confiance. Cela confirme l'hypothèse selon laquelle il existe un haut niveau de corrélations entre les informations sémantiques contenues dans les plus haut niveaux de la carte et celles dans les plus bas niveaux. Les performances affichées pour les trois différents seuils de confiance ne donnent pas d'indication claire sur l'évolution des performances, bien que l'on puisse s'attendre à une diminution des performances avec la diminution du seuil de confiance. Ceci peut s'expliquer par le fait qu'à mesure que s'étend la zone prédite avec la diminution de p_{th}, de nouveaux vrais positifs s'ajoutent en même temps que les faux positifs, ce qui maintient le résultat global inchangé. Cependant une raison pour ne pas diminuer encore le seuil et étendre ainsi la zone prédite tient en ce que les résultats deviennent moins significatifs avec la distance. L'espace libre prédit s'étend au delà des limites de la pièce et ne correspond plus dès lors, à l'espace accessible via l'instance de la classe identifiée.

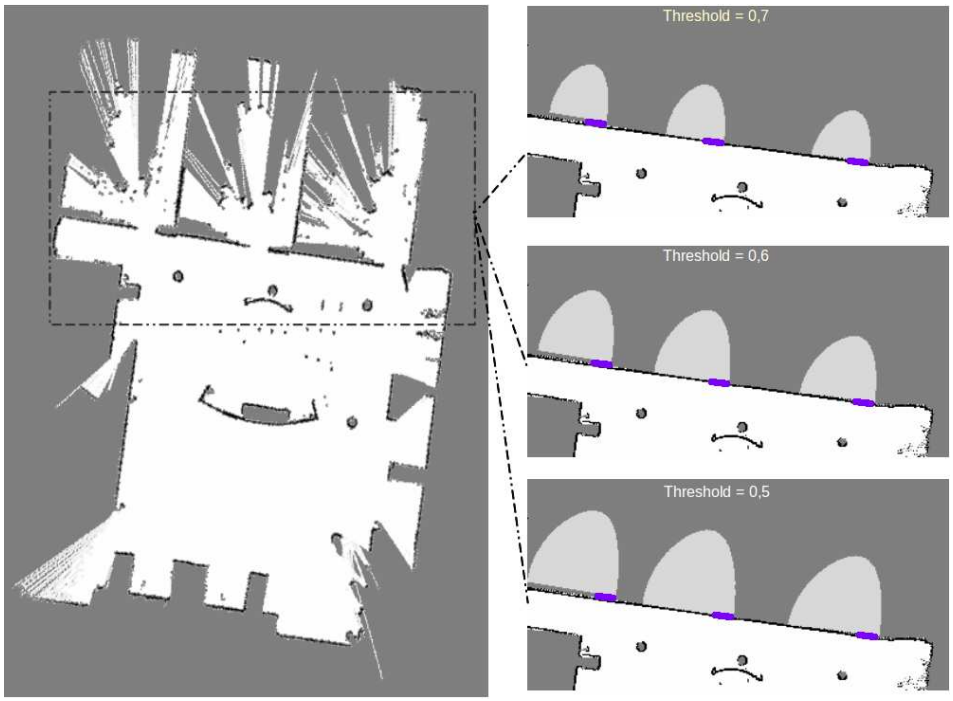


FIGURE 7.6 – *Extrapolation de l'espace libre à partir d'objets quasi-statiques. A gauche : vérité terrain réalisée avec les portes ouvertes. A droite : espace libre extrapolé suite à la détection de porte, matérialisées en violet. Les résultats sont présentés pour trois seuils de confiance différents, respectivement 0.7, 0.6 et 0.5 de haut en bas.*

Ceci souligne l'une des limitations de l'approche envisagée ici qui ne tient pas compte de la structure locale de l'environnement pour la prédiction. Cependant la carte générée demeure une véritable amélioration par rapport aux méthodes classiques.

Finalement, deux exemples qualitatifs additionnels sont présentés. La figure 7.7 montre un exemple d'extrapolation de carte réalisée en exploitant la présence d'un être humain qui sort du champ visuel en passant par une porte. Les images consécutives montrent les résultats de l'extrapolation de l'espace libre le long de la trajectoire suivie par l'être humain et son extrapolation à court terme après que celui-ci ait quitté le champ visuel. La zone grise matérialise l'espace libre correctement prédit derrière la porte, bien que celui-ci n'ai pas été observé. La figure montre un exemple à plus grande échelle où l'extrapolation de carte est produite à partir de la détection de portes. La partie gauche montre la carte réalisée avec les portes ouvertes et servant de vérité terrain. La partie gauche montre le résultat de la cartographie avec les portes closes et en gris clair les zones navigables prédites. Avec cette approche, la surface cartographiée est augmentée de 10% pour $p_{th} = 0.7$ et de 20% pour $p_{th} = 0.5$. Ces résultats peuvent potentiellement être augmentés selon la richesse des indices sémantiques et dynamiques contenus dans l'environnement. Ainsi, contrairement à la plupart des approches de cartographie, le dynamisme de la scène est utilisé comme un atout permettant d'élargir le champ de perception, plutôt que comme une contrainte le restreignant.

TABLE 7.2 – *Résultats de l'extrapolation de l'espace libre*

Classe	Seuil p_{th}		
	0.5	0.6	0.7
Statique	0.91	0.93	0.93
Quasi-Statique	0.98	0.99	0.99
Dynamique	0.95	0.94	0.93
Total	**0.95**	**0.95**	**0.95**

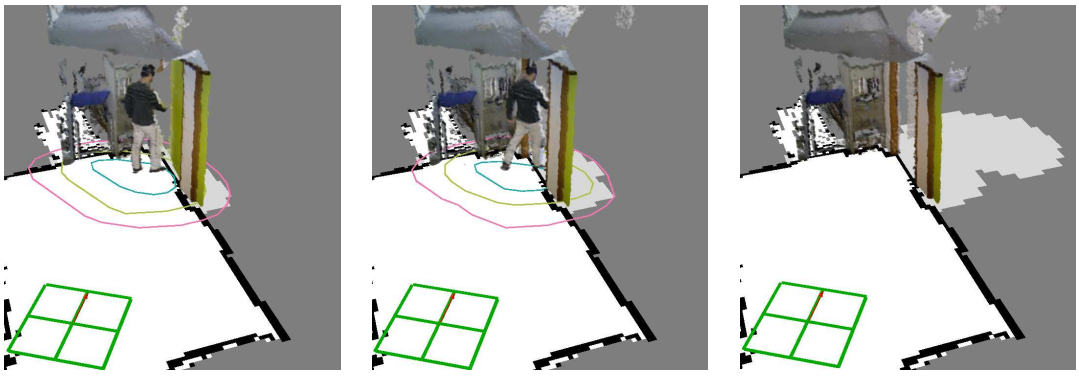


FIGURE 7.7 – *À gauche : illustration de l'espace libre attribué à une personne. À droite : trajectoire prédite lorsqu'une personne entre et sort du champ de vision du robot.*

7.5 Conclusion

Ce chapitre a permis de proposer une nouvelle approche de la cartographie basée sur l'analyse de l'information sémantique extraite de l'environnement. En exploitant le contexte et en utilisant le dynamisme de la scène comme un atout, une méthode permettant d'extrapoler l'aspect de la carte dans des zones non observées de l'environnement a été proposée. Les expériences réalisées dans un environnement réel ont permis de démontrer la capacité du système à découvrir l'espace libre et la topologie de l'environnement en dehors des zones visitées lors de l'acquisition avec de très bonnes performances. Au delà des résultats présentés, cette méthode permet d'envisager une nouvelle approche de la cartographie qui ne consiste plus uniquement à représenter des mesures ou leurs interprétations, mais aussi le résultat des prédictions que celles-ci permettent. En ce sens, la méthode d'extrapolation de carte proposée ici ouvre de nouvelles perspectives quant à l'utilisation de l'information sémantique pour la cartographie.

Conclusion

Cette partie a permis de présenter un nouveau modèle de carte conçu pour la navigation des robots. Il étend le modèle précédemment proposé par [Meilland 2011] en s'appuyant largement sur la compréhension de scène et l'analyse contextuelle pour offrir une description compacte, de haut niveau et évolutive de l'environnement qui dépasse les limites perceptuelles du robot. Cette carte est construite par une succession de processus, regroupés dans un schéma global à la figure 7.8. Les processus sont matérialisés par des flèches entourées de rouge lorsque l'information utilisée est locale, c'est à dire contenue uniquement dans la carte locale, bleue lorsqu'elle est globale et mauve lorsque qu'elle est les deux à la fois. Les contributions de cette thèse sont illustrées en vert et la partie préexistante sur laquelle elle s'appuie en bleu.

Ce modèle de carte permet d'étendre significativement les capacités d'un robot en autorisant la modélisation d'environnements complexes. Il permet la cartographie de grands environnements en répartissant l'information utile à la navigation dans des cartes locales (chapitre 3) pour lesquelles une description compacte de haut niveau est fournie par les graphes sémantiques (chapitre 5). Le référencement des cartes locales dans un arbre (chapitre 5) permet en outre de structurer la recherche d'information dans la carte qui sera utile pour la localisation (voir chapitre 8). Ce modèle est, de plus, bien adapté aux environnements dynamiques de par sa capacité à s'accommoder des occlusions. La mise à jour des données (chapitre 6) offre un moyen de reconstituer, avec de très bon résultats, l'apparence locale de la scène dans des zones où les occlusions causées par des objets dynamiques empêchent son observation et ce, à partir d'un faible nombre d'acquisitions. L'extrapolation de carte (chapitre 7) quant à elle, permet d'être robustes aux conditions d'observation en s'affranchissant partiellement des contraintes liés au point de vu et en exploitant le dynamisme de la scène comme un avantage plutôt qu'un inconvénient. Le processus de cartographie est par ailleurs rendu robuste aux erreurs de classification en exploitant le contexte spatial et temporel lors de l'annotation des images puis en détectant et supprimant une parti des erreurs restantes à plusieurs étapes du processus.

Ce modèle permet donc de cartographier de grands environnements dans un contexte opérationnel réaliste, c'est à dire sans envisager de multiples passages dans tous les endroits représentés par la carte ni une adaptation du comportement des autres agents qui s'y trouvent pour les besoins de l'acquisition. En ce sens, il s'intègre dans une démarche globale qui vise à adapter les robots à l'environnement quotidien en les dotant de capacités cognitives de plus en plus évoluées, nécessaires pour envisager leur réelle autonomie.

Outre les avantages qu'il confère par rapport à d'autres modèles, celui présenté ici démontre que bien que la classification d'image soit une étape perfectible, il est non seulement envisageable mais aussi profitable d'intégrer de l'information sémantique dans les modèles de carte, non pas uniquement du fait de l'incrément d'information qu'elle représente, mais

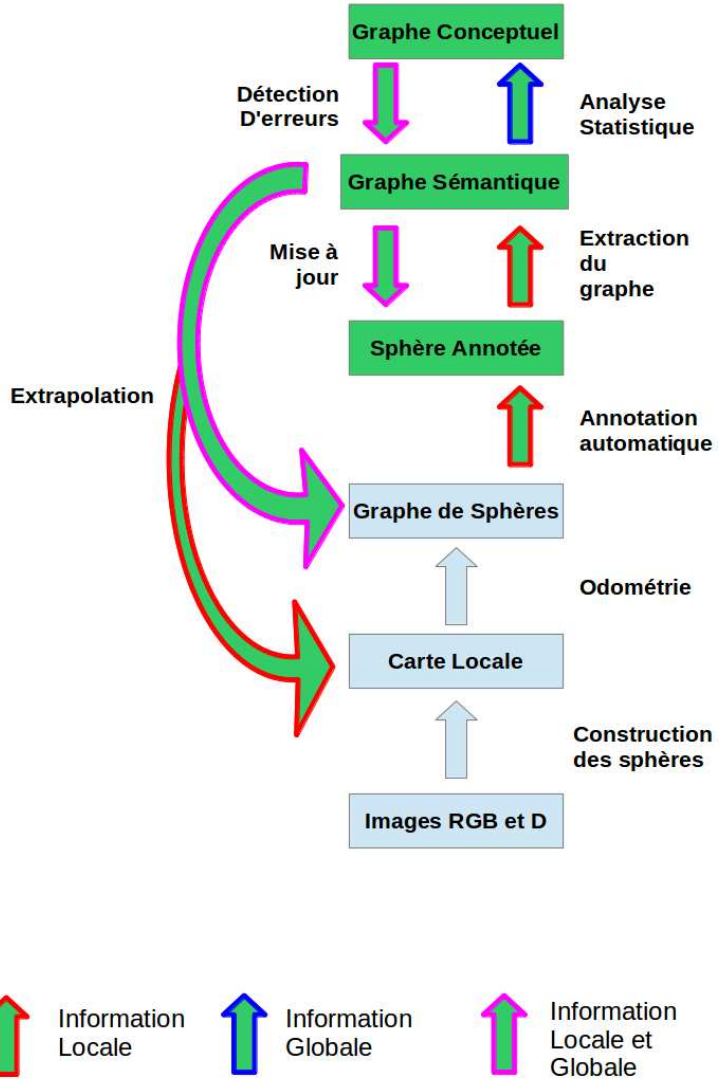


FIGURE 7.8 – *Processus de cartographie complet. L'information circule à la fois du haut vers le bas dans les différentes couches de la carte mais aussi horizontalement via l'utilisation de l'information contextuelle.*

du fait des raisonnements qu'elle permet. Ceux-ci permettent d'accroître la robustesse, la souplesse et la précision du modèle tout en facilitant son utilisation pour la navigation, comme il sera montré dans la prochaine partie.

Troisième partie

Navigation Sémantique

CHAPITRE 8

Localisation Sémantique

8.1 Introduction

La localisation est l'une des étapes les plus fondamentales des processus de navigation et de cartographie. Sa réussite conditionne l'ensemble des performances du robot. Elle est nécessaire à la planification et au suivi de trajectoire, à la fermeture de boucle ou encore à la recherche de contenu dans la carte. Ici, la localisation s'entend au sens large, c'est à dire qu'elle fait référence aussi bien au fait de trouver la position courante du robot que d'un objet quelconque. Si la localisation dans l'environnement immédiat est relativement triviale, la localisation en dehors du champ perceptif devient plus complexe. Il faut en effet disposer d'une carte de l'environnement et être capable de réaliser une recherche efficace à l'intérieur de celle-ci pour trouver la position associée au robot ou à l'objet recherché. Ceci est d'autant plus difficile que l'environnement devient grand. La recherche dans la carte peut alors s'avérer délicate du fait de la quantité importante de données à analyser.

La localisation basée vision a été étudiée depuis longtemps et l'utilisation d'images sphériques pour la navigation remonte au début des année 90 [Yagi 1995]. Il existe aujourd'hui des méthodes performantes permettant de retrouver une image dans des bases de données particulièrement vastes. La plupart des méthodes modernes s'appuient sur un processus de mise en correspondance d'images en trois phases. Premièrement des points d'intérêt sont extraits des images et les descripteurs associés sont calculés. Deuxièmement, les descripteurs de l'image de référence et l'image courante sont comparés et associés le cas échéant. Finalement un score est donné à la mise en correspondance des deux images. Le plus haut score correspond au meilleur appariement. Dans le modèle des sacs de mots, une structure d'arbre est souvent utilisée pour accélérer le processus de recherche et de comparaison des descripteurs extraits des images. Plusieurs variantes existent. On peut par exemple citer [Lowe 2004] qui utilise le contexte autour des points d'intérêt pour augmenter le pouvoir discriminant des descripteurs et améliorer leur mise en correspondance. L'idée est de sélectionner uniquement les bons candidats pour éviter l'aliasing perceptuel. D'autres méthodes récentes offrent de bonnes performances en utilisant d'autres stratégies. Par exemple, un réseau bayésien à structure arborescente est utilisé par [Cummins 2009] pour coder la co-occurrence des mots visuels dans les images. Un vocabulaire compact est construit par [Tardos 2012] en discrétisant l'espace des descripteurs binaires utilisés afin d'améliorer l'efficacité de la comparaison et de la recherche de similitudes entre les images.

Mais si elles offrent de très bonnes performances, compatibles avec la navigation, ces méthodes ne résolvent qu'une partie du problème de la localisation. En effet la description bas niveau n'est utilisable que pour trouver la localisation correspondant à la vue courante mais n'autorise pas la recherche au niveau des objets. De plus cette modélisation bas niveau fait que ces méthodes sont peu adaptées aux cartes sémantiques qui codent l'information à un haut niveau et qui visent justement à permettre des requêtes sémantiques.

D'autres méthodes de localisation basées cette fois sur l'information sémantique ont aussi été proposées. Une première manière d'utiliser l'information sémantique, consiste à estimer la position métrique du robot par rapport à des objets observés dans la scène. Plusieurs algorithmes de ce type ont été proposés, utilisant une approche bayésienne [Yi 2008, Dong 2013] et plus spécifiquement des filtres particulaires [Anati 2012]. Dans ce cas les objets sont attachés à une position spécifique, comme le seraient des points d'intérêt et la localisation se fait dans une carte métrique. Bien que la démarche d'utiliser des objets soit intéressante, la sémantique ne sert ici qu'à définir des amers visuels. Elle est utilisée en remplacement et pas en complément de l'information spatiale usuelle, ceci au sacrifice de la précision, la position des objets n'étant estimée qu'approximativement [Anati 2012]. La seconde manière d'utiliser la sémantique pour la localisation consiste à inférer la nature du lieu où se trouve le robot en fonction des objets observés. Il s'agit d'une localisation topologique puisque la position estimée correspond non pas à une pose métrique mais à un nœud dans un graphe. Cette stratégie repose sur l'hypothèse qu'une classe d'objets est attachée à un lieu particulier [Vasudevan 2008] ce qui n'est pas toujours garanti dans le monde réel. Par ailleurs, cela suppose que l'environnement peut être découpé en zones distinctes sans ambiguïté, ce qui est rarement le cas en extérieur. Enfin une autre approche a été proposée dans [Salas-Moreno 2013]. Elle repose sur l'idée de repositionner un graphe local d'objets dans un graphe global. Pour construire le graphe, non seulement la position 3D de l'objet doit être connue mais aussi son orientation. Cela suppose de disposer d'un modèle 3D complet de l'ensemble des objets et qu'ils aient tous une orientation clairement définie et observable. Dans un environnement réel, il est difficile de l'envisager, notamment du fait de la variance intra-classe qui rend impossible la définition de propriétés géométriques aussi simples et communes à toutes les instances d'une classe. Comment définir l'orientation d'un arbre ou d'un bâtiment ?

Les méthodes de localisation basées sur l'information sémantique proposées jusqu'à aujourd'hui sont donc mal adaptées aux environnements extérieurs ou de grande dimension et reposent le plus souvent sur des hypothèses qui sont autant de contraintes fortes qui limites leur déploiement dans un contexte réaliste. Dans la suite de ce chapitre est proposée une nouvelle stratégie de localisation qui s'appuie sur les graphes sémantiques et qui permet de localiser n'importe quel contenu de haut niveau dans la carte avec des performances au moins égales et souvent supérieures que les méthodes basées sac de mots.

8.2 Localisation dans un graphe de sphères

La localisation dans une carte métrique-topologique-sémantique composée de représentations sphériques locales intervient en deux étapes. Étant donné une requête R, correspondant soit à la perception courante de l'environnement, soit à un objectif à atteindre, le but de la localisation est de fournir la carte locale la plus semblable à R puis, le cas échéant, la position métrique de l'observation dans le référentiel de cette carte locale. Il s'agit donc d'un processus de localisation topologique suivit d'une estimation de la pose métrique dans le référentiel local. En notant $L(.)$ le processus de localisation, on a donc :

$$L(R) = (L_M \circ L_T)(R) \tag{8.1}$$

pour R une requête quelconque et où L_T correspond au processus de localisation topologique et L_M au processus de localisation métrique.

La seconde étape, L_M, est bien maîtrisée, notamment avec les méthodes de mise en correspondance dense d'images. Cependant, comme évoqué au chapitre 3, le choix de la sphère de référence consistant à rechercher celle dont la distance métrique est la plus faible est peu performante. Elle suppose d'avoir une première estimation de la position associée à la requête R, ce qui n'est pas toujours le cas, par exemple lors de l'initialisation de la localisation. De plus elle souffre des défauts de l'odométrie qui produit de la dérive et ne tient pas compte de la ressemblance des images. Deux images acquises de part et d'autre d'un mur sont proches dans l'espace métrique mais présentent sans doute peu de similarités visuelles.

Un processus de localisation topologique basé sur l'analyse du contenu sémantique des cartes locales a donc été proposé. Il s'appuie sur la modélisation de la scène locale sous forme de graphe sémantique. Soit $R \in \mathbb{O}_{\mathcal{G}_\mathcal{S}}$ une requête qui exprime la perception d'un ensemble d'objets et leur connexité sous forme d'un graphe sémantique. On définit une fonction $f(.)$ qui compare R au graphe sémantique $\mathcal{G}_{\mathcal{S}i}$ décrivant une carte locale :

$$\begin{array}{rccc} f: & \mathbb{O}_{\mathcal{G}_\mathcal{S}} \times \mathbb{O}_{\mathcal{G}_\mathcal{S}} & \rightarrow & [0,1] \\ & R, \mathcal{G}_{\mathcal{S}i} & \rightarrow & s_i \end{array} \tag{8.2}$$

où s_i correspond, pour chaque couple $(R, \mathcal{G}_{\mathcal{S}i})$, à la similitude des deux graphes exprimée par :

$$s_i(R, \mathcal{G}_{\mathcal{S}i}) = e^{1-\frac{k}{k_m}} \tag{8.3}$$

avec $k = min(|R|, |\mathcal{G}_{\mathcal{S}i}|)$ le cardinal de R où $\mathcal{G}_{\mathcal{S}i}$ et k_m le nombre de nœuds partagés par les graphes. Plus le nombre de nœuds partagés est grand plus leur similitude est élevée.

La localisation topologique est alors réalisée par une fonction L_T telle que :

$$L_T = \mathcal{S}_i : \max_{i \in N}(f(R, \mathcal{G}_{\mathcal{S}i})) \tag{8.4}$$

où $\mathcal{S}_i$ est la sphère choisie, N est le nombre de graphes sémantiques de la carte, c'est à dire le nombre de noeuds dans le graphe global $\mathcal{G}$.

Une requête exprimée par un graphe sémantique permet d'envisager, avec le même formalisme, la localisation à partir d'images traduites sous cette forme aussi bien que la localisation de contenus spécifiques décrit à haut niveau, ce que ne permettent pas les méthodes de localisation classiques bas niveau. Il est par exemple possible de formuler des requêtes du type "localiser un bâtiment avec des voitures à sa gauche et des arbres à sa droite", qui sera traduit par un graphe sémantique à trois nœuds, celui correspondant au bâtiment étant connecté aux nœuds "voiture" et "arbre".

8.2.1 Comparaison de graphes sémantiques

Pour réaliser la localisation topologique, il est nécessaire de pouvoir estimer la ressemblance de deux graphes sémantiques : celui correspondant à la requête et celui de la sphère de référence. Pour cela on souhaite réaliser un appariement univoque, c'est à dire que chaque nœud d'un graphe soit appairé ou non à un seul nœud de l'autre graphe avec une correspondance des arêtes. Ce problème d'appariement de deux graphes est NP-difficile. Cependant il est possible de réduire la complexité combinatoire du problème de mise en correspondance de deux graphes en utilisant des contraintes pour l'appariement des nœuds et des arêtes. Elles sont de deux ordres :

- Les contraintes unaires, qui s'appliquent sur une paire de nœuds à associer. Elles restreignent les possibilités de mise en correspondance entre les nœuds en fonction de leurs propriétés intrinsèques.
- Les contraintes binaires, dépendant du voisinage des nœuds et qui restreignent les possibilités d'appariement des nœuds en fonction de leurs voisinages.

Les possibilités d'appariements sont fortement réduites par l'utilisation de ces contraintes et la meilleure mise en correspondance peut être trouvée efficacement. L'algorithme utilisé, illustré à la figure 8.1, est dérivé de la recherche basée sur un *arbre d'interprétation*. Il est présenté à l'algorithme 1 et détaillé ci-dessous.

Soient $\mathcal{G}_{\mathcal{S}1}$ et $\mathcal{G}_{\mathcal{S}2}$ deux graphes à comparer. Pour chaque nœud de $\mathcal{G}_{\mathcal{S}1}$ on cherche parmi ceux de $\mathcal{G}_{\mathcal{S}2}$ des candidats potentiels avec les contraintes unaires suivantes :

- le label doit être identique, l
- la variation d'aspect, notée ν_r et exprimée par le ratio du petit axe sur le grand axe de l'ellipse englobante associée au nœud, doit être inférieure à un seuil
- la variation de l'angle de l'ellipse associée, noté ν_α, et qui doit être faible

Une liste de candidats potentiels est alors constituée. Ces candidats sont triés du plus ressemblant au moins ressemblant en fonction de la valeur de ν_r. Puis pour chacun d'eux, des contraintes binaires avec le voisinage sont utilisées pour déterminer si le candidat correspond ou non au nœud :

- Le nombre de voisins de chaque nœud doit être équivalent
- Le label des voisins doit être équivalent

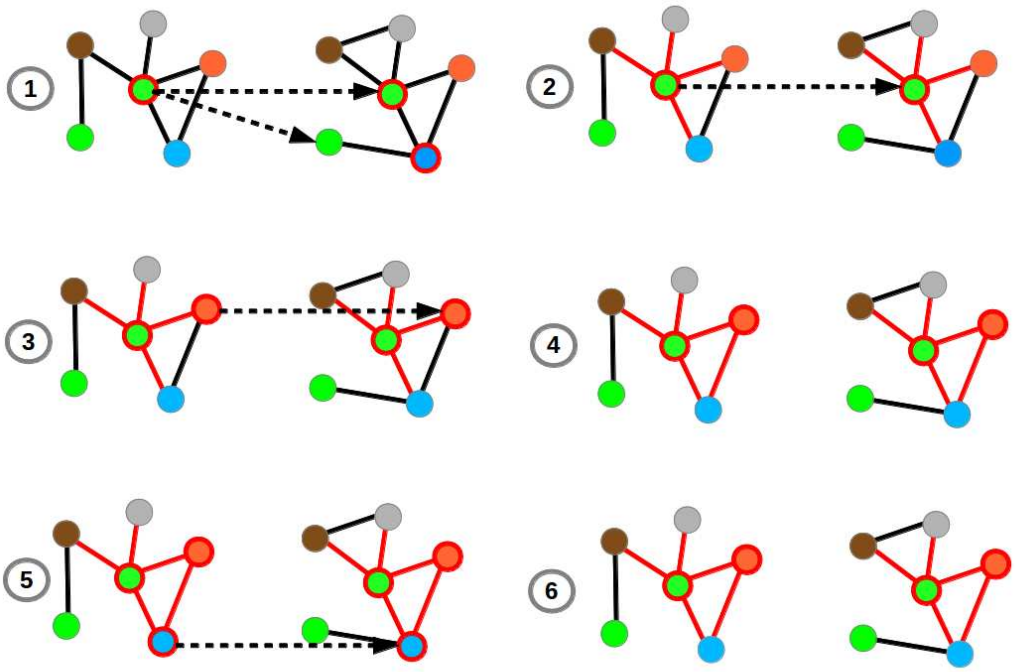


FIGURE 8.1 – *Illustration du processus de mise en correspondance de deux graphes : (1) un nœud du label vert du premier graphe est associé à deux candidats du second graphe dont les propriétés intrinsèques sont similaires, (2) l'analyse du voisinage révèle qu'un des deux candidats est compatible, (3) un nœud voisin est associé avec un nœud du second graphe (4) dont le voisinage révèle qu'il est compatible. Un troisième nœud du premier graphe est associé à un nœud du second graphe (5) mais le voisinage démontre qu'ils sont différents (6). Finalement, seul deux nœuds sur 6 sont mis en correspondance*

8.2.2 Accélération du processus

La comparaison des graphes peut être faite efficacement avec l'algorithme 2. Cependant lorsque la carte devient grande le nombre de graphes à comparer croit significativement. Pour accélérer le processus de localisation, l'index qui référence les cartes locales est utilisé (voir chapitre 5). Étant donné une requête R, la branche contenant les mêmes classes est sélectionnée au premier niveau de l'arbre puis celle contenant le même nombre d'instances de chacune des classes. La feuille terminale donne une ou un petit ensemble de cartes locales qui correspondent potentiellement à la recherche. La comparaison des graphes n'est effectuée que pour ce sous-ensemble de cartes ce qui permet d'accélérer significativement le processus de localisation.

Algorithme 2 Algorithme de comparaison de deux graphes sémantiques basé sur un arbre d'interprétation

ENTRÉES : $\mathcal{G}_1$, $\mathcal{G}_2$:graphes sémantiques extraits de l'image courante et de l'image de référence
SORTIES : Score de la mise en correspondance des deux graphes (liste des nœuds mis en correspondance)
pour tout Noeud $n_i \in \mathcal{G}_1$ **faire**
  pour tout Nœud $n_j \in \mathcal{G}_2$ **faire**
    si ContrainteUnaire(n_i, n_j) == Vrai **alors**
      Ajouter (n_i, n_j) à *ListNoeudsAppariés*
    finsi
  fin pour
  si ListNoeudsAppariés ≥ 1 **alors**
    Trier *ListNoeudsAppariés*
    pour tout (n_i, n_j) *in ListNoeudsAppariés* **faire**
      Ajouter (n_i, n_j) à *InterpList*
      si ContrainteBinaire(*InterpList*) == Faux **alors**
        Retirer (n_i, n_j) de *InterpList*
      finsi
    fin pour
  finsi
fin pour

8.2.3 Requête de haut niveau

Dans le cas de requêtes autres que des images, qui peuvent être formulées par un système artificiel ou une personne, la méthode de localisation est ajustée pour accepter les requêtes qui décrivent partiellement la scène ou auxquelles correspondent plusieurs positions. Premièrement, la recherche dans l'arbre d'indexation inclue comme résultats *toutes* les branches contenant *au moins* les nœuds de la requête et pas uniquement celles contenant *exactement* le même nombre d'instances. Ceci permet de fournir plusieurs réponses et autorise donc les requêtes de type "localiser tous les endroits où se trouvent des instances de la classe X". Deuxièmement, la comparaison des graphes n'utilise que les informations disponibles. Par exemple si la dimension d'une zone sémantique n'est pas donnée, le test de dimension est considéré comme positif. De cette manière l'algorithme produit toutes les réponses correspondant potentiellement à la requête. Cet ajustement se justifie par le désir d'une plus grande souplesse dans la formulation des requêtes d'origine humaine.

8.3 Résultats

Le processus de localisation topologique proposé a été évalué sur la base de données acquise sur le campus de l'INRIA Méditerranée. Les résultats sont comparés à ceux obtenus avec une méthode récente utilisant les sac-de-mots [Tardos 2012] dont on utilise l'implémentation disponible ici [1]. Dans cet approche la similarité entre l'image courante et l'image de référence est évaluée en dénombrant les descripteurs partagés par les deux images. L'algorithme construit un arbre qui discrétise l'espace des descripteurs de façon à accélérer la

1. http ://webdiis.unizar.es/ dorian/index.php ?p=3

mise en correspondance de ceux-ci. Le nombre de mots visuels qui peuvent être codés par la structure dépend de deux facteurs : le nombre de branches possibles partant de chaque nœud, noté K et la profondeur de l'arbre, notée L. Le nombre de mots visuels est donné par le nombre de feuilles, à savoir K^L. Plus le nombre de mots visuels est grand plus la description de l'image sera précise et le risque d'aliasing perceptuel faible. Cependant le temps nécessaire à la comparaison d'image croit lui aussi avec le nombre de mots visuel. Deux couples de valeurs sont utilisés, $(K = 10, L = 5)$ produisant 100000 mots et $(K = 8, L = 4)$ produisant 4096 mots (pour plus de détails concernant les paramètres, se reporter à l'article de référence [Tardos 2012]).

Mesures Selon les cas, quatre mesures sont rapportées : le temps de récupération de la position noté t_r, la précision p_r, la matrice de similarité $\mathcal{M}_d$ et le pouvoir discriminant $\mathcal{D}$. La précision donne la capacité de l'algorithme a retrouver la position correspondant à la requête. La matrice de similarité donne, pour chaque requête, la similitude avec l'ensemble des images de la base de données. Le pouvoir discriminant mesure la capacité à distinguer les images correspondant à une requête, des autres images de la base de donnée. Pour chaque requête, la similitude moyenne avec les images correspondantes, notée S_{vrai} et celle avec les images ne correspondant pas, notée S_{faux} sont calculées. Le pouvoir discriminant est défini par $\mathcal{D} = \frac{S_{vrai}}{S_{faux}}$. Il est d'autant plus grand que l'algorithme est capable de faire la différence entre le lieux requis et le reste de la carte.

Scenarios Cinq scénarios sont proposés pour démontrer l'efficacité, la robustesse et la souplesse du processus de localisation proposé. Dans le premier, la base de données est constituée de l'ensemble des images de référence et on évalue la capacité de l'algorithme à retrouver efficacement une image dans cette base. La précision et le temps d'exécution sont rapportés. Dans le second scénario, on évalue la robustesse de l'algorithme aux changements de points de vue en demandant de trouver la position correspondant à des images qui ne sont pas contenues dans la base de données mais proches. Il peut s'agir d'images prises à des positions différentes des images de référence ou d'images monoculaires. $\mathcal{M}_d$ et $\mathcal{D}$ sont rapportées. Deux stratégies pour la localisation en environnement dynamique sont proposées dans les scénarii 3 et 4. Notamment, le scénario 3 montre l'amélioration des performances apportées par la mise à jour de la carte présentée au chapitre 6. Enfin, le cinquième scénario permet de démontrer les capacités de l'algorithme proposé à traiter des requêtes de haut niveau, ce qui est impossible avec des méthodes bas niveaux.

8.3.1 Scénario 1 : récupération d'images

Dans ce scénario on évalue la capacité de l'algorithme à retrouver des images appartenant à la base de données. Celle-ci est composée de l'ensemble des images de référence acquises sur le campus de l'INRIA. Pour cela, chacune des images est fournie à l'algo-

TABLE 8.1 – *Temps de récupération moyen d'une image dans la base de données et précision de l'algorithme*

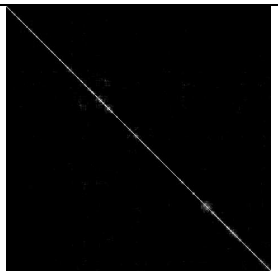
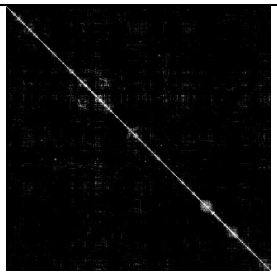
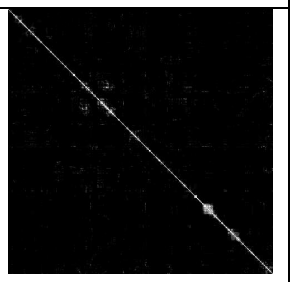
Algorithme	t_r	p_r
BoW K=10, L=5	22ms	1
BoW K=8, L=4	16ms	1
Graph	8.4ms	1
Graph+Index	0.12ms	1
Index	**54**μs	88%

rithme qui renvoie l'image la plus proche dans la base de donnée. Les résultats sont rapportés à la table 8.1 pour cinq algorithmes : les méthodes basées sac-de-mots avec 100000 ou 4096 mots, référencées par le terme "BoW", la comparaison des graphes noté "Graph", la comparaison des graphes précédée de la recherche dans l'arbre d'indexation des sphères en fonction de leur contenu sémantique, notée "Graph+Index", l'utilisation de l'arbre d'indexation seul, noté "Index".

Les temps indiqués ne prennent pas en compte l'extraction des représentations des images, que ce soit la construction des graphes sémantiques ou l'extraction des sac-de-mots, ceci parce que l'algorithme de localisation proposé peut s'accommoder de requêtes formulées autrement que sous forme d'images. Par exemple, le robot peut rechercher un ensemble d'objets décrit directement par un graphe sémantique, le temps d'extraction de la représentation est alors nul. Dans ce cadre, on constate que l'algorithme de recherche le plus rapide est celui utilisant l'arbre d'indexation des sphères. La localisation est en effet extrêmement rapide puisqu'il s'agit simplement de dénombrer les nœuds d'un graphe. Cependant la précision est inférieure à 1 ce qui signifie que l'algorithme n'est pas toujours en mesure de fournir une réponse correcte, souvent parce que celle-ci n'est pas unique, ce qui n'est pas souhaitable. L'utilisation de la comparaison des graphes en conjonction avec l'arbre d'indexation fournit les meilleurs résultats avec un temps de récupération plus de dix fois inférieur à la méthode basée sac-de-mots utilisée ici. Ceci vient du fait que le nombre de zones sémantiques dans une image est relativement faible, de l'ordre de 20 et que leur comparaison est très rapide puisque seulement 3 valeurs sont nécessaires (label, ν_r, ν_α) alors que la comparaison de descripteurs est souvent plus complexe. De plus l'utilisation de l'arbre d'indexation réduit drastiquement le nombre de comparaisons à effectuer avec habituellement quelques dizaines d'images dans les feuilles de l'arbre d'indexation. La méthode de localisation topologique proposée est donc d'une grande efficacité.

Dans le cas où la localisation est faite à partir d'une image, il faut ajouter le temps de labelisation de l'image qui est de 6.8 secondes pour la base de données INRIA (voir chapitre 4). Dans ce cas, l'extraction de sac-de-mots est significativement plus rapide. Cependant, en diminuant la taille des images utilisées pour la localisation on peut considérablement accélérer le processus de labellisation sans dégradation significatives des performances car

TABLE 8.2 – *Performance de récupération d'une image dans la base de données en fonction de la résolution des images*

Résolution	512x167	341x111	256x83
Labelisation	1.7s	0.73s	0.39s
$\mathcal{M}_d$			
Précision	1	1	0.98

le temps de labellisation est approximativement linéaire par rapport au nombre de pixels. Le tableau 8.2 donne le temps approximatif nécessaires à la labelisation pour différentes résolution d'images ainsi que la matrice de distance et la précision. On remarque que pour une résolution de seulement 341x111 pixels, la précision de la localisation reste maximale pour un temps de labelisation inférieur à la seconde. En parallélisant le code utilisé pour la labelisation, qui est mono-threadé, il est possible de diminuer significativement ce temps. Notamment, les étapes d'extraction des descripteurs et d'utilisation de Random Forest sont facilement parallélisable. L'algorithme proposé, bien que reposant sur le processus complexe d'analyse de scène, présente donc des performances en terme de temps d'exécution qui s'approchent de celles des méthodes BoW pour la localisation à partir d'images et qui les dépassent dans le cas d'une requête formulée directement sous forme de graphe et de sac-de-mots visuels.

8.3.2 Scénario 2 : Robustesse au changement de point de vue

Dans ce scénario, la base de données Bd est constituée d'une image sur quarante de la séquence d'acquisition, ce qui représente environ 250 images (voir figure 8.2). Une première base de requête $Br1$ est créée à partir des images restantes en prenant une image sur quarante mais décalée de vingt images par rapport à Bd, de telle façon que les images de $Br1$ et Bd correspondent à des points de vue légèrement différents. Une seconde base de requête $Br2$ est construite à partir des images monoculaires utilisées pour construire les images sphériques. A chaque élément de Bri correspondent quatre éléments de Bd qui sont les deux images précédentes et les deux images suivantes dans la séquence d'acquisition. L'objectif est d'évaluer comment l'algorithme s'accommode des changements de point de vue dûs à la distance ($Br1$) ou à la projection ($Br2$).

Les résultats sont présentés à la figure 8.3. La vérité terrain correspond à une image

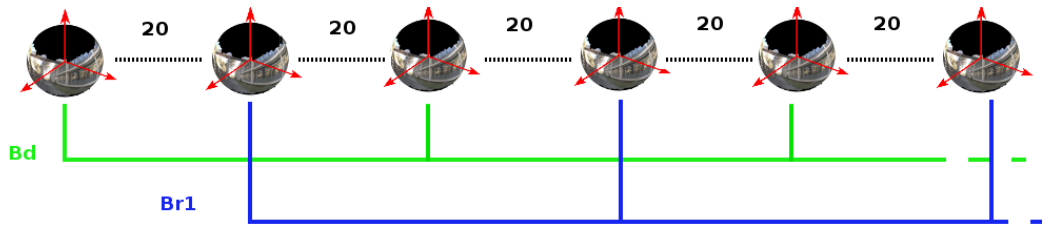


FIGURE 8.2 – *Construction de la base de données et de la base de requête 1. Une image sur 40 est sélectionnée avec un décalage de 20 images entre les deux bases.*

noire dont la diagonale est blanche. Cependant il peut subsister une seconde diagonale, perpendiculaire à la première, qui correspond au retour du véhicule par le même chemin. Les deux approches, "BoW" et "Graphe", donnent une précision de 1 pour la localisation des images des deux bases $Br1$ et $Br2$. Les matrices $\mathcal{M}_d$ montrent que l'approche basée BoW a tendance à trouver plus de similitudes avec des images distantes de la position recherchée que la méthode basée sur les graphes. Ceci vient du fait que les mêmes mots visuels peuvent être présents à différents endroits de la scène ou détectés sur de relativement longues distances. A l'inverse les graphes sémantiques encodent un point de vue particulier de la scène et ne sont donc valables que localement. Dans le cas de la base de requête $Br2$, à savoir les images monoculaires, la seconde diagonale qui correspond au retour du véhicule par le même chemin, n'est plus visible avec la méthode des graphes. Ceci vient essentiellement de la modification de la forme des zones sémantiques à l'aller et au retour du véhicule qui, du fait de la réduction du champ de vision, rendent l'appariement de graphe plus difficile.

Les deux méthodes s'accommodent donc très bien de changements de point de vue relativement faibles puisqu'elles donnent une précision parfaite, mais l'approche sac-de-mots est plus robuste pour des changements importants. Cependant dans le cadre d'une carte composée de sphères égo-centrées, la localisation métrique est réalisée par une mise en correspondance dense d'une image de référence avec l'image courante. Pour cela les deux images doivent être proches. Il est donc préférable d'utiliser un algorithme qui privilégie fortement les images de référence dont le contenu est proche de l'image courante, ce que fait la méthode basée sur les graphes. En effet le pouvoir discriminant $\mathcal{D}$ de cette méthode est 3 à 4 fois plus important que celui des méthodes basées sac-de-mots. Cela signifie que la confiance dans la localisation obtenue est plus élevée avec la méthode des graphes. Ceci provient du fait que les graphes reposent sur la structure entière de l'image et que l'information qu'ils encodent est très caractéristique d'un endroit particulier. Ils sont donc moins sujet à l'aliasing perceptuel que les méthodes BoW.

Les deux approches sont de nature fondamentalement différente. Dans le cas des sacs-de-mots, l'appariement entre les descripteurs locaux est soumis à des contraintes fortes et le contexte est secondaire. A l'inverse, dans le cas des graphes, l'appariement des nœuds

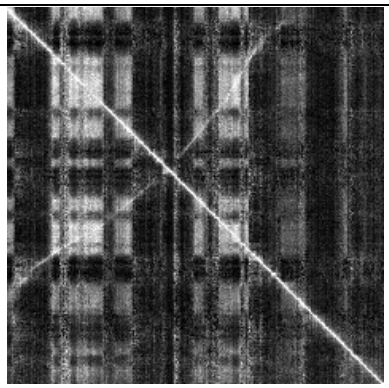
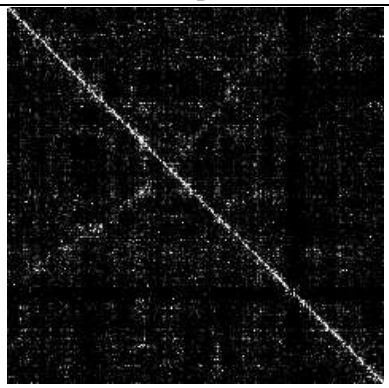
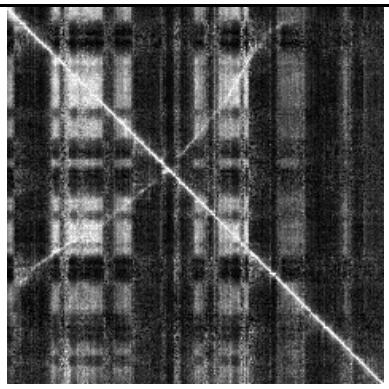
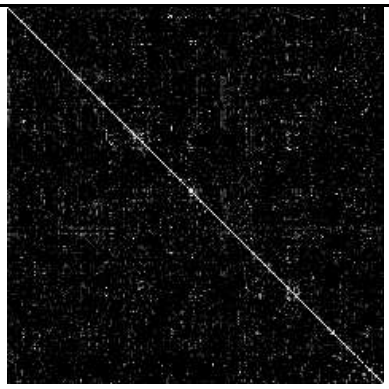
Base	BoW K=10, L=5	Graphe
$Br1$		
	5.68	17.38
$Br2$		
	2.99	13.37

FIGURE 8.3 – *Résultats de la localisation à partir d'images distantes ($Br1$) ou monoculaires ($Br2$). Les matrices de similarité sont données pour les deux algorithmes ainsi que le pouvoir discriminant $\mathcal{D}$.*

deux à deux est plus souple mais le voisinage exerce de fortes contraintes pour la mise en correspondance des graphes. Les méthodes BoW ont tendance à caractériser un lieu, tandis que les graphes sémantiques caractérisent un point de vue. Les premières sont donc plutôt adaptées aux cartes référencées monde tandis que la méthode proposée ici est plus adaptée à notre modèle de carte.

8.3.3 Scénario 3 : Localisation après mise à jour

Dans ce scénario, deux cas sont envisagés. Dans le premier, la base de données Bd est constituée d'images acquises sur le campus de l'INRIA et la base de requête Br, d'images acquises au même endroit mais trois années plus tard. Dans le second cas, les bases de données et de requête sont composées de la même manière que précédemment mais mises à jour par le processus présenté au chapitre 6. Le but est d'évaluer le gain potentiel que procure la mise à jour de la carte pour la localisation. Les résultats sont comparés à ceux

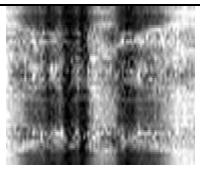
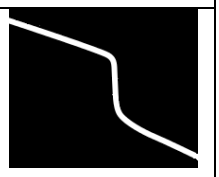
Mesure	BoW K=10, L=5	Graphe	Graphe+Màj	Vérité terrain
$\mathcal{M}_d$				
$P_r(3)$	0.005	0.44	0.55	-
$P_r(5)$	0.01	0.52	0.64	-

FIGURE 8.4 – *Comparaison des résultats pour les méthodes basées sac-de-mots (BoW) et basées Graphes Sémantiques sans (Graph) ou avec(Graph+Màj) mise à jour des données. La seconde montre de bien meilleurs performances qui s'expliquent entre autre par la robustesse de la classification.*

obtenus avec la méthode basée sac-de-mots présentée plus haut. Deux mesures sont rapportées, les matrices de similitude et la précision. Lors du calcul de cette dernière il faut définir une vérité terrain correspondant aux images les plus proches de l'image recherchée. Puisqu'il n'y a pas de correspondance directe entre les images de Bd et de Br, il faut définir un voisinage dans la base de données correspondant à chaque image de la base de requête. La précision est donnée pour des voisinages de 3 et 5 images. La précision est alors d'autant meilleures que l'algorithme considère l'une de ces images comme la plus proche de l'image courante. Les résultats sont présentés à la figure 8.4. La vérité terrain n'est cette fois plus une diagonale du fait de la différence de vitesse entre les véhicules pendant les deux acquisitions.

Les matrices de similitude montrent que les méthodes BoW sont très peu performantes dans ce cas. La trajectoire réelle du véhicule n'est pas identifiable avec ces méthodes et la précision est quasi nulle. Ceci provient sans doute des changements d'intensité lumineuse entre les deux bases qui ont été acquises dans des conditions différentes. Les modifications de l'aspect local des images rendent difficiles l'appariement des descripteurs qui ne correspondent plus d'une base à l'autre. Dans une moindre mesure, le déplacement des objets peut aussi participer à dégrader les performances de l'algorithme en modifiant la position des descripteurs, en occultant certains et en ajoutant de nouveaux. Les algorithmes de classification sont beaucoup plus robustes aux variations des conditions d'éclairement puisqu'ils gèrent, par construction, la variance intra-classe et donc le changement d'aspect des descripteurs. La classification des pixels est donc moins affectée par des changements modérés de luminosité et les résultats de la méthode basée sur les graphes sémantiques sont bien meilleurs. Mais les tentatives de relocalisation avec les images non mises à jour montrent qu'il subsiste des erreurs avec notamment des zones identifiées à la figure 8.5 par des flèches rouges, où l'algorithme discrimine difficilement l'image correspondant à la position réelle de celles correspondant à des positions distantes. La mise à jour des données permet d'augmenter significativement le pouvoir discriminant de l'algorithme et la trajec-

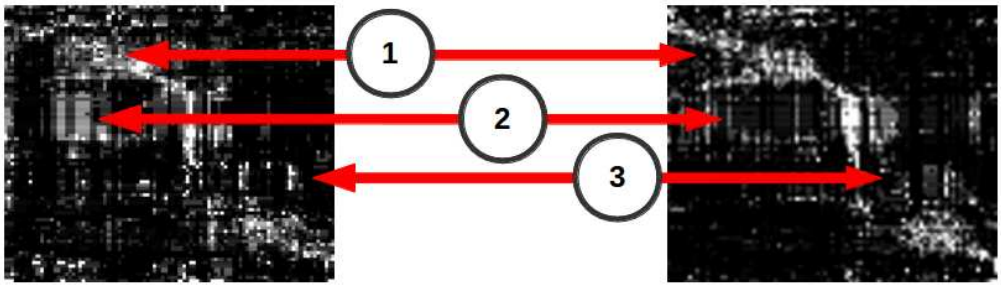


FIGURE 8.5 – *Amélioration des performances de localisation due à la mise à jour des données. A gauche : matrice de similitude avant mise à jour. A droite : matrice de similitude après mise à jour. En (1) le véhicule effectue une marche arrière. La position est mieux estimée localement avec la trajectoire effective bien visible à droite. En (2) et (3) les "colonnes blanches" qui apparaissent à gauche sont dues à la ressemblance entre la position courante et des positions distantes que l'algorithme n'arrive pas à discriminer. Après mise à jour, elles sont considérablement réduites.*

toire réelle est mieux estimée dans ce cas. Par ailleurs, la précision de la localisation est meilleure avec la carte mise à jour, de 12% avec un voisinage de 5 images et de 9% avec un voisinage de 3 images. Bien qu'imparfait, ces résultats montrent une fois de plus les avantages d'utiliser l'approche basée sur les graphes sémantiques par rapport aux méthodes bas niveau.

8.3.4 Scénario 4 : Localisation sous contrainte

Un second mécanisme de localisation en environnement dynamique peut être proposé en s'appuyant sur l'information sémantique et le formalisme des graphes. Il consiste à n'utiliser que les parties stables de l'environnement pour la recherche des similitudes entre les images. Plus précisément, la méthode consiste à ajouter des contraintes pour la comparaison des graphes, qui imposent de ne pas tenir compte de certains nœuds en fonction de leur label. Ainsi l'analyse des similitudes entre deux images ne se fait qu'avec des graphes réduits. Pour évaluer la qualité de cette approche, le même schéma d'expérience que pour le scénario 3 a été utilisé mais au lieu de mettre à jour les données, la contrainte de ne pas tenir compte des nœuds labellisées comme "voiture" est utilisée.

Les résultats sont présentés à la figure 8.6. On constate là encore une nette amélioration des performances, avec un accroissement de la précision de 10% pour l'utilisation d'un voisinage de 3 images et 9% pour un voisinage de 5 images. La matrice de similitude montre que ce procédé est plus discriminant (plus de contraste entre les bons et les mauvais pixels, et donc entre les bonnes et les mauvaises positions) que la méthode précédente mais qu'il y a aussi plus d'erreurs (pixels blancs en dehors de la zone de vérité terrain). Ce comportement peut s'expliquer de la façon suivante. Dans un certain nombre de cas les objets dynamiques occupent une petite portion de l'image. Le simple fait de la négliger

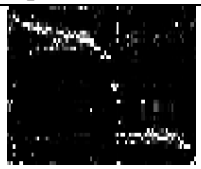
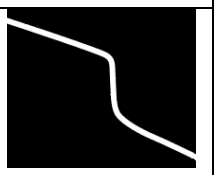
Mesure	Graphe	Graphe+Contraintes	Vérité terrain
$\mathcal{M}_d$			
$P_r(3)$	0.44	0.54	-
$P_r(5)$	0.52	0.61	-

FIGURE 8.6 – *Matrice de similitude et précision pour la relocalisation sous contrainte. Ici la contrainte utilisée est de ne pas tenir compte de la classe voiture. Les résultats sont sensiblement meilleurs avec la contrainte que sans.*

Requête	t_r
2 immeubles	$56\mu s$
Pas de voiture	$55\mu s$
Voiture sur la route	0.95ms
Arbre à coté d'un bâtiment	0.86ms

TABLE 8.3 – *Temps de récupération pour des requêtes de haut niveau*

permet donc de se localiser sans difficulté. Il n'est alors pas nécessaire de mettre à jour les images. Mais lorsque la portion de l'image occupée par des objets dynamiques devient plus importante, la mise en correspondance des nœuds restants est difficile sans correction de leur forme. Le nombre de nœuds partagés entre deux graphes chute et la localisation devient erratique.

Cette méthode permet d'envisager une relocalisation dans une base de données pour laquelle la mise à jour n'est pas possible, faute d'observations. A défaut d'atteindre une précision de 100%, elle offre un moyen de gérer les aspects dynamique de la scène là où les méthodes bas niveau ne le permettent tout simplement pas. Elle offre un moyen naturelle de détecter les objets dynamique lorsque les approches classiques en sont incapables, faute d'un nombre d'observation suffisant.

8.3.5 Scénario 5 : Requêtes complexes

Le dernier scénario est consacré à la localisation basée sur des requêtes de haut niveau telles qu'elles pourraient être formulées par l'utilisateur humain. La base de données est composée de l'ensemble des images sphériques. Chaque requête est présentée sous la forme d'un graphe sémantique et la recherche s'effectue comme expliqué au paragraphe 8.2.3. Pour chaque requête, le temps de récupération est donné à la table 8.3.

Avec la méthode de localisation proposée, il est extrêmement rapide retrouver un sous ensemble de cartes locales correspondant à une requête simple (liste de classes présentes)

ou plus complexe (classes plus relations entre elles). L'utilisation de l'arbre d'indexation permet d'accélérer considérablement les requêtes les plus simples puisque la comparaison de graphes n'est même pas nécessaire. Dans tous les cas le temps de récupération est faible ce qui permet d'envisager d'effectuer ce type de requête en ligne. Ceci démontre un autre avantage de l'algorithme de localisation topologique proposé qui permet d'envisager de nouvelles applications.

8.4 Conclusion

La méthode de localisation proposée, qui s'appuie largement sur le formalisme des graphes sémantiques, permet donc d'atteindre, voir de dépasser, les méthodes de localisation classiques dédiées aux grands environnements tout ne proposant de résoudre de nouveaux problèmes comme la localisation de contenus sémantiques. Si sa robustesse pour des changements de points de vue importants est moins élevée que celle des méthodes basées sac-de-mots, ce qui est un inconvénient mineur dans notre cas comme expliqué plus haut, elle est en revanche plus robustes aux changements des conditions d'illumination et au dynamisme de la scène. L'utilisation d'imagettes de faible résolution permet par ailleurs d'atteindre des temps de localisation à partir d'images du même ordre de grandeur que les méthodes basées sac-de-mots. En outre, elle permet d'envisager la localisation sous un nouvel angle en autorisant des requêtes complexes, telles qu'elles pourraient être formulées par l'être humain, et ceci en ligne. Ce dernier point est particulièrement intéressant pour la planification de trajectoire puisqu'il permet d'envisager la définition et la recherche de position à atteindre directement à un haut niveau. L'objet du prochain chapitre est précisément d'étudier comment l'information sémantique, particulièrement lorsqu'elle est codée sous forme de graphes sémantiques, permet la navigation intelligente des robots en s'appuyant sur les processus de localisation développés ici.

CHAPITRE 9

Planification de chemin basée sémantique

9.1 Introduction

La navigation autonome dans un environnement réel est l'un des plus grand défis à relever pour un robot mobile. Outre le fait de devoir construire sa propre représentation du monde et de se localiser, le robot doit aussi être en mesure de choisir le meilleur chemin pour atteindre une position donnée. Le choix de ce chemin est soumis à des contraintes qui dépendent à la fois des capacités du robot et de l'environnement. Dans le cadre de très grands environnements, on peut distinguer la planification de trajectoire locale, qui prend en compte les obstacles immédiatement observables par le robot pour définir une trajectoire dans le référentiel courant et la planification de chemin dont l'objectif est de définir une route à grande échelle, sous forme de points de passage par exemple. Dans le cas d'une carte de sphères augmentées, la première de ces tâches est réalisée au niveau local avec l'information contenue dans une sphère. La seconde est, quant à elle, réalisée au niveau du graphe global et consiste à choisir les sphères de référence qui permettront la navigation depuis la position courante jusqu'à l'objectif. Si la navigation locale est étudiée depuis des années et est aujourd'hui bien maîtrisée, la planification de chemin reste aujourd'hui perfectible. En effet la plupart des stratégies proposées considèrent le meilleur chemin comme étant celui reliant la position courante à la position cible suivant le parcours le moins long. Or, s'il est évidement intéressant de réduire la longueur du chemin entre deux positions, bien d'autres contraintes doivent être pris en compte dans un scénario de déploiement réaliste. Pour n'en citer que quelques unes, un robot évoluant dans un environnement réel est amené à rencontrer des objets dynamiques dont le comportement est rarement prévisible. Un agent intelligent peut légitimement accorder moins d'importance à un chemin sur lequel le risque de rencontrer de tels objets est plus élevé qu'ailleurs, fut-il plus long. D'autres contraintes comme la minimisation du risque de collision ou la stabilité des amers visuels peuvent être des critères utiles à prendre en compte dans la définition d'un chemin optimal, qui ne sera pas nécessairement le plus court.

L'information sémantique s'est déjà montrée très utile pour améliorer la planification de tâches,notamment lorsque l'espace de recherche est grand, à travers sa capacité à transposer le problème dans un espace plus petit [Galindo 2008]. Dans le cas spécifique de

la navigation, la sémantique a été utilisée principalement pour la modélisation de lieux, permettant de spécifier des objectifs à haut niveau pour la navigation du robot. Par exemple [Borkowski 2010] utilise la sémantique pour modéliser l'environnement comme un graphe de lieux distincts. Le robot est alors capable de raisonner sur ce graphe où les arêtes représentent l'information d'accessibilité, en définissant un lieu particulier comme l'objectif à atteindre et un ensemble d'actions comme le moyen de le faire. Dans [Posada 2014], la sémantique est utilisée pour définir des lieux et les actions du robot pour aller d'un lieu à l'autre. Des images sphériques sont labellisées puis classées par un module de reconnaissance de lieux basé sur un SVM. La navigation est alors décrite comme une séquence de comportements bas niveau organisés de manière hiérarchique. D'une manière similaire, des régions et des positions sémantiques sont définis dans [Klauss 2011] pour permettre au robot de planifier sa trajectoire à haut niveau. Chaque nœud du graphe est connecté aux autres via des arêtes qui encode l'accessibilité d'un lieux depuis un autre. La planification est faite en utilisant des contraintes métriques comme la distance entre deux nœuds. Dans tous ces travaux, l'information sémantique a été employée pour la construction du modèle utilisé pour la navigation, essentiellement comme un moyen de spécifier des lieux ou les actions du robot. Cependant, elle n'a que peu été utilisée pour guider le choix d'un chemin dans la carte, cette décision reposant essentiellement sur des considérations d'ordre métrique. Ceci est d'autant plus étonnant que l'intérêt de tenir compte de l'information sémantique dans le choix du chemin semble important.

Le problème connexe de l'interprétation d'un chemin à haut niveau a lui aussi reçu une attention significative ces dernières années. Il est en effet intéressant de disposer d'une description à haut niveau d'un chemin pour permettre la communication entre différents agents intelligents, biologiques ou artificiels. Une méthode combinant la labelisation automatique de lieux et l'interprétation de descriptions fournis par l'homme est présentée dans [Landsiedel 2013]. Pour gérer les ambiguïtés éventuelles dues aux erreurs de perception, une distribution de probabilité sur les chemins possibles est calculée et le meilleur chemin est choisi en calculant le MAP de la distribution. Ici le terme meilleur fait référence au chemin correspondant avec le plus de vraisemblance à celui décrit par l'utilisateur. Dans le même esprit, une méthode de traduction automatique du langage naturel est employée dans [Matuszek 2013] pour inférer le chemin le plus probable dans un graphe de Voronoï dont les nœuds sont des lieux bien identifiés. Ces méthodes solutionnent principalement le problème de savoir comment interpréter un chemin à partir d'une description fournie par l'homme. Ici on s'intéresse au problème inverse, à savoir, étant donné un chemin, comment le robot peut-il en générer une description à haut niveau. Dans un contexte opérationnel réaliste, il est particulièrement intéressant de développer cette capacité dès lors que le robot peut être amené à collaborer avec d'autres agents intelligents, biologiques ou non, ne bénéficiant pas forcément des mêmes capteurs. Cette description peut alors servir de langage universel commun à tous les utilisateurs.

De manière générale, la plupart des travaux utilisant la sémantique pour la navigation s'appuient sur la description de l'espace sous forme d'un ensemble de lieux, sémantique-

ment identifiables, différentiables et connectés par les actions possibles du robot ou des contraintes métriques. Or l'attribution d'un label donné à un lieu n'est pas toujours possible, particulièrement en environnement extérieur où le concept de lieu lui-même est mal défini. L'objectif de ce chapitre est de fournir une méthode de sélection automatique de chemin en fonction de contraintes sémantiques ainsi qu'une description de ce dernier qui soient utilisables en extérieur, contrairement aux précédentes méthodes.

9.2 Contraintes sémantiques

Un chemin est défini comme un ensemble orienté de sphères $\{\mathcal{S}_i\}$ adjacentes dans le graphe global et qui relient la position courante à la position cible. Une fois ces positions identifiées avec l'algorithme de localisation présenté au chapitre 8, le problème de la planification de trajectoire consiste à trouver le meilleur chemin dans le graphe au regard d'un jeu de contraintes, pour aller d'une position à l'autre. Ici on souhaite utiliser l'information sémantique contenue dans les cartes locales pour définir ces contraintes sous la forme d'un poids associé à chacune des arêtes du graphe global. Pour cela on définit une heuristique $\mathcal{H}(p)$ qui calcule le poids d'un chemin p en fonction des graphes sémantiques associés aux nœuds du graphe global qui le constituent. Plus élevé est ce coût, plus la trajectoire sera pénalisée. Quatre propriétés de la scène sont utilisées qui modélisent la stabilité et la robustesse de l'information. Elles sont expliquées et justifiées au fur et à mesure de leur introduction.

Nombre d'objets partagés : Deux cartes locales adjacentes partagent une partie de leur information. Ceci se traduit par un ensemble d'objets (zones sémantiques) communs aux deux cartes. Si cet ensemble est petit l'information partagée entre les sphères est faible. Ceci peu provenir de deux choses : soit l'environnement change très vite à cet endroit, soit il existe des erreurs de labelisation significatives. Dans ces deux cas, le robot risque de perdre sa localisation. Il est donc préférable de favoriser les arêtes qui correspondent à des changement modérés de l'environnement, c'est à dire à un grand nombre d'objets partagés par les sphères adjacentes. Pour cela on calcule un score S_P associé à chaque arête :

$$S_P(\mathcal{G}_{\mathcal{S}1}, \mathcal{G}_{\mathcal{S}2}) = 1 - s(\mathcal{G}_{\mathcal{S}1}, \mathcal{G}_{\mathcal{S}2}) \tag{9.1}$$

où s est la similitude des deux sphères, calculée suivant la formule 8.3. Ce score favorise naturellement les parcours utilisant des sphères adjacentes ayant de forte similitude, la redondance d'information étant une condition nécessaire à la robustesse de la navigation.

Classe des objets : les objets peuvent être de plus ou moins bons amers visuels pour la navigation suivant leur degré de stabilité. Par exemple un immeuble ou un panneau sont de bons points de repère tandis qu'une voiture est par définition un mauvais point de repère

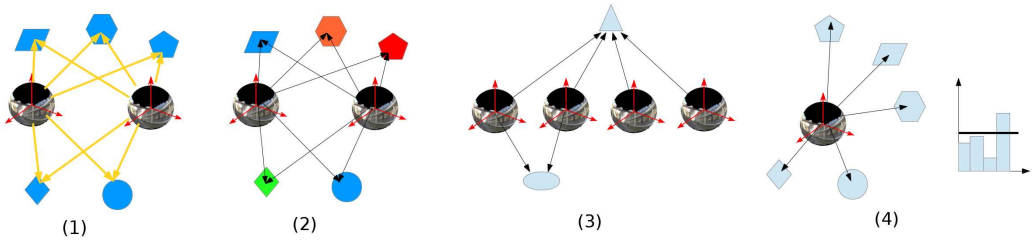


FIGURE 9.1 – *Critères sémantiques utilisés pour la pondération des arêtes : (1) le nombre d'objets partagés par deux sphères adjacentes, (2) la classe de ces objets, (3) l'observabilité des objets, c'est à dire le nombre de sphères dans lesquelles ils sont visibles, (4) la répartition de ces objets.*

puisqu'elle est amenée à se déplacer. En donnant un poids à chacune des classes, on peut pénaliser celles qui sont le moins utile à la navigation et favoriser celles qui le sont le plus. Ainsi à une arête donnée est associé un score S_C dépendant de la classe des objets partagés entre les deux sphères et calculé suivant la formule :

$$S_C(G_1, G_2) = \frac{1}{N_m} \sum_{i \in A_{12}} w(c_i) \tag{9.2}$$

où N_m est le nombre d'objets partagés par les deux sphères, w_i le poids associé à la classe c_i de l'objet i et A_{12} l'ensemble des nœuds partagés par les deux graphes G_1 et G_2. Le fait de moyenner sur l'ensemble des objets permet de donner une mesure de la dynamique globale de la scène et plus généralement de la qualité des objets observés pour la navigation.

Observabilité : En dépits des corrections apportées à la labelisation lors de la construction de la carte, il peut subsister des erreurs qui se manifestent par des pseudo-objets visibles furtivement dans les images. Une manière de minimiser leur impact est de pénaliser d'autant plus fortement un objet qu'il n'est visible que brièvement. Par ailleurs, les objets stables et visibles plus longtemps sont de meilleurs amers visuels et sont donc à privilégier pour la navigation. Chaque objet est donc indexé dans une table qui stocke l'indice des sphères dans lesquelles l'objet est visible. Le nombre de ces sphères, qu'on appelle observabilité, est notée o_i et o_{max} est la valeur maximale des o_i dans cette table. On définit alors un score d'observabilité, noté S_O par :

$$S_O(G_1, G_2) = \frac{1}{N_m} \sum_{i \in A_{12}} (1 - \exp(1 - \frac{o_{max}}{o_i})) \tag{9.3}$$

Répartition spatiale : Pour contraindre correctement le mouvement entre deux sphères, il est important que le robot observe des objets de manière isotrope, c'est à dire que le nombre de ces objets soit sensiblement le même dans toutes les directions. Cette répartition idéale est modélisée par une distribution de probabilité uniforme de trouver des objets dans

toutes les directions d'observations, notée P. Chaque image sphérique est coupée en 4 quartiers correspondant à l'avant, la droite, l'arrière et la gauche, notés Q_{avant}, Q_{droite}, $Q_{arriere}$ et Q_{gauche} respectivement. Pour chacun, les objets sont dénombrés de façon à construire la fonction de répartition effective, notée Q. La pénalité liée au fait d'avoir une répartition non optimale est calculée en utilisant la divergence de Kullback-Leibler de Q par rapport à P :

$$D_{KL}(Q||P) = \sum_{i=1}^{4}(P(i).log(\frac{P(i)}{Q(i)}) \tag{9.4}$$

où la somme est effectuée sur les quatre parties de l'image. Le score associé à la répartition imparfaite des objets, noté S_R est donné par :

$$S_R = 1 - \exp(-D_{KL}(Q||P)) \tag{9.5}$$

Finalement le coût associé à un chemin p dans le graphe constitué de N arêtes est :

$$H(p) = \sum_{i=1}^{N} S_p(G_i, G_{i+1}) + S_C(G_i, G_{i+1}) + S_O(G_i, G_{i+1}) + S_R(G_i, G_{i+1}) \tag{9.6}$$

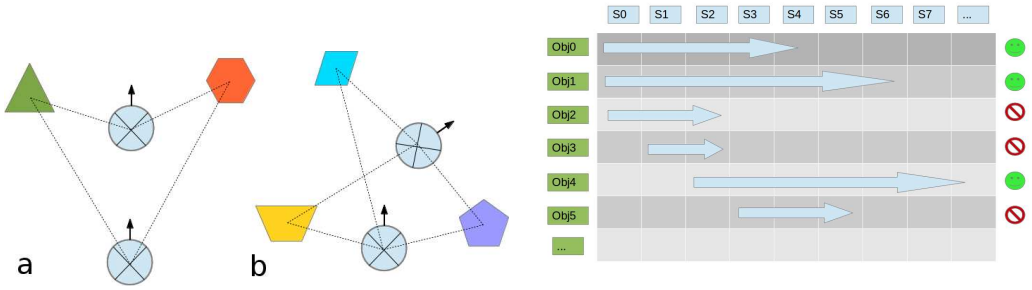
La somme sur l'ensemble des arêtes prend implicitement en compte la longueur du chemin. Une fois tous les poids associés aux arêtes calculés, le meilleur chemin en terme de sémantique est trouvé avec l'algorithme de Dijkstra.

9.3 Caractériser un chemin

Une fois le meilleur chemin trouvé dans le graphe, il peut être intéressant de disposer d'une description automatique de celui-ci, que ce soit pour la partager avec une personne ou un autre robot. L'approche proposée ici permet de décrire un chemin à haut niveau sans avoir besoin de recourir au concept de lieux, difficile à définir en extérieur. En outre, la méthode est rendu robuste aux erreurs de labelisation en privilégiant les objets stables dans le temps.

9.3.1 Langage pour la description de chemin

Un chemin p est défini comme un ensemble de sphères $p = \{\mathcal{S}_i\}$ adjacentes dans le graphe global. Comme expliqué précédemment, chacune de ces sphères partage avec ses voisines un certain nombre d'objets qui sont utilisés ici pour caractériser le déplacement du robot. L'orientation d'une sphère est décrite sémantiquement par le rattachement de chaque objet au quartier dans lequel se trouve son barycentre. Ceci conduit à un jeu de *contraintes d'orientation* de la forme $\mathcal{C}_o(i) = \{objet1 \in Q_{avant}, objet2 \in Q_{droit}...\}$ où i est l'index de la sphère à laquelle est associée la contrainte.



(1) (2)

FIGURE 9.2 – *(1) Exemple de mouvement du robot : en a) les deux objets apparaissent successivement dans Q_{avant} puis dans Q_{droit} et Q_{gauche}, le type de mouvement est une translation ; en b) la position de l'objet violet est la même alors que celle des objets jaune et bleu s'est déplacée vers l'arrière, le type de mouvement est donc une rotation vers la droite. (2) Index des objets utilisés pour calculer l'observabilité. Uniquement les objets ayant l'observabilité la plus longue sont conservés pour décrire le chemin.*

Pour décrire p, une description du mouvement du robot pour chaque transition $\mathcal{S}_i \rightarrow \mathcal{S}_{i+1}$ est associée à l'arête correspondante. Elle se décompose en :

- un type de mouvement : avancer/reculer, tourner à droite/à gauche
- une amplitude exprimée sous la forme : faire *mouvement* jusqu'à ce que *contrainte d'arrêt*

La contrainte d'arrêt est donnée par la position spécifique d'un objet dans la sphère final. Par exemple le mouvement peut être décrit de la manière suivante : "avancer jusqu'à ce que immeuble 1 apparaît dans Q_{avant}".

Pour définir le type de mouvement entre deux images, les positions occupées par les objets dans $\mathcal{S}_i$ et $\mathcal{S}_{i+1}$ sont comparées. Les rotations décalent la position apparente des objets dans la même direction tandis que les translations décalent les objets dans des directions différentes sur la sphère. Un système de vote permet de choisir le mouvement le plus probable. Les contraintes d'arrêt sont déterminées par les objets dont la position a changé entre les deux images. Le processus est illustré à la figure 9.2 (1).

9.3.2 Compression de la description

L'approche proposée pour la description des chemins s'appuie sur l'ensemble des objets observés le long de la trajectoire. Leur nombre peut être arbitrairement grand, résultant en une quantité de contraintes importante. Or ces contraintes peuvent être les mêmes entre plusieurs images successives si les objets observés sont les mêmes et le mouvement de même nature. Il est donc possible de simplifier la description du chemin en utilisant uniquement les objets visibles sur de longue distances et en conservant le plus longtemps possible le même jeu de contraintes. Ces objets sont sélectionnés en fonction de leur observabilité comme illustré à la figure 9.2 (2). Ainsi au lieu de construire de nouvelles contraintes à

chaque étape, le même jeu est conservé tant qu'il est valide. Ceci permet de réduire considérablement la taille de la description et d'atteindre une meilleure robustesse aux erreurs de labellisation qui produisent des objets dont l'observabilité est faible.

9.4 Résultats

Deux séries d'expériences sont proposées pour évaluer d'une part l'algorithme de sélection du meilleur chemin et d'autre part, l'algorithme de description automatique de chemin. Pour chacun, une première série d'expérience est réalisée dans un environnement virtuel dans lequel un capteur permet d'acquérir des images sphériques. L'environnement de simulation permet de générer des situations spécifiques pour mettre en avant les capacités du système. Dans un second temps les algorithmes sont évalués dans un environnement réel composé des images de la base de données acquises sur le campus de l'INRIA de Sophia-Antipolis.

9.4.1 Planification de trajectoire

9.4.1.1 Environnement virtuel

Le premier test réalisé vise à évaluer le comportement de l'algorithme de sélection de chemin en fonction des caractéristiques de l'environnement. Le modèle créé correspond à un milieu urbain dans lequel on retrouve les même classes que dans la base de données INRIA, à savoir : la route, le ciel, des immeubles, des trottoirs, des arbres, des voitures, des panneaux de signalisation, des marquages au sol. Les poids associés aux classes sont arbitrairement fixés de la manière suivante : 0.9 pour les voitures qui sont des objets dynamiques, 0.5 pour la route et les trottoirs qui ne sont pas des objets dynamiques mais ne sont pas non plus caractéristiques d'un lieu et 0.1 pour les arbres, les panneaux, les immeubles et les signes au sol qui sont stables et caractéristiques.

La carte construite est composée de quatre chemins qui partent de la même position et arrive au même endroit mais le long desquels l'environnement est différent. Elle est illustrée à la figure 9.3. Le premier chemin (noir) est composé de longue lignes droites de telle façon que les objets sont visibles sur de longue distances. Le robot y rencontre beaucoup d'objets dynamiques. Le second chemin est composé segments droits plus courts et le robot observe des panneaux et quelques voitures de telle façon que l'environnement apparaît modérément dynamique. Le troisième chemin se distingue par un grand nombre de panneaux qui sont des amers visuels stables et seulement quelques voitures. Il est constitué de trois segments linéaires. Enfin le quatrième segment est composé de deux longues lignes droites le long desquelles le robot n'observe que peu d'objets.

Les résultats sont présentés à la table 9.1 où les scores pour chacun des chemins sont rapportés ainsi que le score relatif. Il convient de noter que le score le plus élevé est le

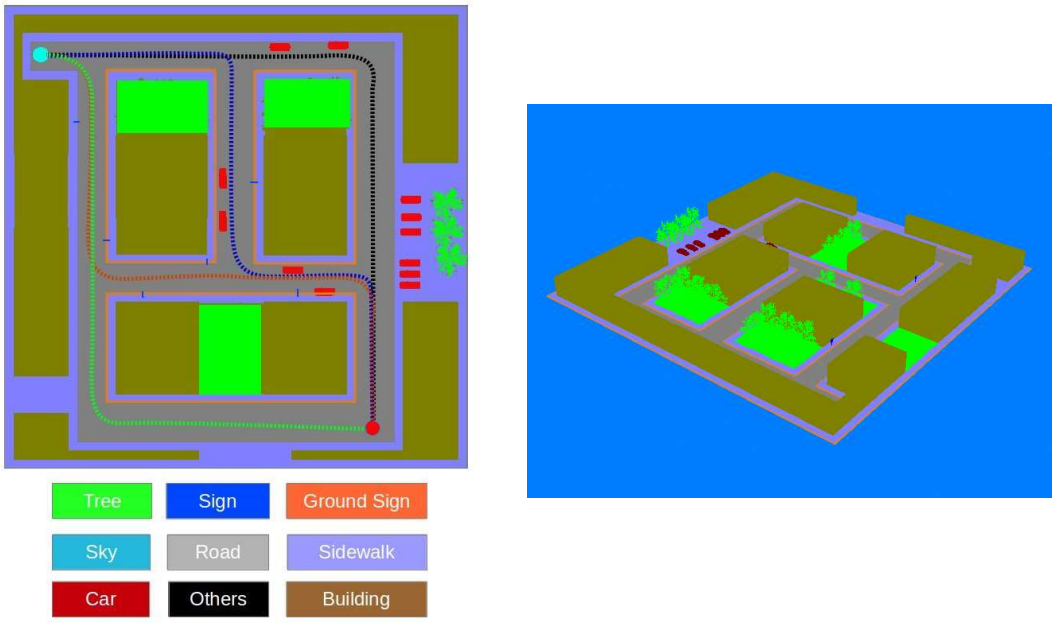


FIGURE 9.3 – *A gauche : Carte 2D de l'environnement virtuel 3D utilisé pour les tests. a droite : vue 3D.*

TABLE 9.1 – *Résultat pour le choix du meilleur chemin. Environnement virtuel.*

Chemin	S_P	S_C	S_O	S_R	Poids Relatif
(1)	19	102	93	**148**	100
(2)	21	85	91	153	98
(3)	**18**	**81**	**88**	149	**92**
(4)	19	**81**	90	158	95

plus pénalisant. Le meilleur chemin est le troisième. Celui-ci offre en effet le meilleur compromis entre les différents scores avec un grand nombre d'objets visibles, la plupart étant stables. A l'inverse le premier chemin contient beaucoup d'objets dynamiques, il offre donc le plus mauvais score S_C. De plus, lorsque deux voitures sont garées côte à côte, elles peuvent apparaître comme une zone sémantique unique puis subitement comme deux zones séparés. L'algorithme de mise en correspondance des nœuds ne peut alors pas traquer les véhicules et les considère comme de nouveaux objets ce qui réduit leur observabilité. Le plus mauvais score S_O correspond donc aussi à ce chemin. Le quatrième chemin ne contient aucun objets dynamique et est constitué de long segments de droite, donc d'une bonne observabilité. Son poids est donc relativement bas. Cependant le faible nombre d'objets observables suivant ce chemin le rend moins attractif que le troisième.

Enfin le deuxième chemin possède le plus mauvais score S_P. Ceci peut s'expliquer par la présence de plusieurs virages qui modifient rapidement les objets visibles et rend de ce fait difficile la mise en correspondance des zones sémantiques.

Au travers de cet exemple simple, on observe que l'algorithme favorise naturellement les trajectoires linéaires (S_O) pour lesquelles l'observabilité est meilleure, les zones riches en objets qui favorisent la redondance d'information entre deux images (S_p) et aident à contraindre le mouvement (S_R) et les zones dont la dynamique est faible (S_C). L'algorithme permet donc d'ajuster finement le choix de la trajectoire en fonction de paramètres complexes qui tiennent compte de multiples facteurs influençant les performances de navigation du robot. Il est capable de discriminer des trajectoires de longueurs sensiblement égales mais qui s'avèrent inégales pour la navigation d'un robot, ce qui n'est pas possible avec une approche purement métrique.

9.4.1.2 Environnement réel

La seconde expérience est réalisée en utilisant les images de référence de la base de données acquise sur le campus de l'INRIA. Du fait de la configuration du campus il n'est pas possible depuis un unique point de départ de rejoindre un unique point d'arrivée en utilisant plusieurs chemins. Le test a donc consisté à discriminer deux chemins partant de points différents et arrivant au même endroit, comme illustré à la figure 9.4 où le chemin 1 est en bleu et le chemin 2 en rouge. Chaque chemin a sensiblement la même longueur, celle-ci est rapportée en valeur relative à la table 9.2. Le premier est une ligne droite depuis la position de départ jusqu'à la position d'arrivée. Deux sections comportent beaucoup de voitures garées le long de la route. Le second chemin passe dans un environnement changeant avec des arbres qui génèrent beaucoup d'occlusions, de nombreux endroits où de petits groupes de voitures sont garées.

TABLE 9.2 – *Résultats pour le monde réel.*

Chemin	S_P	S_C	S_O	S_R	Poids relatif	Longueur relative
(1)	49	**53**	**58**	**91**	**92**	100
(2)	49	65	65	93	100	**97**

Le premier chemin, bien que légèrement plus long, est le meilleur avec un score relatif inférieur de 8% à celui du second et notamment un score S_C inférieur de 12%. Ceci s'explique en regardant la répartition des voitures le long du chemin. Dans le premier cas, beaucoup de voitures sont présentes au début et à la fin du chemin mais elles sont toutes côte à côte. Elles apparaissent donc comme un petit nombre de zones sémantiques. A l'inverse dans le second cas, plusieurs petits groupes de voitures apparaissent tout au long de la route, générant de ce fait plus de zones sémantiques "voitures" qui pénalisent la trajectoire. Le score S_C ne dépend pas de la quantité de voitures observées mais de la quantité de zones

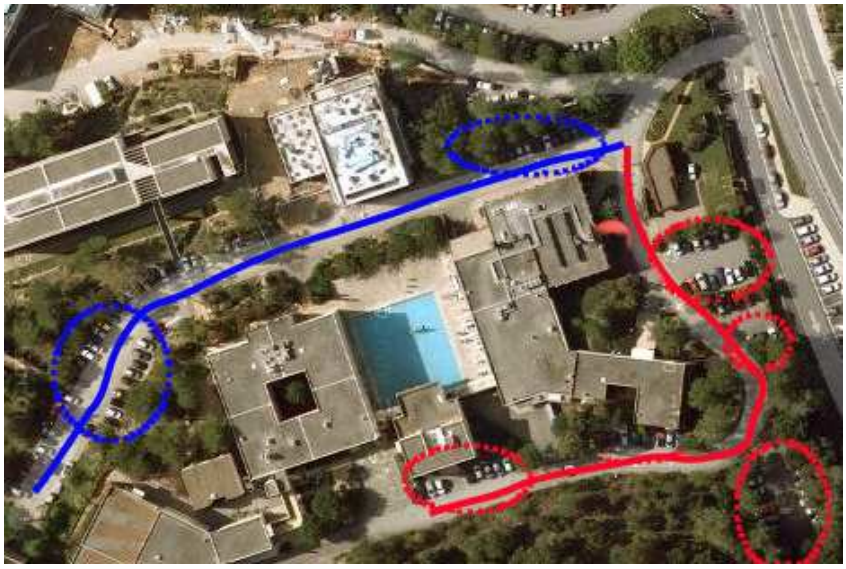


FIGURE 9.4 – *Vue satellite des deux chemins utilisés sur le campus de l'INRIA pour réaliser les tests. Les cercles matérialisent les endroits où se trouve des voitures garées.*

sémantiques, chacune pouvant regrouper plusieurs voitures. De plus la première trajectoire est rectiligne ce qui permet de traquer les objets sur de longues distances contrairement au second chemin. Ceci valide donc l'approche proposée dans un environnement réaliste où le chemin le plus court n'est pas nécessairement le plus intéressant en tenant compte des contraintes de la navigation.

9.4.2 Description de chemin

La description automatique de chemin a été évaluée suivant deux paramètres : la vraisemblance de la description produite et sa compacité. Étant donnée la longueur de la description pour un chemin complet, la vraisemblance de la description n'a été évaluée que sur un chemin court dans un environnement de simulation. La compacité de la description a ensuite été évaluée dans un environnement réaliste composé d'images de la base de données acquise sur le campus de l'INRIA.

9.4.2.1 Environnement virtuel

Vraisemblance : Le chemin dans l'environnement virtuel, illustré à la figure 9.5, consiste en une ligne droite modélisant un corridor urbain, le long duquel le robot génère trois ensembles de contraintes rapportés ci-dessous. Les lignes noires montrent les objets utilisés par le robot comme objets de référence.

– **`Orientation:`** `immeuble (1) gauche, panneau (1) avant,`

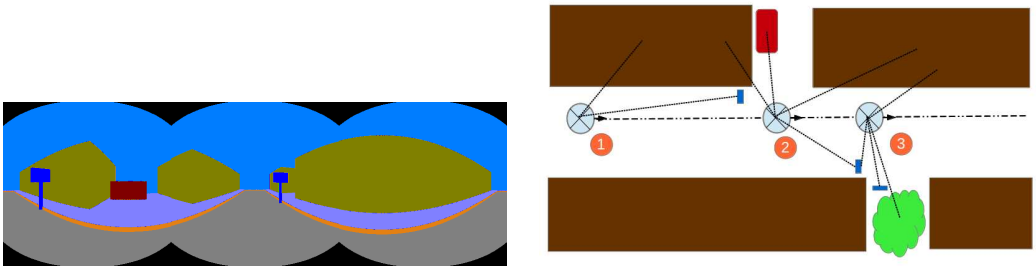


FIGURE 9.5 – *Illustration de l'environnement simulé utilisé pour valider la vraisemblance de la description générée. A gauche : image sphérique de l'environnement annoté ; à droite : carte de l'environnement.*

Avancer jusqu'à ce que: car (1) gauche,
- **Orientation:** immeuble (1)gauche, panneau (2) avant, voiture (1) gauche **Avancer jusqu'à ce que:** immeuble (1) plus visible, panneau (2) droite
- **Orientation:** immeuble (2) arrière, panneau (1) arrière, immeuble (3) avant, panneau (2) droite, panneau (3) droite

- **Avancer jusqu'à ce que :** fin

La description fournie est vraisemblable et compréhensible, ce qui, en dépit de la simplicité de l'exemple, montre la capacité du robot à offrir une description accessible à l'homme.

9.4.2.2 Environnement réel

Compacité : Le second critère d'évaluation des performances de l'algorithme est la compacité de la description fournie. Pour l'évaluer les descriptions de deux chemins ont été générés, chacun étant composé de 200 images de la base de données INRIA, échantillonnée à une fréquence de 5Hz. La relative haute fréquence des images assure une forte redondance de l'information entre des sphères successives et permet d'évaluer la capacité de l'algorithme à ne conserver que l'information utile. Le long du premier chemin, une ligne droite visible à la figure 9.6 gauche, le robot a généré automatiquement 7 jeux de contraintes représentant une compacité de 7/200. Pour le second chemin, une courbe visible à la figure 9.6 à droite, le robot a généré 17 jeux de contraintes, pour une compacité de 17/200. La grande différence entre les deux chemins s'explique par la faible observabilité des objets dans le second cas due à la rotation du véhicule et à l'horizon situé à courte distance du fait de la présence d'obstacles, principalement des arbres. A l'inverse, dans le premier cas, la route dégagée permet au robot de voir les objets sur de plus longues distances et donc de réduire le nombre de contraintes nécessaires pour décrire le chemin.

FIGURE 9.6 – *Deux chemins utilisés pour évaluer la compacité de la description générée par l'algorithme.*

Dans tous les cas cette méthode permet de fournir une description relativement compacte d'un chemin en utilisant uniquement les objets observés avec un taux de compression de supérieur à 10 par rapport à une désinscription naïve utilisant n'importe quel objet.

9.5 Conclusion

Ce chapitre a permis d'introduire une méthode de planification de trajectoire capable de tenir compte du contenu sémantique de la scène pour choisir le meilleur chemin permettant d'atteindre la position souhaitée. Contrairement à beaucoup de méthodes classiques, le chemin le plus court n'est pas systématiquement choisi et une analyse de l'information sémantique permet la sélection d'un parcours qui limite les risques de collision ou de perte de localisation. Un algorithme pour générer automatiquement une description compacte du chemin à partir des objets observés dans la scène a aussi été proposé. Les deux approches ont été testées d'abord dans des environnements de simulation se prêtant bien à la génération de situations particulières, puis validés dans un environnement réel. Ces algorithmes permettent d'envisager la planification de trajectoire sous un angle nouveau en simplifiant l'interface homme/robot et en autorisant la prise en compte de contraintes complexe dans la décision du meilleur chemin.

Quatrième partie

Conclusion et Perspective

Conclusion

Les travaux présentés dans cette thèse ont mis en évidence l'importance d'intégrer de l'information sémantique dans les cartes utilisées pour la navigation. Non seulement elle permet de simplifier les interactions hommes/robots et d'améliorer les performances de ces derniers dans la réalisation des tâches qui leurs sont attribuées, mais elle permet également d'envisager de nouveaux comportements.

Le modèle de carte proposé décompose l'environnement en représentations locales egocentrées, décrites à haut niveau par un *graphe sémantique* et indexées globalement par le contenu de ces graphes. L'analyse statistique de ces représentations permet de doter le robot d'un embryon de sens commun qui peut être utilisé pour détecter des erreurs de perception et ainsi augmenter la fiabilité du modèle. La mise à jour de la carte avec un jeu réduit de données, est rendue possible par l'analyse de l'environnement à haut niveau s'appuyant sur le modèle des graphes sémantiques. A partir des occlusions observées et la relation entre les classes, le robot est capable de généraliser ses observations à d'autres parties de l'environnement. La stratégie proposée s'est montrée particulièrement utile dans de très grands environnements où la réalisation de multiples acquisitions est à la fois coûteuse et délicate puisqu'elle génère d'importantes quantités de données qu'il est difficile de gérer par la suite. Les tests réalisés avec des images acquises à plusieurs années d'intervalle ont montré la capacité du système à retrouver l'aspect de l'environnement avec un très haut niveau de précision. La compréhension de la scène permet en outre au robot de déduire l'existence de lieux non observés via le processus d'extrapolation de carte. En exploitant les objets dynamiques qu'il observe et le contexte sémantique, le robot est capable par ce procédé, de révéler les structures de l'environnement qui ne sont pas visibles mais pour lesquelles des indices de présence existent. Les expériences réalisées ont montré qu'un robot est capable d'étendre de 20% le domaine couvert par la carte, à la fois en inférant l'espace libre et la topologie avec une précision élevée.

L'intérêt de l'information sémantique pour la navigation a aussi été démontré. L'algorithme de localisation proposé, basé sur la comparaison de graphes sémantiques et l'indexation des images sphériques, s'est avéré à la fois efficace, robuste et souple dans toutes les expériences réalisées. La localisation basée graphe sémantique est plus rapide que celle basée sac-de-mots. Elle est robuste aux changements de point de vue modérés et l'est davantage aux changements d'apparence de la scène que les méthodes usuelles, ce qui la destine particulièrement aux environnements dynamiques. Elle autorise par ailleurs la formulation de requêtes complexes, ce qui est impossible avec les méthodes classiques bas niveau. Enfin, le meilleur chemin reliant deux positions peut être choisi en fonction de contraintes complexes exprimées par des concepts évolués. Il a été proposé de tenir compte de la nature et du nombre d'objets observés, de leur stabilité et de leur répartition pour modéliser le dynamisme de la scène et l'intérêt d'un chemin. Les expériences en simulation et en environnement réel ont montrés la capacité du robot à choisir un chemin en fonction de critères autres que métriques, ce qui s'avère d'autant plus judicieux que le chemin le plus court

peut s'avérer dangereux ou qu'il peut être difficile de s'y localiser.
La prise en compte de l'information sémantique améliore donc significativement les performances de navigation des robots. Ces bons résultats peuvent être pondérés par leur sensibilité à la qualité de la labellisation si les erreurs d'interprétation de scène deviennent importantes. Cependant, comme il a été montré, il existe aujourd'hui des algorithmes performants capables de produire des images annotées avec un taux de confiance élevé et ceci dans des environnements réels complexes. Par ailleurs, plusieurs algorithmes ont été proposés pour corriger les erreurs de perception par un filtrage temporel des résultats ou un raisonnement à haut niveau qui ont été, là encore, validés dans un environnement réel.

Perspectives

La difficulté à naviguer en utilisant des modèles métriques et topologiques peut être surprenante de prime abord. En effet, les efforts de recherche massifs des dernières décennies dans le domaine de la cartographie ont permis de développer des systèmes à même de construire des modèles de l'environnement avec une précision bien plus grande que ce dont est capable l'être humain. Qui pourrait aujourd'hui prétendre estimer la longueur d'un chemin avec une précision approchant celle obtenue par les robots ? Pourtant, les capacités limités de ces systèmes à naviguer dans des environnements réels interrogent. S'ils sont effectivement mieux à même de représenter l'espace que les êtres vivants, pourquoi ne peuvent-ils pas y naviguer avec au moins autant d'efficacité ?
Une partie de la réponse tient visiblement dans la nature de l'information représentée dans ces cartes. En effet, si la géométrie modélise l'apparence de l'environnement et des objets qui s'y trouvent, elle n'informe en rien sur la nature de ces objets. Pourtant la compréhension de cette dernière est essentielle pour modéliser le comportement dynamique de la scène et découvrir sa structure. Il est donc fondamental de l'intégrer dans les représentations utilisées par les robots pour naviguer. De nouvelles stratégies de modélisation basées sur l'interprétation de l'information reçue des capteurs ont donc été développées, dont la cartographie sémantique est l'évolution la plus récente et sans doute la plus aboutie. Ce processus d'abstraction croissant des concepts encodés dans les représentations a permis d'envisager une approche enrichie de la cartographie qui ne se limite plus à mettre en relation les données acquises, mais tente d'en découvrir la structure et la signification.

Dans ce cadre, le modèle présenté dans cette thèse s'appuie largement sur l'information sémantique pour proposer une représentation utilisable pour la navigation dans des environnements vastes et dynamiques. A la lumière des résultats présentés, on constate que l'information sémantique n'est pas seulement utile à la navigation, mais qu'elle est aussi nécessaire, la simple mémorisation de l'apparence des lieux, aussi parfaite soit-elle, étant insuffisante pour naviguer dans un environnement réel. Il est même envisageable de faire évoluer ce modèle pour naviguer sans faire appel explicitement à l'information métrique. Étant donnée une série d'images provenant d'une acquisition, la distance métrique entre

chacune d'elles peut être remplacée par une estimation de leurs similitudes en terme de contenu des graphes sémantiques associés. Le graphe des sphères peut alors être construit en pondérant les arêtes selon le procédé proposé dans le chapitre 9 pour le choix du meilleur chemin plutôt qu'en utilisant l'odométrie. Les fonctions de navigation proposées dans cette thèse restent alors réalisables de la même façon que ce qui a été présenté précédemment. A priori contre-intuitif puisque non spatial, ce type de modèle donne des indices utiles sur la nature de l'information utile pour la navigation. Tout autant qu'une carte spatiale de son environnement, le robot a besoin d'un modèle cognitif de celui-ci, lui permettant d'accéder à la compréhension de ce qu'il perçoit. Il est donc nécessaire de redéfinir l'objectif de la cartographie, qui, initialement considérée comme le fait d'acquérir un modèle spatial de l'environnement, est appelée à évoluer et probablement à se scinder en deux tâches distinctes consistant d'une part, en l'acquisition d'un modèle non spatial de l'environnement, identifiant les objets présents dans celui-ci, leurs comportements et leurs relations et d'autre part la construction d'une représentation mémorisant la perception conjointe de ces objets à des instants donnés. Des modèles de cartes basées sur la mémorisation des perceptions selon un schéma comparable à la mémoire biologique ont déjà été proposés, par exemple dans [Milford et al. 2004], mais sans leur adjoindre de représentation sémantique. Il semble donc que les tentatives de construire des systèmes autonomes en environnement réel n'aient pas rencontré le succès attendu en dépits de la précision des modèles construits parce qu'une étape fondamentale du processus de cartographie, préalable à la construction d'un modèle spatiale, était omise : la compréhension de la scène. Ce fossé est appelé à être comblé par la cartographie sémantique qui ambitionne de fournir aux robots la compréhension de l'environnement qui leur manquait.

Bien qu'encore au début de l'exploration des méthodes de cartographie sémantique, il est possible de voir plus loin en envisageant ce qui sera probablement la prochaine étape menant vers l'autonomie des robots. Les recherches récentes conduisent aujourd'hui au développement de modèles sémantiques permettant de rendre accessible aux robots une compréhension minimale de leur environnement. Mais beaucoup de problèmes ne pourront être résolus par l'ajout d'information sémantique. Comment comprendre le reflet d'un objet sur une vitre sans comprendre la propagation de la lumière ? Comment déterminer si un chemin est praticable sans comprendre les notions d'adhérence et d'équilibre ? Comment déterminer les interactions possibles avec un objet sans en connaître ses propriétés physiques comme son poids ? Il est vraisemblable qu'il faudra, à terme, intégrer dans ces représentations une modélisation physique de la scène, qui est, finalement, la seule manière objective de décrire la réalité. Ceci permettra de renforcer l'adaptabilité des robots aux problèmes nouveaux auxquels ils seront nécessairement confrontés dans le monde réel.

En marge de cette évolution, souhaitable à moyen terme, des modifications de moindre envergure peuvent être apportées aux travaux récents, notamment ceux présentés dans cette thèse. Par exemple, l'analyse statistique des graphes sémantiques et des observations a été utilisée à la fois pour détecter des erreurs de perception et pour mettre à jour la carte. Il pourrait être intéressant de fusionner ces approches pour créer un modèle global des rela-

tions entre les classes. Ceci est aussi valable pour l'extrapolation de cartes, qui est rendu possible par l'analyse sémantique et dynamique de la scène. Concevoir un modèle cognitif intégrant toutes ces informations, qui pourrait prendre la forme du graphe conceptuel présenté ici mais enrichi de bien d'autres propriétés, serait sans doute utile au robot.

Bibliographie

Publications personnelles

[Drouilly et al 2014a] Drouilly, R. and Rives, P. and Morisset, B. *Fast Hybrid Relocation in Large Scale Metric-Topologic-Semantic Map* IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'14, Chicago, IL, p1839–1845 (Cited on page 4.)

[Drouilly et al 2014b] Drouilly, P. Papadakis, R. and Rives, P. and Morisset, B. *Local Map Extrapolation in Dynamic Environments* IEEE International Conference on Systems, Man, and Cybernetics, San Diego, USA, Octobre 2014 (Cited on page 5.)

[Drouilly et al 2015] Drouilly, R. and Rives, P. and Morisset, B. *Semantic Representation For Navigation In Large-Scale Environments* IEEE Int. Conf. on Robotics and Automation, ICRA'15, Seattle, WA, Mai 2015 (Cited on page 5.)

[Rives et al 2014] P. Rives, R. Drouilly, T. Gokhool *Représentation orientée navigation d'environnements à grande échelle* In Reconnaissance de Formes et Intelligence Artificielle, RFIA 2014, France, Juin 2014 (Not cited.)

Références

[Anati 2012] Anati, R., Scaramuzza, D., Derpanis, K. G., and Daniilidis, K. *Robot localization using soft object detection* In Robotics and Automation (ICRA), 2012 IEEE (Cited on pages 24 and 100.)

[Audras 2011] Audras, C., Comport, A.I., Meilland, M. and Rives, P. *Real-time appearance-based slam for rgb-d sensors* In Australian Conference on Robotics and Automation 2011. 7, 48 (Cited on page 34.)

[Aydemir 2012] Alper Aydemir, Patric Jensfelt and John Folkesson *What can we learn from 38,000 rooms ? Reasoning about unexplored space in indoor environments* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2012 (Cited on page 23.)

[Bailey 1999] Bailey T., Nebot E. M., Rosenblatt J. K., Durrant-Whyte H. F., *Robust Distinctive Place Recognition for Topological Maps* International Conference on Field and Service Robotics, 1999. (Cited on page 15.)

[Baker 2001] Baker, P., Fermuller, C., Aloimonos, Y. and Pless, R. *A spherical eye from mula profondeur de l'architecture doit être suffisante pour permettre une bonne généraliltiple cameras* IEEE International Conference on Computer Vision and Pattern Recognition, 576. 40, 46, 123 (Cited on page 33.)

[Barreto 2006] João P. Barreto *A unifying geometric representation for central projection systems* Computer Vision and Image Understanding, Volume 103, Issue 3, September 2006, Pages 208-217, ISSN 1077-3142 (Cited on page 33.)

[Bengio 2007] Y. Bengio *Learning deep architecture for AI* Technical Report 1312, Université de Montréal,2007 (Cited on page 46.)

[Biber 2003] Biber, P. and Strasser, W. *The normal distributions transform : A new approach to laser scan matching.* IEEE International Conference on Intelligent Robots and Systems, IROS, 2003 (Cited on page 13.)

[Biber 2008] Peter Biber and Tom Duckett, *Experimental Analysis of Sample-Based Maps for Long-Term SLAM* 2008 (Cited on page 74.)

[Bo 2011] Bo, Liefeng, Xiaofeng Ren, and Dieter Fox *Hierarchical matching pursuit for image classification : Architecture and fast algorithms* Advances in neural information processing systems. 2011. (Cited on page 43.)

[Borkowski 2010] Borkowski, Adam and Siemiatkowska, Barbara and Szklarski, Jacek *Towards Semantic Navigation in Mobile Robotics* Lecture Notes in Computer Science, Graph Transformations and Model-Driven Engineering, Springer Berlin Heidelberg, p719-748, 2010 (Cited on pages 24 and 116.)

[Brunskill 2007] Brunskill, E. ; Kollar, T. ; Roy, N., *Topological mapping using spectral clustering and classification* Intelligent Robots and Systems, 2007. IROS 2007. (Cited on pages 17 and 18.)

[Burgard 1999] W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. *Experiences with an interactive museum tour-guide robot.* Artificial Intelligence , 114(1-2) :-55, 1999. (Cited on page 10.)

[Burgard 2003] D. Hähnel, D. Schulz, and W. Burgard *Mobile robot mapping in a populated environments* Advanced Robotics, vol. 17, no. 7, pp. 579-597, 2003 (Cited on page 73.)

[Byung-soo 2013] Byung-soo Kim Kohli, P. et Savarese, S., *3D Scene Understanding by Voxel-CRF* Computer Vision (ICCV), 2013 IEEE International Conference on, 2013 (Cited on page 43.)

[Cadena 2013] Cadena Lerma, C., and J. Kosecka *Semantic parsing for priming object detection in RGB-D Scenes* Semantic Perception, Mapping and Exploration Workshop, Karlsruhe, Germany (2013). (Cited on pages 42 and 43.)

[Chapoulie 2012] Alexandre Chapoulie, Patrick Rives, David Filliat. *Topological segmentation of indoors/outdoors sequences of spherical views* IEEE Conference on Intelligent Robots and Systems, IROS, 2012 (Cited on page 15.)

[Chatila 1985] R. Chatila and J.-P. Laumond. *Position referencing and consistent world modeling for mobile robots.* IEEE International Conference on Robotics and Automation, 1985 (Cited on page 12.)

[Cobzas 2003] Cobzas, D., Zhang, H. and Jagersand, M. *Image-based localization with depth- enhanced image map* In IEEE International Conference on Intelligent Robots and Systems, 1570–1575. 19, 39, 48 (Cited on page 34.)

[Cole 2006] Cole D, Newman P *Using laser range data for 3D SLAM in outdoor environments.* IEEE Int. Conf. on Robotics and Automation (ICRA) 2006 (Cited on page 12.)

[Courbon et al. 2009] Courbon, J., Mezouar, Y. et Martinet, P. *Autonomous navigation of vehicles from a visual memory using a generic camera model.* Intelligent Transport System, 10, 392–402, 2009. (Cited on page 29.)

[Cummins 2009] Cummins, Mark AND Newman, Paul *Highly scalable appearance-only SLAM, FAB-MAP 2.0* Proceedings of Robotics : Science and Systems, 2009, Seattle, USA (Cited on page 99.)

[Curless 1996] B. Curless and M. Levoy. *A volumetric method for building complex models from range images.* ACM Transactions on Graphics (SIGGRAPH), 1996 (Cited on page 13.)

[Davidson 2007] Davison, A.J. ; Reid, I.D. ; Molton, N.D. ; Stasse, O. *"MonoSLAM : Real-Time Single Camera SLAM,"* Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.29, no.6, pp.1052,1067, June 2007. (Cited on page 12.)

[Dayoub 2011] Feras, Dayoub and Grzegorz, Cielniak and Tom, Duckett *Long-term experiments with an adaptive spherical view representation for navigation in changing environments* Robotics and Autonomous Systems,2009 (Cited on page 73.)

[Dedeoglu 1999] G. Dedeoglu, M. Mataric, and G. S. Sukhatme. *Incremental, online topological map building with a mobile robot.* Mobile Robots XIV - SPIE, pages 129–139, 1999 (Cited on page 15.)

[Dong 2013] Dong Wook, K., Chuho, Y., Il Hong S., *Semantic mapping and navigation : A Bayesian approach* Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ (Cited on pages 24 and 100.)

[Dryanovski 2010] Dryanovski I, Morris W, Xiao J *Multi-volume occupancy grids : An efficient probabilistic 3D mapping model for micro aerial vehicles.* IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 2010 (Cited on page 10.)

[Elfes 1987] A. Elfes. *Sonar-based real-world mapping and navigation*. IEEE Journal of Robotics and Automation, RA-3(3) :249-265, June 1987. (Cited on page 10.)

[Elfes 1989] A. Elfes. *Occupancy Grids : A Probabilistic Framework for Robot Perception and Navigation* PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989. (Cited on page 10.)

[Engelson 1992] Engelson, S.P. ; McDermott, D.V. *Error correction in mobile robot map learning* IEEE International Conference on Robotics and Automation, 1992. (Cited on page 15.)

[Filliat 2007] Filliat, David *A visual bag of words method for interactive qualitative localization and mapping* International Conference on Robotics and Automation, 2007 (Cited on page 24.)

[Fitzgibbon 1995] Andrew W. Fitzgibbon, R.B.Fisher. *A Buyer's Guide to Conic Fitting* Proc.5th British Machine Vision Conference, Birmingham, pp. 513-522, 1995. (Cited on page 69.)

[Furukawa 2009] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. *Reconstructing building interiors from images.* International Conference on Computer Vision (ICCV), 2009. (Cited on page 10.)

[Galindo 2005] Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J. A., and González, J. (2005, August). *Multi-hierarchical semantic maps for mobile robotics.* In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on (pp. 2278-2283). IEEE. (Cited on pages 22 and 23.)

[Galindo 2008] Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, Alessandro Saffiotti *Robot task planning using semantic maps* Robotics and Autonomous Systems, Volume 56, Issue 11, 30 November 2008, Pages 955-966, ISSN (Cited on page 115.)

[Galindo 2011] Cipriano Galindo, Juan-Antonio Fernandez-Madrigal, Javier Gonzalez, and Alessandro Saffiotti. *Robot task planning using semantic maps* Robot. Auton. Syst. 56, 11 (November 2008), 955-966. (Cited on page 24.)

[Gallegos et al. 2010] Gallegos, G., Meilland, M., Rives, P. and Comport, A.I. *Appearance-based slam relying on a hybrid laser omnidirectional sensor* In IEEE International Conference on Intelligent Robots and Systems, 3005 –3010. 7, 17, 48 (Cited on page 34.)

[Gaussier 2000] P. Gaussier, C. Joulain, J.P. Banquet, S. Lepretre, and A. Revel. *The visual homing problem : an example of robotics/biology cross-fertilisation.* Robotics and autonomous systems, 2000. (Cited on page 15.)

[Geiger et al. 2010] Geiger, A., Roser, M. and Urtasun, R. *Efficient large-scale stereo matching* Asian Conference on Computer Vision 2010 (Cited on page 38.)

[German Ros 2015] G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez and A.M. Lopez. *Vision-based Offline-Online Perception Paradigm for Autonomous Driving* In Winter Conference on Applications of Computer Vision (WACV), 2015. (Cited on pages 54, 55 and 56.)

[Grzonka 2010] Grzonka, S. and Dijoux, F. and Karwath, A. and Burgard, W. *Mapping indoor environments based on human activity* Int. Conf. on Robotics and Automation, 2010 (Cited on page 84.)

[Habbeck 2007] Habbecke M, Kobbelt L *A surface-growing approach to multi-view stereo reconstruction.* Conf. on Computer Vision and Pattern Recognition (CVPR) 2007 (Cited on page 13.)

[Hadsell 2009] Hadsell R, Bagnell JA, Hebert M *Accurate rough terrain estimation with space-carving kernels* Robotics Science and Systems (RSS) 2009 (Cited on page 10.)

[Hall 1966] Edward Hall *The Hidden Dimension : Man's Use of Space in Public and Private* The Bodley Head Ltd, 1966 (Cited on page 89.)

[Hane 2013] Hane, C., Zach, C., Cohen, A., Angst, R., and Pollefeys, M. *Joint 3d scene reconstruction and class segmentation* In Computer Vision and Pattern Recognition (CVPR), 2013 (Cited on page 23.)

[Herbert 1989] Hebert M, Caillas C, Krotkov E, Kweon IS, Kanade T *Terrain mapping for a roving planetary explorer* IEEE Int. Conf. on Robotics and Automation 1989 (Cited on page 10.)

[Hornung 2013] Armin Hornung and Kai M. Wurm and Maren Bennewitz and Cyrill Stachniss and Wolfram Burgard. *OctoMap : An Efficient Probabilistic 3D Mapping Framework Based on Octrees.* Autonomous Robots, 2013. (Cited on pages 11 and 17.)

[Huber 1981] P.J. Huber. *Robust Statistics* New york, Wiley, 1981. (Cited on page 40.)

[Jia 1999] Jia Li ; Najmi, A. ; Gray, R.M., *Image classification by a two dimensional hidden Markov model* Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on , vol.6, no., pp.3313,3316 vol.6, 15-19 Mar 1999 (Cited on page 43.)

[Jiang 2013] Yun Jiang and Koppula, H. and Saxena, A. *Hallucinated Humans as the Hidden Context for Labeling 3D Scenes* IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2013 (Cited on page 84.)

[Kim et Hilton 2009] Kim, H. and Hilton, A. *Environment modelling using spherical stereo imaging* In International Conference On 3-D Digital Imaging and Modeling, 1534–1541. 48 (Cited on page 35.)

[Klauss 2011] Klaus, Uhl and Arne, Roennau and Rudiger, Dillmann *From Structure to Actions : Semantic Navigation Planning in Office Environments* IROS 2011 Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment (Cited on page 116.)

[Krähenbühl 2011] Philipp Krähenbühl and Koltun, Vladlen *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials* Advances in Neural Information Processing Systems 24, 2011 (Cited on pages 44, 50, 51 and 55.)

[Krajník 2014] Krajník, T. and Fentanes, J.P. and Cielniak, G,. and Dondrup, C. and Duckett, T. *Spectral analysis for long-term robotic mapping* International Conference on Robotics and Automation (ICRA), 2014 (Cited on page 74.)

[Kuipers 2000] Benjamin Kuipers *The Spatial Semantic Hierarchy* Artificial Intelligence, vol 119, 2000 (Cited on page 21.)

[Kunz 1997] C. Kunz, T. Willeke, and I. Nourbakhsh. *Automatic mapping of dynamic office environments.* IEEE International Conference on Robotics and Automation (ICRA), 1997. (Cited on page 15.)

[Lafferty 2001] John Lafferty, Andrew McCallum, Fernando Pereira *Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data* In Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001), 2001. (Cited on page 49.)

[Landsiedel 2013] Landsiedel, C.; de Nijs, R.; Kuhnlenz, K.; Wollherr, D.; Buss, M. *Route description interpretation on automatically labeled robot maps* Robotics and Automation (ICRA), 2013 IEEE International Conference on , vol., no., pp.2251,2256, 6-10 May 2013 (Cited on pages 15 and 116.)

[Leung 2001] T. Leung and J. Malik *Representing and recognizing the visual appearance of materials using three-dimensional textons* International Journal of Computer Vision (Cited on page 43.)

[Li 2006] Li, S. *Real-time spherical stereo. In IEEE International* Conference on Computer Vision and Pattern Recognition, vol. 3, 1046 –1049. 48 (Cited on page 35.)

[Lisien 2005] B. Lisien, D. Morales, D. Silver, G. Kantor, I. M. Rekleitis, and H. Choset. *The hierarchical atlas.* IEEE Transactions on Robotics, 21(3) :473–481, 2005 (Cited on page 17.)

[Lovegrove et Davison 2010] Lovegrove, S. et Davison, A. *Real-time spherical mosaicing using whole image alignment.* In European Conference on Computer Vision, vol. 6313, 73–86, 2010 (Cited on page 33.)

[Lowe 2004] Lowe, David G. *Distinctive Image Features from Scale-Invariant Keypoints* Int. J. Comput. Vision, volume 60, November 2004 (Cited on page 99.)

[Lu 1997] F. Lu and E. Milios *Globally consistent range scan alignment for environment mapping.* Autonomous Robots, 4 :333–349, 1997. (Cited on page 12.)

[Magnusson 2007] Magnusson M, Duckett T, Lilienthal AJ *Scan registration for autonomous mining vehicles using 3D-NDT.* Journal of Field Robotics 24(10) :803–827 2007 (Cited on page 13.)

[Martin 2002] C. Martin and S. Thrun. *Online acquisition of compact volumetric maps with mobile robots.* International Conference on Robotics and Automation (ICRA), Washington, DC, 2002. ICRA. (Cited on page 13.)

[Matuszek 2013] Matuszek, Cynthia and Fox, Dieter and Koscher, Karl *Following directions using statistical machine translation* International conference on Human-robot interaction, p251–258, 2010, IEEE Press (Cited on page 116.)

[Meagher 1982] Meagher D *Geometric modeling using octree encoding.* Computer Graphics and Image Processing 19(2) :129–147 1982 (Cited on page 11.)

[Mei et Rives 2007] Mei, C. et Rives, P. (2007). *Single view point omnidirectional camera calibration from planar grids* In IEEE International Conference on Robotics and Automation, 3945–3950, 43, 44 (Cited on pages 33 and 34.)

[Meilland 2011] Meilland, M., Comport, A. I. and Rives, P. *Dense visual mapping of large scale environments for real-time localisation.* In International Conference on Intelligent Robots and Systems. San Francisco, California, 2011. (Cited on pages 17, 18, 25, 29, 33 and 95.)

[Meilland 2010] Meilland, M., Comport, A. I. et Rives, P. *A Spherical Robot-Centered Representation for Urban Navigation* In IEEE/RSJ International Conference on Intelligent Robots and System, 2010, Taipei, Taiwan. (Cited on page 33.)

[Meilland 2012] Meilland, M *Cartographie RGB-D dense pour la localisation visuelle temps-réel et la navigation autonome.* Ph.d Thesis, Ecole des mines de paris. (Cited on pages 5 and 37.)

[Mezouar et Chaumette 2003] Mezouar, Y. et Chaumette, F. *Optimal camera trajectory with image-based control* International Journal of Robotics Research, 22, 781–804, 2003. (Cited on page 29.)

[Miksik 2013] Ondrej Miksik and Daniel Munoz and J. Andrew Bagnell and Martial Hebert *Efficient Temporal Consistency for Streaming Video Scene Analysis* IEEE International Conference on Robotics and Automation (ICRA) 2013 (Cited on pages 52, 54, 56 and 58.)

[Milford et al. 2004] Milford, M.J. et Wyeth, G.F. et Prasser, D. *RatSLAM : a hippocampal model for simultaneous localization and mapping* Proceedings. ICRA 2004, IEEE International Conference on (Cited on page 131.)

[Milford 2010] Milford, Michael and Wyeth, Gordon *Persistent navigation and mapping using a biologically inspired SLAM system* The International Journal of Robotics Research, 2010, Sage Publications (Cited on page 73.)

[Mirowski 2009] Piotr Mirowski, Yann LeCun *Factor Graphs for Time Series Modeling* Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, 2009-01-01 (Cited on page 44.)

[Moravec 1994] H.P. Moravec and M.C. Martin. *Robot navigation by 3D spatial evidence grids* Mobile Robot Laboratory, Robotics Institute, Carnegie Mellon University, 1994 (Cited on page 11.)

[Munoz 2013] Daniel Munoz *Inference Machines : Parsing Scenes via Iterated Predictions* The Robotics Institute, Carnegie Mellon University, Juin 2013 (Cited on page 44.)

[Nayar 1997] Nayar, S. (1997). *Catadioptric omnidirectional camera.* In IEEE International Conference on Computer Vision and Pattern Recognition, 482–. 43 (Cited on page 33.)

[Nilson 1969] N. J. Nilsson. *A mobile automation : An application of artificial intelligence techniques.* In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI), pages 509–520, 1969. (Cited on page 17.)

[Newcombe 2011] Newcombe R, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison A, Kohli P, Shotton J, Hodges S, Fitzgibbon A *KinectFusion : Real-time dense surface mapping and tracking.* Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on, IEEE (Cited on page 13.)

[Nuchter et Hertzerg 2008] Nüchter, Andreas et Hertzberg, Joachim. *Towards semantic maps for mobile robots.* Robotics and Autonomous Systems, 2008, vol. 56, no 11, p. 915-926. (Cited on pages 22, 23 and 24.)

[O'Callaghan 2012] O'Callaghan, Simon and Ramos, Fabio *Gaussian process occupancy maps* Int. Journal of Robotics Research (Cited on page 83.)

[Papadakis 2014] Papadakis, P. ; Rives, P. ; Spalanzani, A. *Adaptive spacing in human-robot interactions* Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on , vol., no., pp.2627,2632, 14-18 Sept. 2014 (Cited on page 85.)

[Pfaff 2007] Pfaff P, Triebel R, Stachniss C, Lamon P, Burgard W, Siegwart R *Towards mapping of cities.* IEEE Int. Conf. on Robotics and Automation (ICRA), 2007, Rome, Italy (Cited on page 10.)

[Peter 2010] Peter Henry and Michael Krainin and Evan Herbst and Xiaofeng Ren and Dieter Fox *Rgbd mapping : Using depth cameras for dense 3d modeling of indoor environments* RGB-D : Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS, 2010 (Cited on page 13.)

[Posada 2014] Posada, Luis Felipe ; Hoffmann, Frank ; Bertram, Torsten *Visual Semantic Robot Navigation in Indoor Environments* ISR/Robotik 2014 ; 41st International Symposium on Robotics ; Proceedings of , vol., no., pp.1,7, 2-3 June 2014 (Cited on pages 24 and 116.)

[Pronobis et Jensfelt 2012] Pronobis, A., and Jensfelt, P. (2012, May). *Large-scale semantic mapping and reasoning with heterogeneous modalities.* In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 3515-3522). IEEE Systems, 56(6), 522-537. (Cited on pages 22, 44 and 83.)

[Ranganathan 2007] Ranganathan, A., and Dellaert, F. (2007, June). *Semantic modeling of places using objects.* In Proceedings of the 2007 Robotics : Science and Systems Conference (Vol. 3, pp. 27-30). (Cited on page 22.)

[Royer 2005] Royer, E., Lhuillier, M., Dhome, M. et Chateau, T. *Localization in urban environments : Monocular vision compared to a differential gps sensor* In IEEE International Conference on Computer Vision and Pattern Recognition, 114-121, 2005 (Cited on page 29.)

[Salakhutdinov 2009] Salakhutdinov, R., et Hinton, G. E. *Deep boltzmann machines* In International Conference on Artificial Intelligence and Statistics (pp. 448-455), 2009 (Cited on page 44.)

[Salas-Moreno 2013] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, *Slam++ : Simultaneous localisation and mapping at the level of objects* IEEE Conference on Computer Vision and Pattern Recognition, vol. 0, 2013 (Cited on page 100.)

[Shatkay 2002] H.Shatkay and L.P.Kaelbling. *Learning geometrically-constrained Hidden Markov Models for robot navigation : Bridging the topological-geometrical gap.* Journal of Artificial Intelligence Research,16 :167-207,2002. (Cited on page 15.)

[Shotton 2011] Shotton, J. and Fitzgibbon, A. and Cook, M. and Sharp, T. and Finocchio, M. and Moore, R. and Kipman, A. and Blake, A. *Real-time human pose recognition in parts from single depth images* Int. Conf. on Computer Vision and Pattern Recognition, 2011 (Cited on page 89.)

[Siemiatkowska 2011] B. Siemiatkowska and J. Szklarski and M. Gnatowski, *Mobile robot navigation with the use of semantic map constructed from 3D laser range scans* Control and Cybernetics, Vol. 40, no 2, 2011 (Cited on page 24.)

[Stoyanov 2010] Stoyanov T, Magnusson M, Andreasson H, Lilienthal AJ *Path planning in 3d environments using the normal distributions transform.* IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 2010 (Cited on page 14.)

[Tardos 2012] Galvez-Loópez, D. and Tardos, J.D. *Bags of Binary Words for Fast Place Recognition in Image Sequences* Robotics, IEEE Transactions on, 2012 (Cited on pages 99, 104 and 105.)

[Tomatis 2002] N.Tomatis, I. Nourbakhsh, and R.Siegwart *Hybrid simultaneous localization and map building : clossing the loop with with multi-hypothesis tracking.* In Proceedings of the IEEE International Conference on Robotics and Automation,May 2002. (Cited on page 17.)

[Thrun 1999] S. Thrun *Learning metric-topological maps for indoor mobile robot navigation* AI Journal, 21-71, 1999 (Cited on page 17.)

[Thrun 2000] S. Thrun, W. Burgard, and D. Fox. *A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping* IEEE International Conference on Robotics and Automation (ICRA-2000), 2000. (Cited on page 12.)

[Ulh 2011] Uhl, K., Roennau, A., and Dillmann, R. From Structure to Actions : Semantic Navigation Planning in Office Environments, IROS 2011 Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment (Cited on page 24.)

[Ullah 2008] Ullah, M.M. ; Pronobis, A. ; Caputo, B. ; Luo, J. ; Jensfelt, R. ; Christensen, H.I., *Towards robust place recognition for robot localization* Robotics and Automation, ICRA 2008. (Cited on page 15.)

[Ulrich 2000] Ulrich, I. ; Nourbakhsh, I., *Appearance-based place recognition for topological localization* International Conference on Robotics and Automation, 2000 (Cited on page 15.)

[Vasudevan 2008] Vasudevan, S., and Siegwart, R. . *Bayesian space conceptualization and place classification for semantic maps in mobile robotics.* Robotics and Autonomous Systems 2008. (Cited on pages 22, 24 and 100.)

[Wichert 1998] G. Von Wichert. *Mobile robot localization using a self-organised visual environment representation.* Robotics and Autonomous Systems, 1998. (Cited on page 15.)

[Wojek 2008] C. Wojek and B. Schiele, *A dynamic conditional random field model for joint labeling of object and scene classes* European Conference on Computer Vision (ECCV), October 2008. (Cited on page 52.)

[Wolf 2005] Wolf, Denis F and Sukhatme, Gaurav S *Mobile robot simultaneous localization and mapping in dynamic environments* Autonomous Robots, 2005, Springer (Cited on page 73.)

[Xiao 2009] J. Xiao and L. Quan, *Multiple view semantic segmentation for street view images* ICCV 2009 (Cited on page 52.)

[Xiong 2010] Xuehan Xiong and Daniel Huber, *Using Context to Create Semantic 3D Models of Indoor Environments* Proceedings of the British Machine Vision Conference (BMVC), 2010 (Cited on page 43.)

[Yagi 1995] Y.Yagi, Y.Nishizawa, and M.Yachida, *Map-based navigation for a mobile robot with omnidirectional image sensor copis* Robotics and Automation, IEEE Transactions on, vol.11, no.5, Oct 1995. (Cited on page 99.)

[Yamauchi 1996] B. Yamauchi and R. Beer. *Spatial learning for navigation in dynamic environments.* Systems, Man, and Cybernetics, Special Issue on Learning Autonomous Robots, 1996. (Cited on page 15.)

[Yi 2008] Yi, Chuho and Suh, Il Hong and Lim, Gi Hyun and Jeong, Seungdo and Choi, Byung-Uk *Cognitive Representation and Bayeisan Model of Spatial Object Contexts for Robot Localization* Proceedings of the 15th International Conference on Advances in Neuro-information Processing - Volume Part I, ICONIP'08, 2008 (Cited on pages 24 and 100.)

Titre de la Thèse

Résumé :

Texte.

Mots-clés : MC1, MC2...

Thesis Title

Abstract :

Text.

Keywords : KW1, KW2...