THÈSE de DOCTORAT de L'UNIVERSITÉ PARIS 6

Spécialité : **Informatique**

Présentée par : **David FILLIAT**

pour obtenir le grade de DOCTEUR de L'UNIVERSITÉ PARIS 6

Cartographie et estimation globale de la position pour un robot mobile autonome

Soutenue le 21 décembre 2001

devant le jury composé de :

Pr. Patrick Gallinari	(examinateur)
Pr. Wulfram Gerstner	(rapporteur)
Dr. Jean-Paul Laumond	(examinateur)
Dr. Dominique Luzeaux	(rapporteur)
Dr. Jean-Arcady Meyer	(directeur de thèse)

Résumé

La gestion des déplacements dans l'espace pour un robot mobile autonome est un problème qui a été abordé dès les premiers essais d'intégration de l'intelligence artificielle et de la robotique, mais qui reste difficile et auquel peu de solutions générales ont été apportées. Parmi toutes les stratégies de navigation existantes, nous nous intéressons à celles qui utilisent une carte représentant la structure spatiale de l'environnement et qui permettent de planifier les déplacements jusqu'à un but lointain. La construction d'une carte et l'estimation de la position du robot sont deux sous-problèmes de navigation qui ont reçu isolément des solutions efficaces. Toutefois, leur résolution simultanée reste une tâche complexe.

Le problème de l'estimation de la position, pris isolément, peut notamment être résolu de manière très efficace par des modèles probabilistes utilisant des Processus de Décision Markoviens Partiellement Observables. Toutefois, ces modèles ne peuvent en général pas être utilisés pour construire "en-ligne" la carte qu'ils utilisent.

Nous avons élaboré un modèle de navigation qui s'inspire de ces méthodes mais qui, à partir de capteurs relativement imprécis, permet la construction "en-ligne" de la carte. La capacité d'intégration des informations qu'offre cette méthode de localisation permet de compenser la faiblesse des perceptions. Nous utilisons de plus des procédures de perception active qui permettent, en fonction du contexte, d'utiliser les capteurs de manière efficace. Notre modèle permet ainsi, en faisant peu d'hypothèses sur l'environnement, une estimation robuste de la position et une construction simultanée de la carte. Ses capacités ont été démontrées à la fois en simulation et sur un robot réel. En particulier, nous avons montré qu'il permettait au robot de se localiser correctement dès son introduction dans un environnement nouveau, de se relocaliser rapidement lorsqu'il est transporté passivement d'un endroit à un autre et de planifier une trajectoire permettant de rejoindre de manière robuste un but donné.

Mots-clés: navigation, animat, localisation globale, cartographie, planification

Abstract

Managing the movements of an autonomous mobile robot in its environment is a problem that has been tackled since the early integration of artificial intelligence and robotics. However, this problem remains difficult and no general solution has been devised. Among existing navigation strategies, we will focus on those that use a map to represent the spatial layout of the environment and that allow to plan movements toward distant goals. Map-building and self-positioning within these maps are two sub-problems that have been solved independently. However, solving these sub-problems simultaneously is still a difficult task.

In particular, position estimation decoupled from map-building can efficiently be realized by probabilistic models based on Partially Observable Markov Decision Processes. These models however cannot generally be used to build the map they use on-line.

We have designed a navigation model inspired from these localization methods that is able to build a map on-line using only relatively imprecise sensors. The capacity to integrate various information sources afforded by this method makes it possible to compensate the low quality of a single perception. Moreover, active perception procedures allow to efficiently use available sensors with regard to the current context. Thus, our model allows a robust position estimation along with simultaneous map-building with few hypotheses on the environment. Its capacities have been demonstrated in simulation and on a real robot. In particular, we have shown that it allows a robot to localize itself correctly as soon at it is introduced in a new environment, to rapidly relocalize itself if it is passively transported from one place to another, and to plan a reliable trajectory to a distant goal.

Keywords: navigation, animat, global localization, map-learning, path-planning

Remerciements

Je souhaite tout d'abord remercier les membres du Jury qui ont accepté de m'honorer de leur présence lors de la soutenance de cette thèse.

Je remercie tout d'abord Messieurs Patrick Gallinari et Jean-Paul Laumond de faire partie de mon jury malgé leurs nombreuses contraintes professionnelles.

Merci également à Monsieur Wulfram Gerstner, dont les articles m'ont apporté une bonne part des connaissances à la base de cette thèse et qui a accepté d'en être rapporteur.

Je tiens à remercier Monsieur Dominique Luzeaux qui a également accepté d'être rapporteur de ma thèse alors qu'il soutenait lui-même son habilitation à diriger des recherches quelques jours plus tôt. Je lui suis également reconnaissant de la disponibilité dont il a fait preuve pour me permettre de découvrir en douceur le milieu professionel vers lequel je vais me diriger après cette thèse.

Je dois également de profonds remerciements à mon directeur de thèse, Monsieur Jean-Arcady Meyer, pour m'avoir toujours donné la liberté de conduire mon travail à mon aise et pour m'avoir toujours aidé, conseillé et encouragé. Travailler au sein de son équipe a été un plaisir dont je garderai un souvenir marquant.

Cette thèse ne se serait pas aussi bien déroulée sans toutes les personnes qui font fonctionner le laboratoire au quotidien. Je dois en particulier remercier Jacqueline Le Baquer et Ghislaine Mary qui nous ont beaucoup aidé lors de l'arrivée de l'équipe de l'AnimatLab au sein du Pôle IA. Merci également à Valérie Mangin, Nicole Nardy, Annick Omnès-Ravier, Andrée Musial, Marie-Claire Masbou, Jean-Pierre Arranz et Christophe Bouder pour leur gestion efficace de tous nos soucis administratifs et informatiques. Je souhaite également remercier Patrick Gallinari et Jean-Pierre Briot, qui dirigent le Pôle IA et le thème OASIS dont l'AnimatLab dépend au sein du LIP6, pour leur action dont dépendent les bonnes conditions de travail dont nous bénéficions.

L'équipe de l'AnimatLab était un peu dégarnie quand j'ai commencé ma thèse puisque nous n'étions que deux thésards. Elle constitue, maintenant, une équipe chaleureuse au sein de laquelle il est très agréable de travailler, discuter et se détendre. Je dois pour cela remercier tous les membres qui en font ou en ont fait partie. Merci donc à Agnès, Olivier, Asli, Benoit, Vincent, Gouri, Gabriel, Gildas, Pierre, Stéphane et Fabien.

Même s'ils ont le défaut de ne pas faire partie de l'AnimatLab, cet ambiance est aussi due à Louis, Samuel, François, Nicolas, Valérie, Bruno et Jean-Christophe. Merci à eux aussi.

Mon arrivée au sein de l'AnimatLab remonte à mon stage de DEA, qui était proposé par Jérome Kodjabachian qui faisait à l'epoque partie de l'équipe. Le choix de ce stage découle en grande partie des conseils de Jean-Daniel Zucker et Alexis Drogoul qui enseignent dans le DEA. Je les remercie tous trois de m'avoir aidé à trouver ma voie.

Cette voie avait été esquissée à l'école Polytechnique grâce à Jean-Christophe auquel je dois mon intérêt pour l'intelligence artficielle. Son amitié durant le DEA et durant cette thèse ont souvent été d'une grande importance.

Enfin, au delà de cette thèse, ma famille m'a toujours permis d'avancer dans ma vie et dans mes projets. J'attends toujours avec impatience mes retours dans mon Auvergne natale où je peux me ressourcer parmi les personnes que je connais depuis toujours et qui me réservent chaque fois un accueil merveilleux. Je les en remercie tous.

Je dois notamment beaucoup de choses à mes parents qui ont toujours soutenu mes choix et m'ont donné la possibilité de faire ce que je souhaitais. Ma sœur a également toujours été à mes côtés. Je lui dois des moments inoubliables. Que cette thèse soit encore une fois l'occasion de les en remercier.

Je termine ces remerciements par Gaëlle qui est depuis peu devenue ma femme et qui supporte au jour le jour mes angoisses et mes humeurs. C'est sa présence, plus que tout autre réconfort, qui m'aide à avancer au quotidien. Je l'en remercie infiniment.

Table des matières

Introduction **1**

I Problématique **7**

1 La navigation par cartes **9**

1.1 Cartographie, localisation et planification 9

1.2 Quelques hypothèses de travail 10

1.2.1 Estimation de la position et de la direction 10

1.2.2 Environnements statiques et dynamiques 11

1.3 Sources d'information disponibles 12

1.3.1 Informations proprioceptives 13

1.3.2 Informations extéroceptives 13

1.4 Représentations en robotique 21

1.4.1 Cartes topologiques 22

1.4.2 Cartes métriques 27

1.4.3 Fusion de capteurs 31

1.5 Données biologiques sur la navigation 32

1.5.1 Les cellules de lieu et de direction de la tête 32

1.5.2 Informations utilisées 33

1.5.3 Représentations biologiquement plausibles 33

1.6 Conclusion 34

2 Localisation **37**

2.1 Différentes capacités de localisation 37

2.2 Stratégies de localisation 39

2.2.1 Estimation de la position par les données allothétiques 39

2.2.2 Suivi d'une hypothèse unique 42

2.2.3 Suivi de plusieurs hypothèses 45

2.3 Les modèles markoviens pour la localisation 47
2.4 Perception active 50
2.5 Conclusion 51

3 Cartographie **53**
3.1 Les problèmes de la cartographie 53
3.1.1 Limitation des méthodes de localisation 53
3.1.2 Cohérence de la carte 54
3.1.3 Cartographie incrémentale et retour en arrière 57
3.2 Cartographie incrémentale 59
3.2.1 Estimation de la position 59
3.2.2 Modifications de la carte 60
3.2.3 Stratégies d'exploration 61
3.3 Retour sur les modifications passées 61
3.3.1 Découplage des données idiothétiques et allothétiques 62
3.3.2 Cartographie «hors-ligne» 63
3.3.3 Cartographie «en-ligne» 65
3.4 Conclusion 67

4 Planification **69**
4.1 Exécution d'un plan 69
4.2 Discrétisation de l'espace de recherche 70
4.3 Planification 73
4.3.1 Deux types de plan 73
4.3.2 Choix de l'action avec une position incertaine 74
4.4 Conclusion 75

II Notre modèle **77**

5 Notre modèle **79**
5.1 Choix stratégiques 79
5.2 Structure du modèle 80
5.2.1 Données allothétiques 81
5.2.2 Données idiothétiques 83
5.3 Estimation de la position 83
5.3.1 Choix des lieux à représenter 83
5.3.2 Intégration des données idiothétiques 84

5.3.3 Intégration des données allothétiques omnidirectionnelles 88
5.3.4 Intégration des données allothétiques directionnelles 89
5.3.5 Estimation de la position par les données idiothétiques 90
5.3.6 Reconnaissance de la position 92
5.4 Mise à jour de la carte 94
5.4.1 Mise à jour des nœuds 94
5.4.2 Mise à jour des arêtes 94
5.4.3 Nettoyage de la carte 97
5.5 Algorithme d'exploration 98
5.6 Planification 99
5.6.1 Calcul de la politique 99
5.6.2 Choix de la direction 101
5.6.3 Réaction aux obstacles imprévus 103
5.7 Choix de la direction des capteurs directionnels 103

6 Expérimentations **107**
6.1 Plateforme expérimentale 107
6.1.1 Capteurs omnidirectionnels 108
6.1.2 Capteurs directionnels 109
6.1.3 Capteurs odométriques 110
6.1.4 Architecture logicielle 112
6.2 Expériences en simulation 113
6.2.1 Utilisation d'une caméra panoramique 113
6.2.2 Utilisation d'une caméra directionnelle 122
6.3 Expériences sur un robot réel 131
6.3.1 Cartographie 132
6.3.2 Relocalisation 135
6.3.3 Déplacement vers un but 139

7 Discussion **141**
7.1 Comparaison avec des modèles existants 141
7.1.1 Modèle d'Arléo et Gerstner 141
7.1.2 Modèle de Thrun et al. 144
7.1.3 Résumé 147
7.2 Bilan 147
7.3 Perspectives 149
7.3.1 Estimation de la direction 149

7.3.2 Navigation active 150
7.3.3 Planification 151
7.3.4 Fonctionnement des systèmes biologiques 152
7.3.5 Intégration dans un système de navigation plus général 153

Conclusion **155**

A Paramètres de notre modèle **157**

Bibliographie **159**

Introduction

Agir dans son environnement

L'apparition de la capacité de contrôler ses déplacements constitue un progrès considérable dans l'histoire de la vie. La possibilité de se déplacer permet en effet à un être vivant de répondre de manière beaucoup plus efficace aux sollicitations de son environnement. Il devient capable de rechercher l'énergie qui lui est vitale, ou de fuir un danger qui lui serait mortel s'il ne pouvait l'éviter. Cette mobilité constitue potentiellement un avantage énorme si elle est correctement exploitée. Or, l'évolution biologique s'est chargée de faire apparaître des systèmes capables de développer des stratégies d'une grande complexité autour de cette capacité. Ces stratégies sont extrêmement variées, d'une espèce animale à une autre, mais constituent toujours une base fondamentale pour la survie des individus.

En comparaison avec les capacités de déplacement d'animaux tels que le rat, les capacités actuelles des robots mobiles font pâle figure. Pour eux aussi, cependant, la capacité à se déplacer correctement dans leur environnement est cruciale et leur utilité dans nombre de tâches est subordonnée à cette capacité. Cependant, les robots actuels, s'ils sont évidemment capables de se déplacer, ne sont néanmoins pas totalement autonomes de ce point de vue. Même les plus sophistiqués d'entre eux, qui fonctionnent dans des environnements complexes nécessitent une intervention humaine à un moment ou un autre pour assurer leur mission.

Le travail de thèse présenté ici est un travail de robotique qui s'inspire, au niveau conceptuel, de données biologiques et s'inscrit dans le cadre de l'*approche Animat* [Meyer, 1995, Meyer, 1996, Meyer, 1997]. Un *animat* est un animal simulé sur ordinateur ou un robot dont le fonctionnement s'inspire de données biologiques sur les animaux et ce terme sera employé tout au long de cette thèse pour désigner l'entité sur laquelle nous travaillons, que ce soit la simulation d'un animal ou un robot, réel ou simulé.

L'ensemble des capacités permettant aux animats de se déplacer et de rejoindre un but donné sont regroupées sous le terme de navigation. De très nombreuses stratégies mettant en œuvre de telles capacités ont été explorées ou exploitées, par les ingénieurs comme par les biologistes. Cette thèse présente une étude sur une petite partie de l'ensemble de ces stratégies, puisqu'elle s'attache uniquement à la navigation par cartes pour laquelle un modèle interne du monde extérieur est utilisé par l'animat pour se déplacer.

Différents types de navigation

Les stratégies de navigation à base de cartes sont extrêmement diverses, de même que les classifications qui peuvent en être faites. Aussi, afin de situer ce type de navigation dans son contexte général, nous reprenons ici une classification établie par Trullier et al. [Trullier and Meyer, 1997, Trullier et al., 1997], laquelle présente l'avantage de distinguer les stratégies sans modèles internes et les stratégies avec modèle interne. Cette classification comporte cinq catégories, de la plus simple à la plus complexe :

- **Approche d'un objet :** cette capacité de base permet de se diriger vers un objet visible depuis la position courante de l'animat. Elle est en général implémentée par une remontée de gradient basée sur la perception de l'objet [Braitenberg, 1984]. Cette stratégie utilise des *actions réflexes*, dans lesquelles chaque perception est directement associée à une action. C'est une stratégie locale, c'est-à-dire fonctionnelle dans la zone de l'environnement pour laquelle le but est visible.
- **Guidage :** cette capacité permet d'atteindre un but qui n'est pas un objet directement visible. Ce but est alors caractérisé par la configuration spatiale d'un ensemble d'objets remarquables, ou *amers*, qui l'entourent ou qui en sont voisins. La stratégie de navigation consiste alors à se diriger dans la direction qui permet de reproduire cette configuration. Cette capacité semble utilisée par certains insectes [Cartwright and Collett, 1987] et a déjà été implémentée sur des robots [Gaussier et al., 2000, Lambrinos et al., 2000, Gourichon and Meyer, 2001]. Cette stratégie utilise également des actions réflexes et implémente une navigation locale qui requiert que les amers caractérisant le but soient visibles.

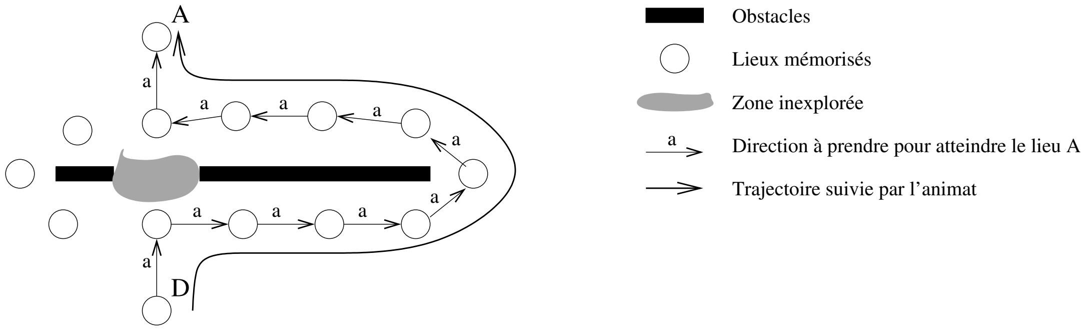


Fig. *1: Action associée à un lieu. En chaque lieu représenté par un cercle, l'action à accomplir pour rejoindre le but A est représentée par une flèche indiquant la direction à suivre à partir de ce lieu. Cette stratégie permet de rejoindre un but distant dans l'environnement mais repose sur des chemins figés. Dans cet exemple, le chemin joignant le lieu D au lieu A et passant par la droite de l'obstacle a été appris. Rejoindre le lieu A depuis le lieu D en utilisant cette stratégie ne pourra alors être réalisé que par ce chemin. Le raccourci empruntant le chemin de gauche est inutilisable.*

- **Action associée à un lieu :** cette capacité est la première capacité implémentant une navigation globale, c'est-à-dire qui permette de rejoindre un but depuis des positions pour lesquelles ce but ou les amers qui caractérisent son emplacement sont invisibles. Elle requiert une représentation interne de l'environnement qui consiste à définir des lieux comme des zones de l'espace dans lesquelles les perceptions restent similaires, et à associer une action à effectuer à chacun de ces lieux (cf. figure 1). L'enchaînement des actions associées à chacun des lieux reconnus définit une *route* qui permet de rejoindre le but. Ces modèles permettent donc une autonomie plus importante mais sont limités à un but fixé. Une route qui permet de rejoindre un but ne pourra en effet pas être utilisée pour rejoindre un but différent. Changer de but entraînera l'apprentissage d'une nouvelle route, indépendante des routes permettant de rejoindre les autres buts.

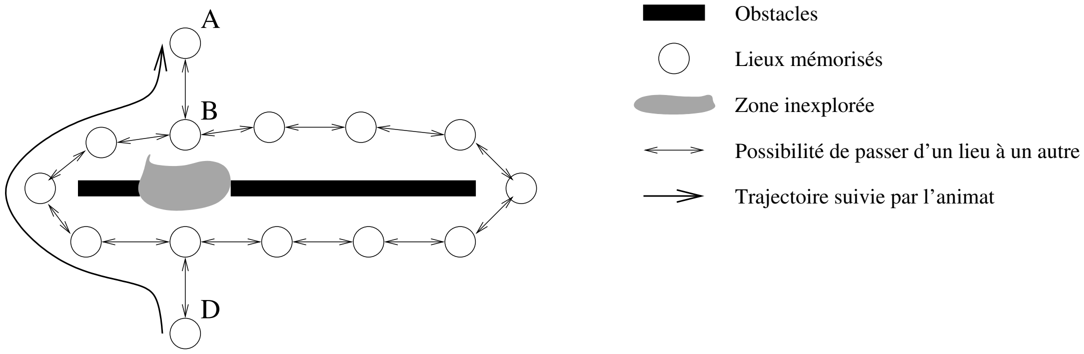


FIG. *2: Navigation topologique. Cette stratégie permet de mémoriser un ensemble de lieux et les possibilités de passer de l'un à l'autre, indépendamment de tout but. Dans notre exemple, le chemin le plus court entre D et A peut alors être calculé, mais uniquement parmi les lieux et les chemins déjà connus. Cette stratégie permet ici de contourner l'obstacle par la gauche mais ne permet pas de le traverser en ligne droite de D à A.*

- **Navigation topologique :** cette capacité est une extension de la précédente qui mémorise dans le modèle interne les relations spatiales entre les différents lieux. Ces relations indiquent la possibilité de se déplacer d'un lieu à un autre, mais ne sont plus associées à un but particulier. Ainsi le modèle interne est un graphe qui permet de calculer différents chemins entre deux lieux arbitraires. Ce modèle ne permet toutefois que la planification de déplacements parmi les lieux connus et suivant les chemins connus (cf. figure 2).
- **Navigation métrique :** cette capacité est une extension de la précédente car elle permet à l'animat de planifier des chemins au sein de zones inexplorées de son environnement. Elle mémorise pour cela les positions métriques relatives des différents lieux, en plus de la possibilité de passer de l'un à l'autre. Ces positions relatives permettent, par simple composition de vecteurs, de calculer une trajectoire allant d'un lieu à un autre, même si la possibilité de ce déplacement n'a pas été mémorisée sous forme d'un lien (cf. figure 3).

Les modèles des deux dernières catégories autorisent donc une navigation globale et permettent de rejoindre un but arbitraire au sein de l'environnement. Ils s'appuient pour cela sur un modèle interne du monde qui supporte une *planification*. Ce modèle interne mémorise donc la structure spatiale de l'environnement, indépendamment d'un but précis. Chacune des positions

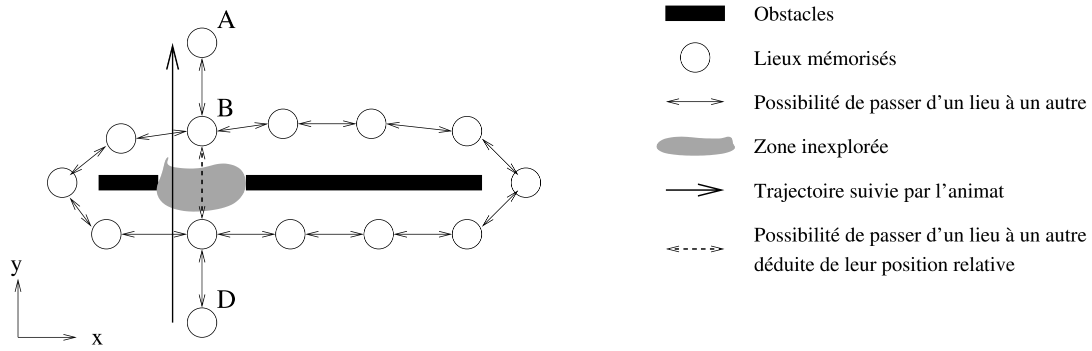


FIG. *3: Navigation métrique. Cette stratégie permet de calculer le chemin le plus court entre deux lieux mémorisés, permettant même de planifier des raccourcis au sein de zones inexplorées de l'environnement. Pour cela, la carte mémorise la position métrique relative de chacun des lieux visités par l'animat. Ainsi il est possible de prévoir un déplacement entre deux lieux, même si la possibilité de ce déplacement n'est pas enregistrée dans la carte. Dans cet exemple, cette stratégie permet de d'aller du lieu A au lieu D en traversant la zone inexplorée.*

mémorisées dans ce modèle interne peut alors être utilisée comme but par le processus de planification dont le rôle est de calculer une route vers ce but. Ces représentations internes, supportant un processus de planification, se regroupent sous l'appellation de *carte cognitive* [Tolman, 1948]. Ce sont ces deux stratégies qui sont regroupées sous le terme de navigation par carte.

D'un point de vue robotique, l'utilisation de cartes semble être une méthode indispensable pour permettre à un robot mobile d'effectuer des tâches de navigation dans toutes les conditions environnementales. La construction et l'utilisation de telles cartes posent cependant de nombreux problèmes, notamment pour garantir l'adéquation entre la carte et le monde réel. En réponse à ces problèmes, la suppression de toute forme de modèle de l'environnement proposée par Brooks [Brooks, 1991], semble être trop radicale. L'utilisation de modèles internes de l'environnement, associée à un processus permettant de prévoir des déplacement en se basant sur une telle représentation semble indispensable à la plupart des robots actuels.

Une telle représentation est naturelle pour les êtres humains, pour lesquels des processus cognitifs de haut niveau sont utilisés pour mettre en correspondance la carte avec l'environnement réel. De plus, cette mise en correspondance est souvent simplifiée par des aménagements particuliers de l'environnement, par exemple celui qui consiste à nommer les rues. Ces processus de haut niveau sont toutefois difficile à implémenter pour un robot réel qui ne dispose que de systèmes rudimentaires de perception et de traitement des informations. Cependant, des preuves de l'existence de représentations internes similaires à de telles cartes ont été trouvées chez les rats. Ces représentations sont identifiables au niveau neurologiques dans certaines parties de leur cerveau. Cela permet de supposer que l'utilisation de cartes peut être réalisée par des êtres vivants, sans le support de concept abstraits tels que les utilisent les humains. Ce type de carte qui fait appel à des structures neurologiques de base est donc un paradigme intéressant pour les robots mobiles.

Notre thèse se focalisera ainsi sur les modèles relevant des deux dernières stratégies de navigation évoquées plus haut, en s'inspirant à la fois des réalisations robotiques actuelles et des informations dont nous disposons sur la manière dont les animaux paraissent utiliser des cartes de leur environnement.

Structure du mémoire

La première partie de ce mémoire présente en détail la problématique abordée, ainsi que l'état de l'art dans le domaine de la navigation en robotique.

Au sein de cette partie, le chapitre 1 présente en premier lieu la navigation basée sur des cartes, ainsi que les sources d'informations disponibles pour accomplir cette tâche. Nous verrons que les problèmes les plus importants surviennent lorsque l'on cherche à résoudre simultanément le problème de l'estimation de la position au sein d'une carte (*localisation*) et le problème de la construction de cette carte (*cartographie*). Nous verrons comment les cartes utilisées en robotique peuvent organiser les différentes informations disponibles et nous présenterons un rapide aperçu de la manière dont la tâche de navigation est traitée par le rat.

Le chapitre 2 présente isolément le problème de la localisation, en considérant qu'une carte complète de l'environnement est disponible. Nous décrirons les différentes stratégies permettant l'estimation de la position d'un animat. Nous verrons notamment que les stratégies les plus performantes ne fonctionnent que dans le cas de cartes complètes de l'environnement. Plus de détails sur les modèles mentionnés sont disponibles dans une publication [Filliat and Meyer, 2001a]. Nous nous attacherons dans cette thèse à étendre ces méthodes au cas des cartes incomplètes, en cours de construction.

Le chapitre 3 présente le problème de la cartographie. Nous verrons comment l'incomplétude de la carte au cours de la cartographie restreint les méthodes de localisation utilisables. Nous verrons également qu'il existe différentes méthodes de cartographie qui utilisent de manière plus ou moins efficace les données qui ont été recueillies par l'animat. Les différents modèles mentionnés sont détaillés dans une publication [Filliat and Meyer, 2001b]. Nous nous attacherons dans cette thèse à adapter les plus performantes de ces méthodes afin de lever les restrictions qu'elles imposent aux méthodes de localisation.

Le chapitre 4 présente succinctement le problème du calcul d'un chemin pour atteindre un but (*planification*). Ce problème n'a pas été central dans notre thèse, aussi présentons-nous simplement les considérations qui nous ont mené à la méthode retenue dans notre modèle.

La seconde partie du mémoire présente le modèle de navigation que nous avons développé dans le but d'assurer à un robot la plus grande autonomie possible, ainsi que les validations expérimentales que nous avons réalisées.

Le chapitre 5 présente notre modèle de navigation et les réponses que nous avons apportées au problème de la localisation et de la cartographie simultanée. Notre modèle permet ainsi de construire une carte de l'environnement en utilisant une méthode de localisation globale qui suppose habituellement une carte complète. Nous présentons également la méthode de planification que nous avons implémentée pour utiliser la carte résultante.

Le chapitre 6 présente les tests réalisés sur notre modèle. Nous présentons une série d'expériences en simulation qui nous ont permis de valider notre modèle dans des environnements contrôlés. Nous présentons également des expériences réalisées avec un robot réel dans les couloirs de notre laboratoire.

Enfin, le chapitre 7 présente les avantages et les inconvénients de notre modèle. Pour cela, nous le comparons à deux modèles existants dont nous nous sommes inspirés. Nous présentons également quelques perspectives d'extension intéressantes de notre travail.

Première partie

Problématique

Chapitre 1

La navigation par cartes

Ce chapitre est consacré à la présentation de la problématique générale de la navigation basée sur une carte. Nous allons décrire dans un premier temps en quoi consiste cette méthode de navigation et quelles sont les informations disponibles pour la réaliser. Nous présenterons par la suite les différentes organisations possibles du modèle interne que constitue la carte dans les applications robotiques existantes. Enfin, nous terminerons ce chapitre par un aperçu de la manière dont un animal, le rat, peut réaliser une telle tâche.

1.1 Cartographie, localisation et planification

Le processus complet qui permet à un animat de mémoriser son environnement, puis de s'y déplacer pour rejoindre un but, peut être découpé en trois phases : la cartographie, la localisation et la planification. Ces trois phases permettent de répondre aux trois questions fondamentales pour la tâche de navigation [Levitt and Lawton, 1990] : Où suis-je ? Où sont les autres lieux par rapport à moi ? et Comment puis-je atteindre mon but ?

- La cartographie est la phase qui permet la construction d'une carte reflétant la structure spatiale de l'environnement à partir des différentes informations recueillies par l'animat.
- Une telle carte étant disponible, la localisation permet alors à l'animat de déterminer la position dans la carte qui correspond à sa position dans son environnement réel.
- La planification, enfin, est la phase qui permet à l'animat, connaissant la carte de l'environnement et sa position actuelle, de prévoir les mouvements à effectuer afin de rejoindre un but fixé dans l'environnement.

Ces trois phases sont évidemment fortement interdépendantes. L'ordre dans lequel elles sont citées fait directement apparaître le fait que la seconde phase dépend de la première. En effet, estimer sa position au sein d'une carte de l'environnement suppose implicitement que cette carte existe et qu'elle contient la position courante de l'animat. De même, la troisième phase dépend des deux premières, car la planification suppose que l'on connaisse sa position et que la carte de l'environnement représente une portion de l'environnement contenant au moins un chemin reliant cette position au but qui doit être atteint.

Mais la relation entre les deux premières phases est plus subtile qu'une simple relation d'antériorité. Cette relation est similaire à celle qui apparaît dans le problème de l'œuf et de la poule. Chacun des deux éléments peut, en effet, être considéré comme préalable à l'autre mais dépend aussi de l'autre pour sa réalisation. Dans le cas de la cartographie et de la localisation, nous avons déjà vu que la localisation repose sur une phase préalable de cartographie. Mais pour construire une carte, il est nécessaire de savoir où ajouter, dans la carte partielle déjà existante, toute nouvelle information recueillie par l'animat. Cela requiert donc une phase préalable de localisation au sein de la carte partielle déjà existante. Ainsi, pour un animat complètement autonome, il est impossible de ne pas traiter ces deux premiers problèmes simultanément.

Dans le cadre plus restreint de la robotique, où l'on autorise un opérateur humain à intervenir dans le processus, il est évidemment possible de découpler ces deux phases. Dans les applications de robotique réelle, il est fréquent que l'on fournisse au robot une carte construite au préalable et qu'on ne s'intéresse qu'à l'estimation de la position du robot au sein de cette carte pour qu'il puisse accomplir sa tâche. La carte peut alors être obtenue de différentes manières. Il est par exemple possible d'utiliser un plan d'architecte d'un bâtiment pour le transformer en une carte utilisable par le robot. Il est également possible d'utiliser le robot dans une phase supervisée de cartographie. Au cours de cette phase, la position du robot est calculée de manière précise par un dispositif externe au système de navigation, et ne nécessite donc pas que le système estime de lui-même la position. Connaissant la position précise du robot, il est alors relativement simple de construire une carte de l'environnement.

Dans le cadre de cette thèse, nous souhaitons éviter toute intervention de ce type. Notre but est donc la réalisation d'un système qui puisse réaliser de manière simultanée et autonome la cartographie et la localisation.

1.2 Quelques hypothèses de travail

1.2.1 Estimation de la position et de la direction

A ce stade, il convient de préciser la notion de position que nous emploierons dans cette thèse. En effet, la position d'un animat est définie à la fois par son emplacement spatial, estimé par rapport à un point de référence et par sa direction, estimée par rapport à une direction de référence. Ces deux quantités sont couplées mais ont des statuts relativement distincts en pratique.

Lors du mouvement, la direction de l'animat influence la manière dont varie sa position, mais peut souvent être contrôlée indépendamment. Ce découplage permet notamment de simplifier le processus de planification en ne tenant pas compte de la direction de l'animat, laquelle peut être contrôlée sans influencer la position. La variable importante est alors la position de l'animat, la direction devant être estimée pour pouvoir agir, mais non pour planifier.

Cette indépendance relative au niveau de la planification peut conduire à des systèmes d'estimation de la position et de la direction séparés. Cette séparation est supportée par le fait que la direction d'un animat peut être mesurée par des capteurs indépendamment de l'estimation de sa position. Il est par exemple possible d'utiliser une boussole qui mesure la direction par rapport à la direction du pôle magnétique, ou un gyroscope qui mesure la direction par rapport à une direction arbitraire fixe. De plus, d'un point de vue biologique, ces données semblent également découplées pour certains êtres vivants. Par exemple, la position et la direction de la tête d'un rat, paraissent être codées dans des parties différentes de son cerveau (cf. paragraphe 1.5).

Le choix de représenter et d'estimer de manière séparée la position et la direction n'interdit toutefois pas des interactions entre ces informations. L'estimation de la position utilisera évidemment celle de la direction pour pouvoir intégrer de nouvelles données lors du mouvement de l'animat. L'estimation de la direction pourra également dépendre de la position par un système de recalage qui utilisera la perception d'un point de référence connu depuis une position connue pour estimer la direction.

En conséquence de la relative indépendance entre la position et la direction, ainsi que de la plus grande importance de la position pour la planification, nous mettrons l'accent dans cette thèse sur l'estimation de la position, en supposant que nous disposons par ailleurs d'une estimation relativement fiable de la direction. Cette estimation sera fournie par des capteurs indépendants mais pourra toutefois être corrigée par notre système de navigation en cas de besoin.

1.2.2 Environnements statiques et dynamiques

Il convient également de préciser quel type d'environnements nous considérons dans cette thèse. En effet, les animats sont amenés à se déplacer dans une grande variété d'environnements qui peuvent être regroupés en deux grandes catégories : les environnements statiques et les environnements dynamiques. Les environnements statiques sont des environnements qui ne subissent pas de modifications au cours du temps. Cette stabilité concerne à la fois leur structure spatiale et leur apparence pour l'animat. Cela exclut la majorité des environnements dans lesquels les humains évoluent quotidiennement. Les environnement dynamiques, pour leur part, présentent des caractéristiques qui évoluent au cours du temps. La plupart des environnements courants appartiennent évidemment à la seconde catégorie. Par exemple, un environnement de bureau est dynamique, du fait des personnes qui y vivent, des chaises qui y sont déplacées ou des portes qui y sont ouvertes ou fermées.

Il est, de plus, possible de distinguer deux catégories d'éléments dynamiques. La première catégorie regroupe les éléments variables qui ne caractérisent pas l'environnement. De tels éléments peuvent être considérés comme du bruit qui n'a pas d'intérêt dans la modélisation de l'environnement pour la planification. C'est, par exemple, le cas des personnes évoluant dans un bureau, ou des chaises déplacées. Ces environnements peuvent être considérés comme constitués d'une partie statique sur laquelle se superpose différentes sources de bruit. La partie statique est la partie la plus importante à modéliser pour parvenir à une navigation efficace. Deux effets du bruit doivent toutefois être pris en compte. Il faut premièrement veiller à ce qu'il n'empêche pas la réalisation de commandes issues de la planification. Cela est en général réalisé par un système de contrôle, séparé du système de navigation, qui réalise l'interface avec la partie physique de l'animat. De plus, il faut prendre en compte ce bruit au niveau de la cartographie et de la localisation afin qu'il ne nuise pas à la modélisation de la seule partie statique de l'environnement et ne conduise pas à une mauvaise estimation de la position.

La seconde catégorie d'éléments dynamiques regroupe les éléments variables qui caractérisent l'environnement et peuvent avoir un intérêt pour la planification. C'est, par exemple, le cas des portes qui modifient la structure spatiale de l'environnement et peuvent entraîner des modifications de trajectoires en fonction de leur état. Ils doivent donc être enregistrés dans la carte si l'on veut pouvoir les prendre en compte.

La plupart des systèmes de navigation robotiques s'intéressent aux environnements appartenant à l'une des deux premières catégories. Les environnements sont donc supposés être soit statiques, soit entachés d'un bruit qui n'influence pas la planification. Ces systèmes s'intéressent donc à modéliser la partie statique des environnements qui va être utile pour la localisation et la planification. Il faut toutefois noter que ces systèmes, qui ne modélisent pas les éléments dynamiques de la seconde catégorie, sont néanmoins capable d'évoluer dans des environnements qui contiennent de tels éléments. Pour ce faire, ces systèmes sont en général capables de vérifier que la trajectoire planifiée est correctement exécutée. En cas de problème d'exécution, un chemin alternatif ne passant pas par la zone qui ne peut être atteinte est alors recherché. Cette méthode, qui ne modélise pas explicitement les portes, par exemple, est néanmoins capable de provoquer des détours si une porte fermée bloque un chemin. Nous nous attachons donc dans cette thèse aux systèmes de ce type uniquement.

1.3 Sources d'information disponibles

Pour naviguer, un animat dispose de deux types d'informations différentes : les informations proprioceptives et les informations extéroceptives.

- Les informations proprioceptives, également appelées idiothétiques, sont des informations internes à l'animat qui le renseignent, dans le cas de la navigation, sur son déplacement dans l'espace. Ces informations peuvent provenir, de la mesure de la rotation de ses roues, pour un robot, ou de la mesure de l'accélération grâce aux otolithes de son oreille interne, pour un mammifère. Un processus d'intégration permet alors, en accumulant ces informations au cours du temps, d'estimer la position relative de deux points par lesquels l'animat est passé.
- Les informations extéroceptives, encore appelées allothétiques ou plus simplement perceptives, sont des informations que l'animat peut acquérir sur son environnement. Ces informations peuvent être de nature très variée. Par exemple, un robot peut mesurer la distance des obstacles avec des capteurs infrarouges ou utiliser une caméra ; un animal peut utiliser sa vision, mais aussi son odorat pour percevoir son environnement.

Ces deux sources d'information ont des propriétés opposées que nous allons maintenant détailler.

1.3.1 Informations proprioceptives

Les informations proprioceptives considérées dans cette thèse renseignent sur le *déplacement* de l'animat dans l'espace. A ce titre, elles constituent une source d'information très importante pour la navigation. Cependant, la qualité de cette information se dégrade continuellement au cours du temps, la rendant inutilisable comme seule référence à long terme. Cette dégradation continuelle provient de l'intégration temporelle des mesures effectuées par les capteurs. En effet, chaque capteur produit une mesure bruitée de la vitesse ou de l'accélération de l'animat. Ce bruit, via le processus d'intégration qui a pour but d'estimer le déplacement, conduit inévitablement à une erreur croissante dans son estimation.

Il faut cependant noter que la plus grande partie de l'erreur se trouve souvent concentrée sur l'estimation de la direction de l'animat. Dans le cadre de cette thèse, qui suppose qu'une estimation correcte de la direction de l'animat est disponible par ailleurs, cette erreur est donc compensée. La qualité des informations proprioceptives résultant de cette compensation est cependant toujours insuffisante pour permettre une navigation à long terme.

Malgré ce défaut important, les informations proprioceptives ont l'avantage de dépendre assez peu des conditions environnementales qui perturbent fortement les informations perceptives. La vision, par exemple sera fortement perturbée si l'environnement est plongé dans le noir, mais les informations proprioceptives fourniront une information identique, que l'environnement soit éclairé ou non. De plus, comme nous le verrons dans la section suivante, si deux lieux identiques se trouvent dans l'environnement, les informations perceptives ne permettent pas de les différencier. Les informations proprioceptives sont alors le seul moyen de les distinguer.

Du point de vue de la robotique, cette information prend une importance accrue du fait de sa simplicité de manipulation. En effet, le processus d'intégration fournit directement une estimation de la position du robot dans un espace euclidien doté d'un repère cartésien. Dans ce type de repère, tous les outils de la géométrie mathématique sont utilisables. Ils permettent, par exemple, d'effectuer des calculs de chemin relativement simples lorsque l'on connaît la position du but et des obstacles.

1.3.2 Informations extéroceptives

Les informations extéroceptives, ou plus simplement perceptives, fournissent un lien beaucoup plus fort entre l'animat et son environnement. En effet, les informations proprioceptives fournissent des informations sur le *déplacement* de l'animat, alors que les informations perceptives fournissent des informations directement sur la *position* de l'animat dans l'environnement. Ces informations assurent un ancrage de l'animat dans son environnement, en lui permettant de choisir des positions qui peuvent être utilisées comme points de repère. Ces points de repère sont indépendants des déplacements de l'animat et seront reconnus quelle que soit l'erreur accumulée par les données proprioceptives. La reconnaissance de ces points est évidemment soumise à une incertitude, mais pas à une erreur cumulative, ce qui les rend utilisables à long terme.

Toutefois, pour être utile, un système perceptif doit être capable de distinguer le plus de lieux possible. Pour cela, il doit être capable de distinguer le plus de détails possibles, afin de faire la différence entre deux lieux différents mais d'apparence similaires. Or cette capacité à distinguer de petites variations dans l'environnement le rend en général aussi plus sensible à des variations non significatives qui proviennent de légers changements de l'environnement. Par exemple, un système visuel reposant sur une identification très fine des couleurs serait capable de distinguer deux objets de couleurs très voisines, mais risquerait d'être perturbé par un changement dans la luminosité générale de l'environnement. Un compromis est donc à trouver entre capacité de discrimination et résistance aux variations non significatives de l'environnement.

En ce qui concerne la robotique, ce compromis conduit en général au problème du *perceptual aliasing*. Ce problème désigne l'incapacité d'un système perceptif à distinguer de manière unique tous les lieux d'un environnement. Cette situation est très courante lorsque les robots utilisent des capteurs de distance aux obstacles tels que les capteurs à ultrasons. Dans un environnement intérieur de tels capteurs sont, par exemple, capables de mesurer la position du robot par rapport à un coin, mais ne fournissent aucune information sur la position le long d'un couloir rectiligne. Toutes les positions le long d'un couloir correspondent alors à des informations allothétiques identiques.

Deux solutions peuvent être apportées à ce problème. La première est d'utiliser des capteurs qui fournissent des données plus précises ou plus discriminantes. Dans le cas des capteurs de distance, il est, par exemple, possible d'utiliser un télémètre laser qui pourra distinguer les renfoncements des portes et sera ainsi plus précis. Il est aussi possible d'utiliser une caméra, qui sera sensible à la couleur des murs, en plus de leur forme, et pourra ainsi discriminer entre différentes positions dans un couloir. Toutefois, il est très difficile de garantir a priori que toutes les positions d'un environnement seront reconnues de manière unique. Cette solution ne permet donc pas, en général, de régler complètement le problème du *perceptual aliasing*, mais seulement d'en repousser l'apparition.

La seconde solution est d'utiliser des informations idiothétiques afin de distinguer deux positions physiquement différentes mais similaires pour le système perceptif. Ainsi deux lieux, dont la position relative mesurée par les données idiothétiques est non nulle, ne seront pas confondus. Cette solution est celle qui est mise en œuvre dans la majorité des systèmes de navigation, car elle permet d'utiliser les deux sources d'informations en limitant les défauts inhérents à chacune. Ainsi la dégradation progressive des informations idiothétiques est compensée par la reconnaissance de positions de l'environnement grâce aux informations allothétiques. Inversement, le problème de *perceptual aliasing* est réglé par l'utilisation des données idiothétiques.

Comme nous le verrons dans le chapitre suivant, il existe différentes méthodes pour utiliser conjointement les deux sources d'informations. Ces méthodes diffèrent par leur capacité à utiliser de manière plus ou moins efficace les avantages des deux types d'informations. D'une manière générale, la qualité d'un système de navigation dépend fortement de cette capacité.

Les capteurs allothétiques peuvent être de nature très variée et être utilisés de nombreuses façons différentes. Il est toutefois possible de distinguer deux utilisations distinctes de ces données pour la navigation. Ces deux utilisations dépendent de l'utilisation ou non d'un modèle métrique associé au capteur, modèle qui permet de prévoir la variation des mesures renvoyées par ce capteur en fonction du déplacement du robot.

Utilisation directe

Les données extéroceptives peuvent être utilisées de manière directe, sans aucun modèle décrivant leur variation en fonction de la position du robot. Cette utilisation permet de reconnaître simplement des lieux déjà visités en examinant les données allothétique recueillies en ces lieux. Cette méthode ne permet cependant que de reconnaître des lieux de l'environnement préalablement explorés par le robot. Sans modèle de la variation des capteurs, il est en effet impossible de prévoir les valeurs que les capteurs relèveront dans un lieu inexploré, même s'il est proche ou entouré de lieux connus.

Pour une telle utilisation directe, seules deux procédures permettant, d'une part, de mémoriser une perception et, d'autre part, de comparer deux perceptions, sont alors nécessaires. Ces procédures peuvent être mises en œuvre à partir de tous les types de capteurs existants. Il est, par exemple, possible d'utiliser la couleur dominante de l'environnement autour du robot, la température (en supposant qu'elle caractérise une zone de l'environnement, comme pour une chambre froide) ou le temps de retour d'une onde sonore quand elle est envoyée dans une direction donnée. La seule propriété utilisée est la constance des valeurs mesurées par un capteur pour un lieu donné. Cette constance permet de reconnaître un lieu déjà visité ou d'identifier un lieu nouveau dans l'environnement.

Une distinction supplémentaire est utile lors de l'utilisation de capteurs extéroceptifs sans modèles. Ces derniers peuvent, en effet, être regroupés en deux catégories différentes : les capteurs omnidirectionnels et les capteurs directionnels. Les capteurs omnidirectionnels permettent d'obtenir des informations dans toutes les directions autour du robot, à chaque mesure qu'ils effectuent. Les exemples classiques, en robotique, sont les ceintures de capteurs sonars, les télémètres laser panoramiques ou les caméras panoramiques. Les capteurs directionnels ne fournissent des informations que sur une portion de l'environnement du robot autour de la direction vers laquelle ils sont pointés (nous rappelons que, dans le cadre de cette thèse, nous supposons cette direction connue). L'exemple classique en robotique est la caméra.

Ces deux types de capteurs ont des fonctions similaires lors de l'utilisation d'un modèle interne complet de l'environnement dans lequel évolue le robot. Dans ce cas, en effet, ils fourniront tous deux une information sur les positions possibles du robot. Cette information peut être déterministe, par exemple en fournissant une liste des positions compatibles avec les données

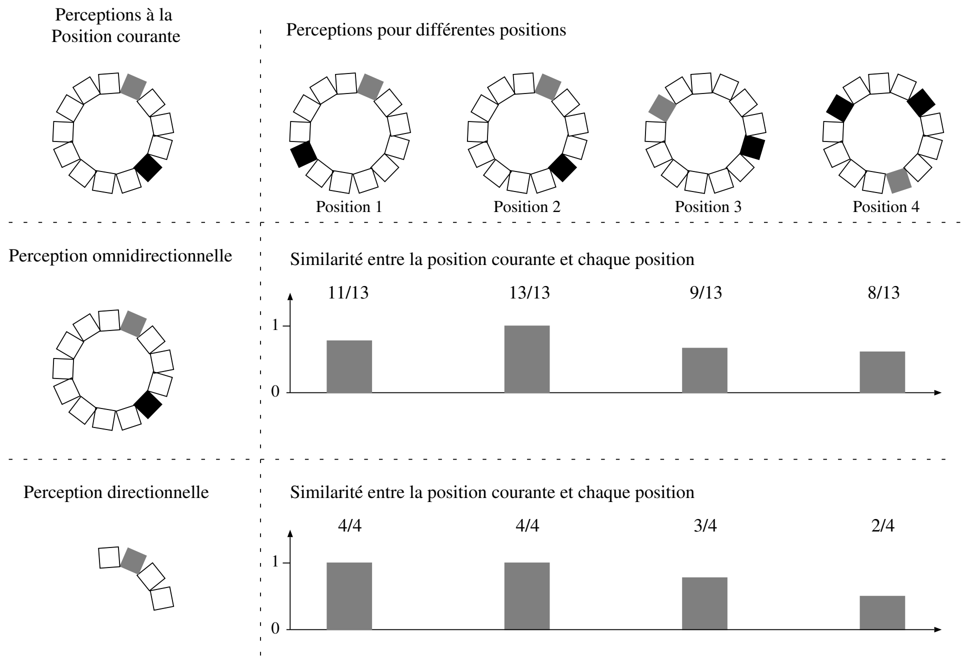


FIG. *1.1: La partie supérieure de la figure montre les données allothétiques mémorisées pour différentes positions, ainsi que les données correspondant à la position courante du robot. L'utilisation de capteurs omnidirectionnels permet d'estimer simplement la similarité des données de la position courante avec les données des différentes positions. Dans cet exemple, la similarité est simplement la proportion du nombre de directions pour lesquelles les données sont identiques. La position 2 est alors directement reconnue comme la position courante. Dans le cas de l'utilisation de capteurs directionnels, seule une partie des données est disponible à la position courante. La comparaison avec les différentes positions se fait donc sur moins de directions et conduit donc à une incertitude sur la position reconnue. Les positions 1 et 2 sont des candidats possibles dans cet exemple.*

allothétiques, ou probabiliste, par exemple en fournissant la probabilité pour le robot de se trouver en chacune des positions possibles au sein de l'environnement. La seule différence entre les données directionnelles et omnidirectionnelles sera une différence de qualité. Les données omnidirectionnelles permettent, en effet, d'utiliser d'emblée toute l'information disponible pour une position donnée. L'estimation de la position est donc relativement simple car il suffit de comparer les données recueillies avec les données disponibles pour différents lieux (cf. figure 1.1). Les données directionnelles ont l'inconvénient d'être partielles, et donc de nécessiter le choix de la direction dans laquelle effectuer la mesure. Le résultat est, en général, une moins bonne discrimination entre les différents lieux possibles, à cause d'une augmentation du *perceptual aliasing* (cf. figure 1.1).

Lors de la construction du modèle interne, en revanche, ces deux types de capteurs ont des propriétés différentes. Ces propriétés sont liées à la partialité des informations disponibles pour les capteurs directionnels, qui entraîne des biais dans l'estimation de la position de l'animat. En effet, dans ce cas, les informations mémorisées dans la carte seront partielles car elles ne couvrent pas

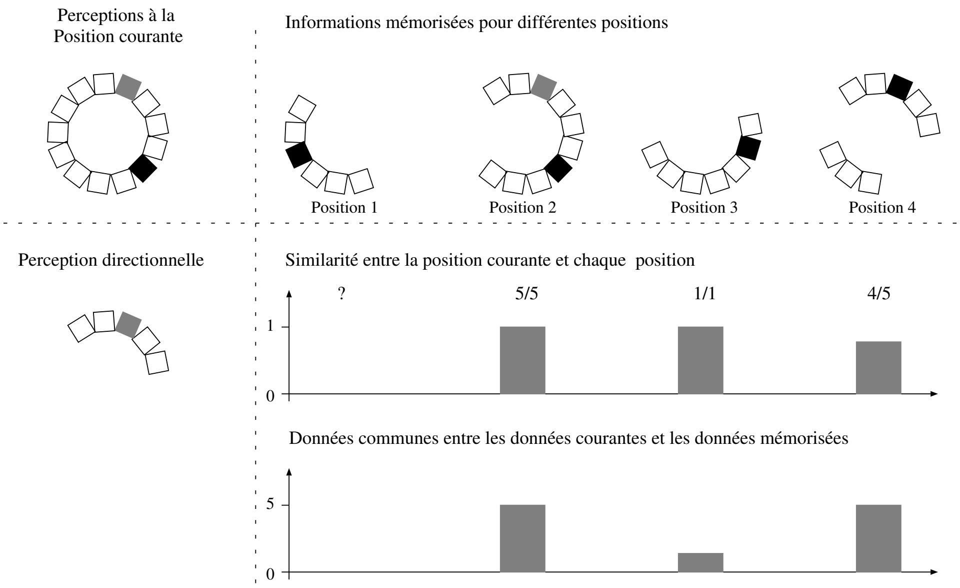


FIG. *1.2: La partie supérieure de la figure montre les données partielles mémorisées pour différentes positions de l'environnement. Le calcul de la similarité entre les perceptions effectuées au lieu courant et les différents lieux mémorisés sera alors compliqué par la partialité de ces données. Simplement prendre en compte la proportion de données communes identiques conduit en effet à une estimation très imprécise de la position. Ce calcul dépendra en effet de la quantité de données communes, qui pourra varier et même être nulle. Il faut alors répondre à la question de savoir si on privilégie des données très similaires sur un petit nombre de valeurs communes, ou des données légèrement différentes, mais sur un grand nombre de valeurs. Dans cet exemple avec des données directionnelles, il est par exemple difficile de décider si les données sont plus similaires avec celles de la position 3 (1 donnée commune identique) ou celle de la position 4 (4 données identiques sur 5 communes).*

toutes les directions pour toutes les positions. La question de savoir à quelle position des données allothétiques partielles correspondent le mieux devient alors difficile. En effet, en plus de comparer les données disponibles, il faut également tenir compte de la quantité de données communes disponibles pour les données comparées (cf. figure 1.2). Des données similaires ayant seulement une petite partie commune apporteront évidemment moins de certitude que des données similaires dans la totalité des directions. Mais le plus gros problème survient quand il faut choisir entre des données identiques sur une petite partie et des données légèrement différentes, mais sur une plus grande partie. Il faut alors décider si on privilégie l'exhaustivité de l'information, ou son exactitude.

Ces problèmes dans l'utilisation de données partielles vont donc nécessiter des choix relatifs à la direction des capteurs et ces choix influenceront la qualité de la carte et la qualité du système de localisation. Des procédures de *perception active*, qui permettront de choisir la direction des capteurs directionnels en fonction de la situation courante, devront donc être utilisées.

Ces problèmes deviennent cruciaux lorsque l'on cherche à comparer globalement les données allothétiques avec les données disponibles pour tous les lieux visités. Nous verrons par la suite que cela concerne essentiellement les système de localisation globale, et moins les système locaux (cf. chapitre 2). Malheureusement, les systèmes de localisation globale sont les plus intéressants pour l'autonomie d'un animat et le système développé dans cette thèse se classe dans cette catégorie. Nous verrons dans le chapitre 5 les solutions qui nous avons apporté à ce problème de partialité de l'information.

Utilisation d'un modèle métrique

La seconde méthode d'utilisation d'un capteur allothétique consiste à utiliser un modèle métrique associé. Un tel modèle permet de traduire les informations données par le capteur dans un espace métrique qui est en général le même que celui utilisé pour estimer la position du robot grâce à l'odométrie. Il est ainsi possible d'estimer la position d'objets de l'environnement par rapport au robot, et ainsi de prévoir les données que ce capteur relèvera pour des positions différentes du robot. L'utilisation d'un tel modèle n'est toutefois possible que pour certains capteurs. Il est, par exemple, possible d'utiliser un tel modèle associé à un capteur à ultrasons, à un télémètre laser ou à une paire de caméras stéréoscopique, mais pas à un capteur d'odeur.

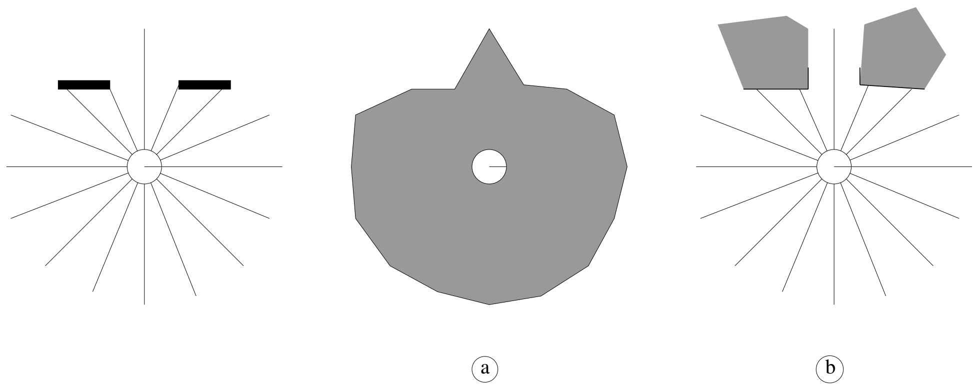


FIG. *1.3: Un modèle métrique pour les données allothétiques permet deux utilisations de ces données. La première est similaire à celle qui traite de telles données sans modèle métrique et requiert la simple mémorisation de ce qui est perçu en un lieu donné (Partie a). La seconde utilise ces données pour reconstituer les objets rencontrés dans l'environnement, objets qui pourront tous être mémorisés dans un cadre de référence commun, indépendamment de la position depuis laquelle ils ont été perçus (Partie b).*

Avec un tel modèle, les valeurs des capteurs peuvent être utilisées simplement pour caractériser chaque lieu atteint par le robot (cf. figure 1.3a). La méthode est alors la même que celle mise en place quand les capteurs sont utilisés sans modèle métrique. L'utilisation d'un modèle métrique présente toutefois l'avantage que les informations recueillies ont une sémantique plus forte. En effet, ces informations caractérisent la structure spatiale locale de l'environnement, en plus de la simple apparence de l'environnement depuis la position du robot. Cette structure spatiale peut alors être utilisée lors de la comparaison de différents lieux. Il est par exemple possible de reconnaître un couloir en fonction de sa largeur, indépendamment de la position du robot dans ce couloir. En effet, sans utilisation de modèles métriques, deux perceptions recueillies en des positions différentes du couloir seront simplement différentes. En utilisant un modèle métrique, il est possible de calculer la largeur du couloir, par exemple, à partir des données recueillies et ainsi de déterminer si ces deux positions peuvent correspondre au même couloir.

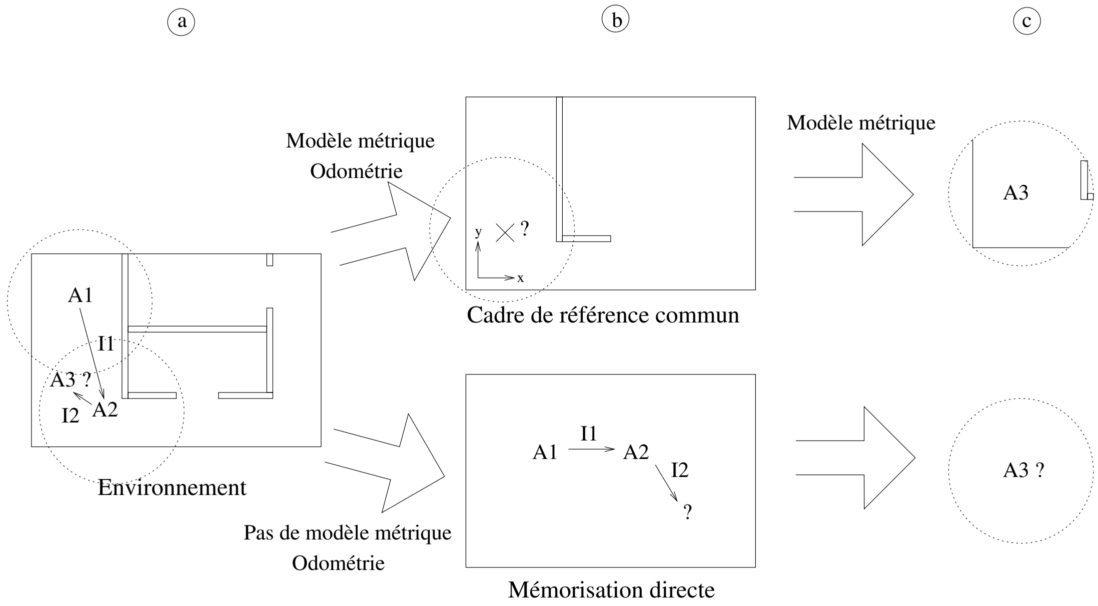


FIG. *1.4: Un modèle métrique des données allothétiques permet d'inférer les données qui devraient être perçues pour des positions encore non visitées. Dans cet exemple, les données allothétiques* $A1$ *et* $A2$ *sont perçues en deux positions reliées par des données idiothétiques* $I1$ *(partie a). L'utilisation d'un modèle métrique permet de fusionner ces informations dans un cadre de référence commun où des objets sont représentés, ici deux murs orthogonaux (partie b, haut). Sans modèle métrique, ces données peuvent seulement être mémorisées de manière séparée (partie b, bas). Dans le cas de l'utilisation d'un modèle métrique, les données peuvent ensuite être utilisées pour estimer les données allothétiques* $A3$ *pour une nouvelle position reliée à la précédente par les données idiothétiques* $I2$*. Ici, le modèle permet d'inférer que les données* $A3$ *correspondent à un coin de murs (partie c, haut). Sans un tel modèle, seules les positions visitées peuvent être reconnues, et aucune inférence ne peut être faite pour les positions non visitées (partie c, bas).*

Cependant, grâce à un modèle métrique, les données perceptives peuvent être utilisées de manière différente. En effet, dans l'utilisation précédente, sans modèle métrique, les données perceptives sont utilisées pour caractériser l'apparence de l'environnement depuis un lieu. Cette caractérisation ne permet pas d'identifier individuellement des objets distants du robot qui pourraient servir de points de repères, appelés *amers*. L'utilisation d'un modèle métrique permet l'identification de tels points (cf. figure 1.3b). La perception de ces amers permet alors, en retour, d'obtenir des informations sur la position du robot. Cette utilisation des informations allothétiques offre l'avantage supplémentaire de permettre au robot d'inférer les valeurs que mesureront les capteurs allothétiques dans des positions différentes, mais voisines de sa position courante (cf. figure 1.4). Par exemple, si un robot perçoit un mur à cinq mètres devant lui, il peut prédire qu'en avançant d'un mètre, il percevra le mur à quatre mètres. Un autre moyen de présenter cette propriété est de dire que les données allothétiques seules permettent d'estimer la position métrique relative de deux lieux (cf. figure 1.5). Ainsi, si un robot perçoit deux fois un mur devant lui, d'abord à cinq mètres puis à quatre mètres, il pourra en déduire qu'il a avancé d'un mètre. Cette propriété permet au robot d'estimer sa position avec précision sur une part plus importante de son environnement et ne limite plus la localisation aux lieux déjà visités. Cet avantage est une conséquence directe de la fusion des informations allothétiques et idiothétiques au sein d'une même représentation, qui permet le passage d'un type d'information à l'autre.

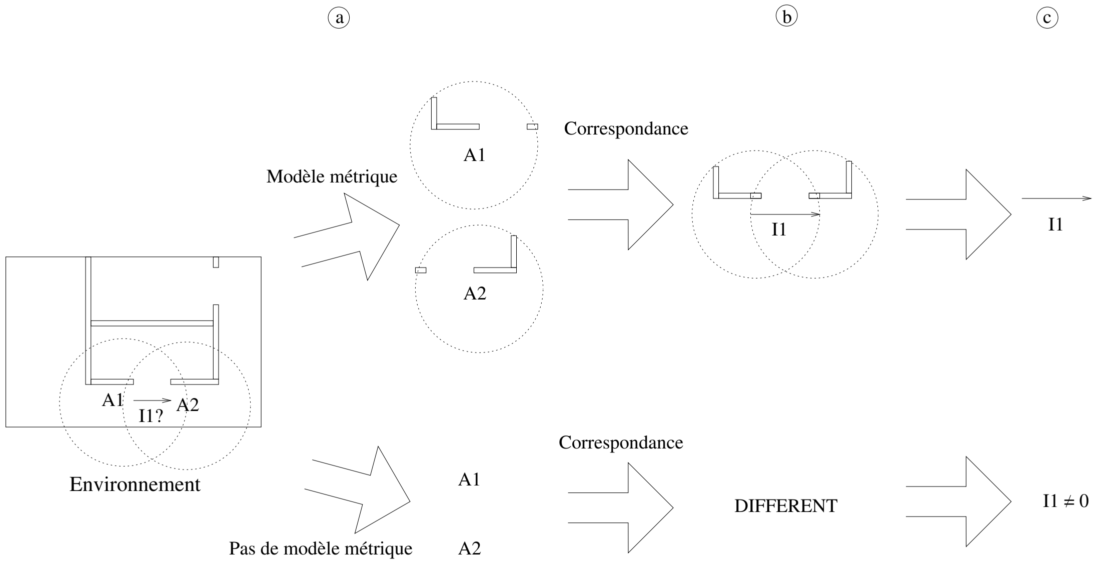


Fig. *1.5: Un modèle métrique des données allothétiques permet d'inférer la position relative $I1$ de deux positions depuis lesquelles des données allothétiques $A1$ et $A2$ ont été perçues (partie a). Cette estimation requiert d'abord la recherche d'un objet de l'environnement commun aux deux données allothétiques (partie b, haut). L'utilisation de cet objet commun rend alors possible l'estimation des données idiothétiques $I1$ reliant les deux positions (partie c, haut). Sans modèle métrique, seule la similarité de deux données allothétiques peut être mesurée (partie b, bas). Il est alors possible d'estimer si ces deux situations peuvent correspondre au même lieu ou non, c'est-à-dire si $I1$ est nulle ou non (partie c, bas).*

Toutefois, la mise au point d'un tel modèle peut être très difficile. La relation qui lie la valeur mesurée par un capteur à la position des objets du monde réel peut être, en effet, très complexe. Dans le cas des capteurs à ultrasons, par exemple, si un mur se trouve juste dans l'axe du capteur, sa distance est simplement mesurée par le temps mis par l'onde sonore pour revenir au capteur. Mais, dans le cas où le mur est fortement incliné par rapport au capteur, l'écho peut ne pas revenir en direction du capteur qui ne détectera alors aucun obstacle. Un autre problème vient de la texture des murs. Un mur recouvert de textile ou d'un matériau souple renverra les échos très différemment d'un mur de béton. En conséquence, pour une distance donnée, le capteur percevra des distances différentes suivant le matériau des murs. Ces deux exemples montrent que le modèle métrique associé à un capteur ne dépend pas que du capteur. Il dépend aussi fortement de propriétés locales de l'environnement qui sont difficiles ou impossibles à prendre en compte dans un modèle du capteur seul.

La partialité de l'information que nous avons mentionnée précédemment lors de l'utilisation de capteurs directionnels sans modèle métrique pose également des problèmes dans le cas de l'utilisation d'un tel modèle. En effet, pour les systèmes de localisation globale (cf. chapitre 2), qui doivent comparer toutes les positions possibles du robot, une information partielle rendra difficile cette comparaison, que l'on utilise un modèle métrique ou non. Toutefois, comme nous l'avons mentionné précédemment, un modèle métrique permet de calculer les données allothétiques perçues en un lieu, en fonction des données allothétiques perçues en d'autres lieux. Ce mécanisme permet donc d'avoir des données allothétiques complètes pour l'ensemble des positions possibles du robot beaucoup plus facilement et ainsi de limiter le problème de la partialité des informations.

1.4 Représentations en robotique

Les deux utilisations possibles des données allothétiques présentées précédemment conduisent naturellement à deux types de représentation de l'environnement. Lorsqu'aucun modèle métrique n'est utilisé pour les capteurs, les données sont mémorisées dans une carte *topologique* [Kuipers and Byun, 1991, Thrun, 1999] (cf. figure 1.6). Dans une telle carte, un ensemble de lieux et leurs relations de voisinage sont mémorisées. Chaque lieu est défini au moyen de données allothétiques recueillies lorsque le robot se trouve en ces lieux. Les relations entre lieux sont, pour leur part, déduites des données idiothétiques. En revanche, lorsque un modèle métrique des capteurs est utilisé, les données peuvent être mémorisées au sein d'une carte *métrique* [Moravec and Elfes, 1985, Chatila and Laumond, 1985] (cf. figure 1.6) qui rassemble dans un même cadre de référence les données idiothétiques et allothétiques. La carte contient alors un ensemble d'objets, ayant chacun une position associée. Naturellement, il est possible de construire une carte topologique lorsqu'un modèle métrique est utilisé. Dans ce cas, toutefois, les données allothétiques ne sont pas utilisées pour estimer la position relative des lieux visités, mais seulement pour caractériser ces lieux.

La notion de topologique et de métrique est différente de celle mentionnée dans l'introduction. Ici, cette notion fait référence à la manière dont les informations sont mémorisées et non à la stratégie de navigation utilisée. Ainsi une carte topologique pourra contenir des informations métriques et pourra être utilisée pour une stratégie de navigation métrique, au sens donné dans l'introduction. Dans la suite de ce mémoire, le concept topologique/métrique fera toujours référence au type de carte utilisé, et non à la stratégie correspondante.

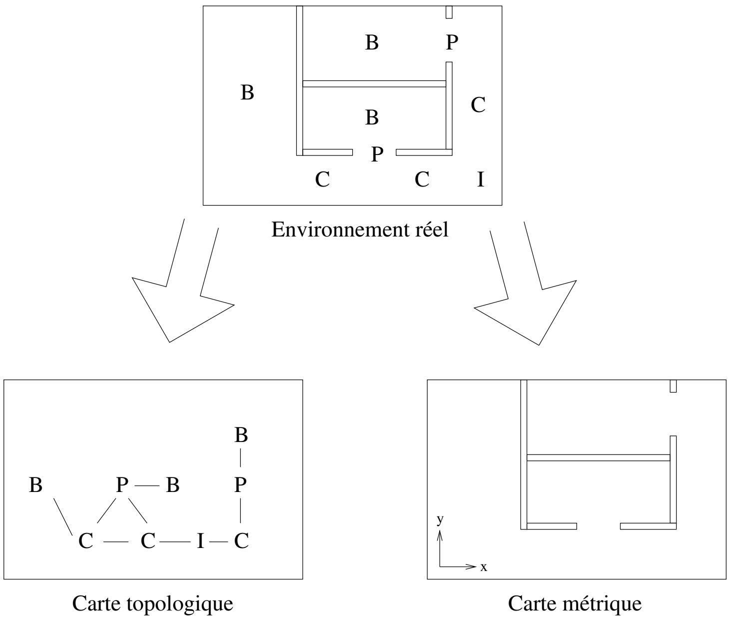


FIG. *1.6: Les cartes utilisées en robotique peuvent être de deux types. Les cartes topologiques, d'une part, mémorisent un ensemble de lieux, ainsi que les manières de se déplacer de l'un à l'autre (dans cet exemple, B=Bureau, C=Couloir, P=Porte et I=Intersection). Les cartes métriques, d'autre part, mémorisent un ensemble d'objets perçus (des murs dans cet exemple) avec une position dans un cadre de référence global.*

1.4.1 Cartes topologiques

Nature

Les cartes topologiques permettent de représenter l'environnement du robot sous forme de graphe. Les nœuds du graphe correspondent à des lieux, c'est-à-dire des positions que le robot peut atteindre. Les arêtes liant les nœuds marquent la possibilité pour le robot de passer directement d'un lieu à un autre et mémorisent la manière de réaliser ce passage.

La détection et la mémorisation des lieux ne reposent que sur deux procédures qui utilisent les données allothétiques. La première permet simplement de comparer deux situations allothétiques et donc de reconnaître un lieu de la carte ou de détecter un lieu nouveau. La seconde procédure permet de mémoriser un nouveau lieu ou d'adapter la définition d'un lieu lors des passages successifs du robot en ce lieu. Comme nous l'avons déjà mentionné, la reconnaissance d'un lieu est soumise au bruit des capteurs et au problème de *perceptual aliasing*. En conséquence, la première procédure peut donner des résultats erronés. Par exemple, un lieu déjà visité peut ne pas être reconnu, ou un lieu nouveau peut être confondu avec un lieu déjà mémorisé. Pour résoudre ces problèmes, la reconnaissance des lieux fera donc appel aux données idiothétiques en plus des données allothétiques. De nombreuses méthodes ont été mises en œuvre dans ce but, elles feront l'objet des chapitres 2 et 3.

Les données mémorisées dans les arêtes du graphe sur les relations de voisinage entre lieux proviennent, pour leur part, exclusivement des données idiothétiques. Cela est caractéristique des cartes topologiques, dans lesquelles les données allothétiques ne sont pas utilisées pour estimer les positions relatives des lieux visités.

Avantages

Un avantage important des cartes topologiques est qu'elles ne requièrent pas de modèle métrique des capteurs pour fusionner les données idiothétiques et allothétiques au sein d'une représentation unifiée de l'environnement. Cela est avantageux pour deux raisons. D'une part, ces modèles peuvent, comme nous l'avons vu précédemment, être difficiles à obtenir ou s'avérer peu fiables. D'autre part, le fait de ne pas fusionner les deux sources d'informations permet de séparer les influences des erreurs correspondantes. En effet, l'estimation de la position d'objets, lorsque l'on utilise un modèle métrique, dépend à la fois des valeurs mesurées par les capteurs et de la position du robot. Une erreur sur la position d'un objet peut donc provenir des deux sources. Déterminer la contribution de chacune des sources peut être très difficile. Dans les cartes topologiques, au contraire, le bruit sur les mesures des capteurs influe principalement sur la reconnaissance des lieux, tandis que le bruit sur l'odométrie influe principalement sur la position associée à chaque lieu.

La mémorisation de l'environnement sous forme d'un ensemble de lieux distincts autorise en général une définition des lieux plus directement reliée aux capacités perceptives du robot. En effet, comme les données allothétiques ne sont pas transformées dans un repère métrique, il n'y a pas de limitation au type de capteurs utilisables (cf. la section 1.3.2). Cette utilisation directe des données allothétiques permet un meilleur ancrage dans l'environnement, c'est-à-dire une meilleure mise en relation du robot avec son environnement. Puisque la carte est très proche des données brutes perçues par le robot, il est en général assez simple de comparer et mémoriser des lieux de l'environnement.

Cette proximité avec les données brutes conduit en général la représentation topologique à utiliser beaucoup moins de concepts de haut niveau que les représentations métriques. La carte topologique reste ainsi proche des possibilités du robot, en mémorisant ses perceptions et ses déplacements possibles, indépendamment de concepts de plus haut niveau tels que des objets ou des obstacles.

La discrétisation de l'environnement correspondant au choix des lieux représentés dans la carte est un autre point fort des cartes topologiques. Cette discrétisation est en effet très utile pour la planification des mouvements du robot, qui se réduit alors à la recherche de chemin dans un graphe. Cette recherche est, en terme de complexité algorithmique, beaucoup plus simple que la recherche d'un chemin dans un espace continu à deux dimensions. Cet avantage est encore plus important lorsque les lieux représentés dans la carte correspondent à des structures humaines telles que les portes, les couloirs ou les pièces. La discrétisation permet alors de décrire et de résoudre les problèmes de manière naturelle pour les humains, par exemple en donnant l'ordre d'aller au bureau B744, plutôt que de dire d'aller à la position définie par les coordonnées x=354,y=285.

Inconvénients

Comme nous l'avons déjà mentionné, l'utilisation directe des données allothétiques sans modèle métrique empêche d'estimer ces données pour des positions non visitées. En conséquence, les cartes topologiques nécessitent en général une exploration très complète de l'environnement pour le représenter avec précision. En particulier, tous les lieux intéressants que l'on souhaite trouver dans la carte devront être visités au moins une fois au cours de la construction de la carte, parce qu'ils ne peuvent pas être perçus à distance. Dans le cas où les lieux représentés sont des structures d'assez haut niveau (comme des couloirs ou des pièces), cela n'est pas gênant car ces lieux sont peu nombreux et une exploration exhaustive est donc relativement rapide. En revanche, dans les cartes topologiques représentant des lieux avec une assez grande densité spatiale, cela peut être un inconvénient, car l'exploration complète de l'environnement demandera un temps important.

La reconnaissance des lieux de l'environnement peut également être difficile dans le cas de capteurs très bruités, ou d'environnements très dynamiques. Elle est, de plus, très sensible au problème de *perceptual aliasing*. Ces difficultés conduisent à des problèmes de fausse reconnaissance, c'est-à-dire à la reconnaissance d'un lieu donné alors que le robot se trouve dans un autre lieu. À leur tour, ces fausses reconnaissances conduisent à une mauvaise topologie de la carte et à des liens qui relient des nœuds de la carte qui ne sont pas physiquement reliés dans l'environnement. Ces difficultés rendent problématique la construction de cartes topologiques dans des environnements de grande taille, car la carte résultante risque d'être incohérente. Il devient alors très difficile d'estimer correctement la position du robot au sein de cette carte et de lui ajouter de nouvelles informations sans erreurs.

Comme nous l'avons vu, la représentation de l'environnement peut être assez proche des données brutes des capteurs du robot, ce qui peut être un avantage du point de vue de l'autonomie du robot. Toutefois, cette représentation centrée sur l'individu peut poser des problèmes pour la réutilisation de la carte. En effet, la carte pourra être difficile à adapter à un robot avec des capteurs légèrement différents, du fait du manque de modèle de capteur. Cette difficulté est liée au manque de représentation indépendante de l'individu de l'environnement et à l'absence d'un modèle métrique des capteurs. En effet, si l'on dispose d'une telle représentation, l'adaptation à un nouveau robot se fait simplement au niveau du modèle de capteur, sans modification de la carte elle-même. Cela est plus difficile avec une carte topologique, au sein de laquelle il est quasiment impossible de changer les données recueillies par un capteur pour les transformer en données telles qu'un autre capteur aurait pu les acquérir. De plus, cette représentation centrée sur un individu est moins naturelle pour un opérateur humain, plus habitué aux représentations objectives du type plan d'architecte, ce qui peut être gênant lorsque l'on souhaite une interaction forte entre un opérateur et le robot.

Implémentation

Définition des nœuds

Le choix de ce que vont représenter les nœuds de la carte détermine tout le processus de construction de la carte topologique. Ce choix est lié aux capacités de perception dont on a doté le robot, lequel devra être capable de détecter les lieux en question. De plus, la localisation et la mise à jour de la carte se feront chaque fois qu'un tel lieu aura été détecté. La détection de ces lieux peut être contrainte par des choix des opérateurs humains ou être complètement autonome.

Nœuds définis par le concepteur

La première possibilité est de définir directement quels lieux doivent être détectés par le robot et comment ils doivent l'être. Des procédures sont alors écrites qui permettent de détecter spécifiquement chaque type de lieu. Le choix le plus courant est l'utilisation de couloirs, de portes et d'intersections [Dedeoglu et al., 1999, Hertzberg and Kirchner, 1996, Kunz et al., 1997, Shatkay and Kaelbling, 1997]. Lorsque ce choix est fait, un très petit nombre de lieux différents peut être détecté, ce qui rend le problème du *perceptual aliasing* omniprésent. Les systèmes concernés dépendent donc en général fortement des données idiothétiques pour résoudre ce problème.

Nœuds définis à des positions canoniques

Plutôt que de définir complètement les lieux que peut détecter le robot, le concepteur peut simplement définir dans quels types de situations le robot peut enregistrer un lieu, laissant au robot le soin de définir chaque lieu précisément. Par exemple, le concepteur peut doter le robot de la capacité générale de détecter des portes. Lorsque le robot détectera une porte, il enregistrera un nouveau nœud dans la carte, mais ce nœud sera défini par la situation précise dans laquelle il se trouve quand il rencontre cette porte. Il pourra, par exemple, enregistrer la couleur de la porte, ou le numéro qui est inscrit dessus. Cette méthode de définition des nœuds a été proposée par Kuipers et Byun [Kuipers and Byun, 1991] sous le nom de *distinctive places*, puis utilisée sous une forme différente par Engelson et McDermott [Engelson and McDermott, 1992] et par Kortenkamp et Weymouth [Kortenkamp and Weymouth, 1994] sous le nom de *gateways*.

Nœuds définis de manière non supervisée

La troisième méthode pour définir les nœuds d'un carte topologique consiste à les définir comme des zones où la situation allothétique est approximativement constante. Cela est obtenu en général par la catégorisation non supervisée des données allothétiques [Bachelder and Waxman, 1994, Duckett and Nehmzow, 1997, Franz et al., 1998, Gaussier et al., 2000, Kurz, 1995, Levitt and Lawton, 1990, Mataric, 1992, Nehmzow and Owen, 2000, Ulrich and Nourbakhsh, 2000, Von Wichert, 1998]. Les données allothétiques sont donc regroupées en catégories contenant des données similaires, sans que ces catégories soient spécifiées par un concepteur humain. Chaque catégorie correspond alors à un ou plusieurs nœuds de la carte. Le nœud correspondant à une catégorie étant unique dans le cas où il n'y a pas de perceptual aliasing. Cette méthode est bien adaptée à des robots autonomes car la catégorisation ne nécessite aucun superviseur, ni aucune définition a priori des données correspondant à un nœud. A ce titre, elle est utilisée dans tous les systèmes de navigation qui s'inspirent des comportements de navigation des animaux [Arleo and Gerstner, 2000, Balakrishnan et al., 1999, Burgess et al., 1994, Sharp, 1991, Touretzky et al., 1994].

Pour mettre en œuvre une telle approche, il faut définir un critère qui permette de décider quand un nouveau lieu a été atteint. Le choix le plus évident est de comparer constamment la situation courante à celle du précédent nœud reconnu. Lorsque la différence est suffisamment importante, on considère qu'un nouveau lieu a été atteint. Cette méthode est utilisée par certains modèles [Franz et al., 1998, Gaussier et al., 2000, Kurz, 1995, Mataric, 1992, Nehmzow and Owen, 2000], mais requiert que les perceptions soient comparées en temps réel, ce qui peut-être difficile pour certains capteurs (les caméras, par exemple). D'autres modèles considèrent donc plus simplement qu'un nouveau nœud a été atteint lorsque la distance parcourue depuis la dernière reconnaissance est assez grande [Arleo and Gerstner, 2000, Touretzky et al., 1994, Von Wichert, 1998, Yamauchi and Beer, 1996].

Définition des arêtes

Les arêtes reliant les nœuds permettent de mémoriser des données sur les relations de voisinage entre lieux représentés par les nœuds. Ces données sont en général obtenues grâce aux informations idiothétiques. Elles peuvent être plus ou moins précises et représentées sous diverses formes.

Relation d'adjacence

La première information que porte une arête est une information d'adjacence entre les deux lieux représentés par les nœuds qu'elle connecte. L'existence d'une arête signifie donc que le robot peut passer directement d'un lieu à l'autre, sans passer par un lieu intermédiaire. Si certains modèles ne mémorisent que cette information d'adjacence [Franz et al., 1998, Gaussier et al., 1998, Hertzberg and Kirchner, 1996, Kortenkamp et al., 1994, Nourbakhsh et al., 1995, Ulrich and Nourbakhsh, 2000], cette information est prise en compte dans tous les modèles, même si des informations supplémentaires sont enregistrées dans les arêtes.

Relations métriques

Des informations métriques sur la position relative des lieux peuvent être mémorisées dans les arêtes. Ces informations portent en général sur la position relative des lieux reliés par l'arête [Engelson and McDermott, 1992, Hafner, 2000, Kuipers and Byun, 1991, Kunz et al., 1997, Nehmzow and Owen, 2000, Shatkay and Kaelbling, 1997, Simmons and Koenig, 1995, Von Wichert, 1998]. Elles sont fournies et quantifiées par les données idiothétiques lorsque le robot se déplace d'un lieu à l'autre. Cette méthode présente l'avantage de limiter l'accumulation de l'erreur des données idiothétiques, puisque ces données ne sont utilisées que sur la distance reliant un nœud à un autre. Cette distance est en général assez courte pour éviter une accumulation d'erreurs trop importante. Les cartes topologiques utilisant de telles informations métriques sont appelées cartes *diktiométriques* [Engelson and McDermott, 1992].

Association de position aux nœuds

Dans le but d'intégrer les données idiothétiques à une carte topologique, il est également possible d'associer une position à chacun des nœuds. Cette position se mesure dans l'espace dans lequel s'expriment les données idiothétiques et correspond à la position des différents lieux dans l'environnement. Ce type de carte se rapproche fortement des cartes métriques, à la différence que seuls les lieux visités par le robot, et non les objets perçus par le robot, sont mémorisés. L'inconvénient, par rapport à l'approche précédente, est qu'il est nécessaire de corriger les informations idiothétiques car elles ne sont plus utilisées localement. Chaque nœud ayant une position dans un cadre de référence global, il est possible de se contenter de cette information, sans ajouter de liens entre les nœuds [Oore et al., 1997, Arleo and Gerstner, 2000, Balakrishnan et al., 1999]. Toutefois, certains modèles utilisent également des liens pour mémoriser l'information d'adjacence [Mataric, 1992, Touretzky et al., 1994, Kurz, 1995, Yamauchi and Langley, 1997, Duckett and Nehmzow, 1997, Von Wichert, 1998, Dedeoglu et al., 1999]. Comme l'information de position de chaque nœud est absolue, ce type de carte est appelé carte *diktiométrique absolue*.

Relation implicite

Dans certains cas, il est possible de retrouver les relations de position entre les lieux au vu de leur seule définition allothétique. Cela est possible, par exemple, lorsque les lieux sont définis par la configuration d'amers distants qui peuvent être perçus par le robot lorsqu'il se trouve à cette position. Un certain nombre d'amers communs, visibles depuis deux lieux différents permettront d'avoir des informations sur la position relative de ces lieux. L'existence d'amers communs peut donc être utilisée comme lien implicite [Levitt and Lawton, 1990, Sharp, 1991, Burgess et al., 1994].

1.4.2 Cartes métriques

Nature

Dans une carte métrique, l'environnement est représenté par un ensemble d'objets auxquels sont associées des positions dans un espace métrique, généralement en deux dimensions. Cet espace est, la plupart du temps, celui dans lequel s'exprime la position du robot estimée par les données idiothétiques. Les données allothétiques permettent, en utilisant un modèle métrique des capteurs, de détecter ces objets et d'estimer leur position par rapport au robot. La position de ces objets dans l'environnement est alors calculée en utilisant la position estimée du robot. La fusion des deux sources d'information au sein d'un même cadre de représentation est caractéristique des cartes métriques.

Les objets mémorisés dans la carte peuvent être très divers et seront détaillés dans la suite de cette section. Dans un grand nombre d'implémentations, cependant, ces objets correspondent aux obstacles que le robot pourra rencontrer dans son environnement. La carte de l'environnement correspond ainsi directement à l'espace libre, c'est-à-dire à l'espace dans lequel le robot peut se déplacer.

Avantages

L'avantage principal des cartes métriques est de permettre de représenter l'ensemble de l'environnement, et non un petit sous-ensemble de lieux comme le font les cartes topologiques. Cette représentation complète permet ainsi d'estimer avec précision et de manière continue la position du robot sur l'ensemble de son environnement. De plus, cette représentation complète ne se limite pas aux positions physiquement explorées, mais s'étend à toutes les zones que le robot a pu percevoir depuis les lieux qu'il a visités. Cette propriété permet la construction d'une carte plus exhaustive de l'environnement en un temps plus court.

Un autre avantage des cartes métriques est lié au fait que la position du robot est définie de manière non ambiguë par ses coordonnées au sein de l'espace dans lequel est représentée la carte. Il s'ensuit une utilisation simple et directe de toutes les informations métriques fournies par les données idiothétiques ou allothétiques. Cela est un avantage par rapport aux cartes topologiques où les positions possibles du robot sont limitées aux nœuds présents dans la carte et sont donc relativement imprécises. Une telle représentation, dans laquelle chaque nœud peut couvrir une zone étendue de l'environnement, rend plus difficile l'utilisation des données métriques car la position relative de deux zones est moins bien définie.

La représentation indépendante de l'individu de l'environnement utilisée dans les cartes métriques apporte un certain nombre d'avantages supplémentaires. Comme nous l'avons mentionné à propos des cartes topologiques, une telle représentation permet une réutilisation plus facile d'une carte sur des robots différents, équipés de capteurs différents, l'essentiel de l'adaptation se déroulant au niveau des capteurs. Ce type de représentation est aussi facilement interprétable par un humain, ce qui peut être important dans le cas où il doit intervenir dans les déplacements du robot.

Cette représentation peut de plus utiliser des concepts de plus haut niveau, tels que des objets, des obstacles ou des murs. Cela permet un apport de connaissance plus facile de la part des humains, par exemple pour imposer que deux murs soient perpendiculaires.

Inconvénients

Lors de l'utilisation de cartes métriques, les données idiothétiques ont en général une importance supérieure à celle qu'elles ont dans l'utilisation d'un carte topologique. Par conséquent, une odométrie plus fiable est requise. Le niveau de fiabilité nécessaire est souvent atteint en imposant des limitations sur l'environnement du robot. Par exemple, il est possible d'imposer que tous les couloirs soient orthogonaux, afin de pouvoir corriger efficacement la dérive de l'estimation de la position.

Comme nous l'avons mentionné dans la section 1.3.2, un modèle métrique des capteurs peut être difficile à obtenir. Les problèmes liés au bruit des capteurs et à la difficulté de modéliser de manière fiable leur relation avec l'environnement constituent donc un point faible important des cartes métriques.

Enfin, le calcul de chemin au sein des cartes métriques peut être plus complexe, car la planification se déroule dans un espace continu et non dans un espace préalablement discrétisé, comme c'est le cas pour les cartes topologiques. De nombreux modèles recourent d'ailleurs à l'extraction d'un carte topologique depuis la carte métrique pour réaliser cette opération de planification [Latombe, 1991].

Implémentation

Deux méthodes principales sont utilisées pour mémoriser des informations sous forme de carte métrique. La première méthode consiste à extraire explicitement des objets des données allothétiques et de les enregistrer dans la carte avec leur position estimée. Les objets peuvent être de types très variés et se situer à différents niveaux d'abstraction. La seconde méthode s'attache à représenter directement l'espace libre accessible au robot et les zones d'obstacles qu'il ne peut pas franchir, sans avoir recours à l'identification d'objets individuels.

Représentation d'objets

Points

Les objets les plus simples qui peuvent être utilisés sont des points [Levitt and Lawton, 1990, Prescott, 1995, Feder et al., 1999]. Ces points correspondent à des objets de l'environnement de taille suffisamment petite, ou situés suffisamment loin du robot, pour pouvoir être considérés comme ponctuels. Ces objets sont les amers les plus simples qui peuvent être utilisés comme points de repère. Cependant, la perception d'un point de l'environnement ne suffit pas à déterminer de manière unique la position du robot. Ce type de points de repère est par conséquent relativement pauvre et contraint à la détection de plusieurs objets pour assurer une localisation précise. De plus, reconnaître un tel point de manière non ambiguë est souvent difficile et requiert une bonne capacité de discrimination de la part des capteurs.

Pour surmonter ces difficultés, certains modèles ont recours à des ensembles de points disséminés sur la surface des objets de l'environnement [Lu and Milios, 1997, Gutmann and Konolige, 2000, Thrun et al., 2000]. Ces points sont en général obtenus par des télémètres laser, qui permettent d'en recueillir un grand nombre avec une résolution spatiale élevée. Les objets sont ainsi définis par la configuration d'ensembles de points, et non plus par des points uniques. Cette méthode présente donc l'avantage de ne pas recourir à l'identification individuelle de chaque point.

Points orientés

Afin d'obtenir plus d'information sur la position du robot par la perception d'un seul objet, il est possible de doter chaque objet ponctuel d'une orientation. La perception d'un tel point orienté permet alors d'estimer la position du robot de manière unique. Un tel type de point peut correspondre à un point de référence sur un objet non ponctuel de l'environnement [Hébert et al., 1996, Smith et al., 1988], par exemple l'angle d'un obstacle, perçu grâce à un télémètre laser [Borghi and Brugali, 1995].

Frontière des objets

Les frontières des différents objets et obstacles de l'environnement peuvent être directement représentées par des objets géométriques de plus haut niveau que des points. Des lignes ou des polygones sont très souvent utilisés. Ces objets sont extraits d'ensemble de points perçus par des capteurs à ultrasons [Dudek and MacKenzie, 1993, Gasós and Martín, 1997] ou des télémètres laser [Moutarlier and Chatila, 1990, Einsele, 1997, Castellanos et al., 1999]. Des cylindres et des plans, détectés par des capteurs à ultrasons sont aussi utilisés [Leonard et al., 1992], ainsi que des structures de plus haut niveau, comme des plans en trois dimension, détectés par stéréo-vision [Ayache and Faugeras, 1989].

Représentation de l'incertitude

Dans la plupart des systèmes, la manière dont est représentée et gérée l'incertitude est cruciale. L'incertitude concernant les objets mémorisés dans la carte est de deux types. Le premier concerne l'incertitude sur les paramètres des objets, par exemple sur leur position dans l'environnement. Ce type d'incertitude provient des erreurs de localisation du robot lors de la perception d'un objet, ou d'un bruit au niveau du capteur. Il est, dans la majorité des cas, représenté par la variance de paramètres considérés [Smith et al., 1988, Ayache and Faugeras, 1989, Moutarlier and Chatila, 1990, Leonard et al., 1992, Hébert et al., 1996, Feder et al., 1999, Castellanos et al., 1999]. Toutefois, d'autres méthodes peuvent être utilisées, par exemple des intervalles [Engelson and McDermott, 1992] ou des ensembles flous [Gasós and Martín, 1997].

Le second type d'incertitude se place à un niveau plus fondamental. Il porte sur la qualité de la correspondance entre la carte et l'environnement. Il mesure avec quelle confiance un objet présent dans la carte correspond effectivement à un objet de l'environnement. En effet, il peut arriver que des erreurs de perception fassent apparaître des objets qui n'existent pas dans l'environnement. Cette incertitude est caractéristique des environnements dynamiques, dans lesquels des objets sont susceptibles de se déplacer, d'apparaître ou de disparaître. Elle est gérée, pour une grande partie, au niveau des capteurs, les procédures permettant la détection d'objet à mémoriser étant conçues pour ignorer au maximum les éléments instables de l'environnement. Au niveau de la carte, la plupart des modèles traitent ce problème au moment de la mise à jour. Il est par exemple possible de supprimer les objets qui auraient dû être perçus, mais qui restent introuvables par le robot. Certain modèles toutefois modélisent explicitement cette incertitude au moyen d'un paramètre de crédibilité [Leonard et al., 1992]. Ce paramètre permet une plus grande tolérance aux accidents de perception en mesurant la fiabilité d'objets comme point de repère.

Représentation de l'espace libre

Le modèle le plus populaire pour ce type de représentation est celui de la *grille d'occupation* [Moravec and Elfes, 1985, Thrun, 1999, Yamauchi et al., 1999]. Dans ce modèle, l'environnement est entièrement discrétisé suivant une grille régulière avec une résolution spatiale très fine (cf. figure 1.7). Une probabilité d'occupation est associée à chaque élément de cette grille. Cette probabilité mesure la confiance dans le fait que l'espace correspondant dans l'environnement est effectivement occupé par un obstacle. L'avantage de cette représentation est qu'elle utilise directement les valeurs des capteurs de distance afin de mettre à jour les probabilités d'occupation des cellules. Elle permet donc de supprimer la phase d'extraction d'objets qui est souvent coûteuse en temps de calcul et soumise à une forte incertitude.

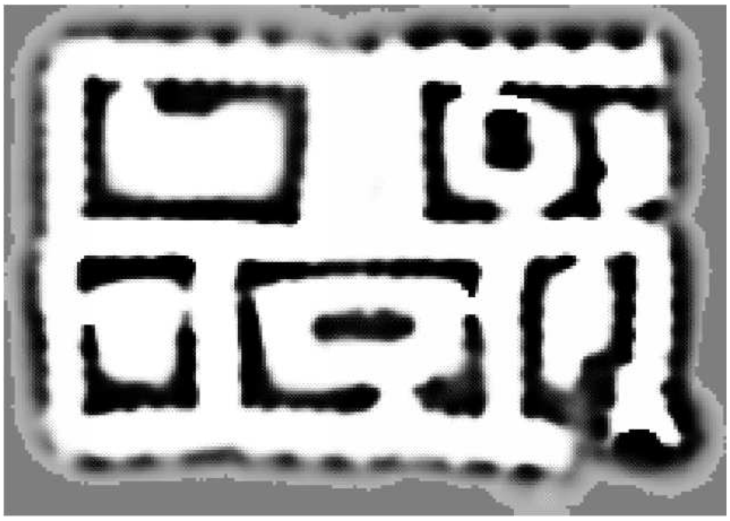


FIG. *1.7: Un exemple de grille d'occupation utilisée pour représenter un environnement. Les zones sombres indiquent une forte probabilité de présence d'un obstacle (Repris de [Thrun, 1999]).*

Les grilles d'occupation utilisent cependant une quantité de mémoire importante, qui croît proportionnellement à la surface de l'environnement. Pour s'affranchir de ce problème, certains modèles font appel à des discrétisations irrégulières de l'espace [Arleo et al., 1999]. De telles discrétisations permettent de s'adapter à la complexité de l'environnement, en représentant de manière grossière les grands espaces libres et plus finement les contours des obstacles.

1.4.3 Fusion de capteurs

Nous avons vu que les différentes données recueillies par les capteurs du robot sont mémorisées au sein d'une carte représentant l'environnement. Une source unique d'information étant en général insuffisante pour construire une représentation efficace et utilisable de l'environnement, cette carte est le lieu d'intégration des données de différents capteurs. Ces capteurs peuvent donner des informations très diverses, soit de type spatial (capteurs à ultrasons, télémètres, vision stéréoscopique), soit de type non spatial (température, couleur). Le problème de l'utilisation conjointe de tous ces capteurs pour la réalisation d'une tâche unique est abordé dans la littérature robotique sous le nom de *fusion de capteurs* [Luo and Kay, 1989]. Les cartes topologiques et métriques réalisent cette fusion de capteurs de deux manières différentes qui permettent de reformuler les avantages et les inconvénients des deux types de cartes.

En effet, les cartes métriques utilisent un cadre de référence global commun pour réaliser cette fusion. Chacun des capteurs utilisés doit donc être associé à un modèle métrique qui permet l'intégration de ces données dans le cadre commun. Cela limite donc les types de capteurs utilisables à ceux fournissant une information spatiale, pour lesquels un tel modèle peut être construit.

Les cartes topologiques, quant à elles, permettent l'utilisation de tous les types de capteurs car elles n'utilisent pas de cadre de représentation unique. La méthode de fusion est une méthode *d'association* qui permet de relier les données de différents capteurs qui correspondent à un même lieu. Cette méthode permet ainsi une fusion des données plus souple et plus générale que la méthode utilisée pour une carte métrique. La reconnaissance d'un lieu peut ainsi être réalisée en combinant facilement des critères autres que des critères spatiaux. Cette seconde technique peut être appelée *intégration de capteurs* [Borghi and Brugali, 1995]. Elle est plus adaptée au cas d'un animat pouvant posséder une large variété de capteurs car la gestion de chacun peut être réalisée indépendamment des autres.

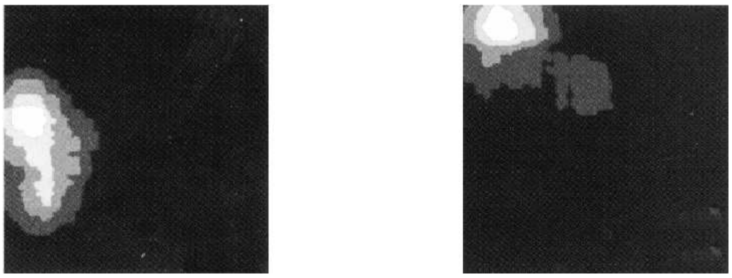
Cette facilité d'intégration est un des éléments qui nous ont conduits au choix d'une carte topologique pour concevoir le système de navigation décrit dans cette thèse.

1.5 Données biologiques sur la navigation

Les types de cartes décrits précédemment sont conçus dans le contexte général de la robotique mobile, en gardant à l'esprit les contraintes techniques ou informatiques correspondantes. Toutefois, certains de ces modèles s'inspirent des nombreuses données acquises par les neurobiologistes sur la manière dont les rats peuvent se représenter l'espace. Nous donnons ici un rapide aperçu de ces données.

1.5.1 Les cellules de lieu et de direction de la tête

L'étude du cerveau des rats lors de tâches de navigation a conduit à la découverte dans différentes zones, notamment dans l'hippocampe [O'Keefe and Dostrovsky, 1971], de neurones dont l'activité dépend uniquement de la position du rat dans son environnement (cf. figure 1.8). Cette activité est indépendante de tout autre facteur, tel que le comportement du rat, ou la récompense qu'il peut recueillir en ce point. Ces neurones, appelés *cellules de lieu*, ont une activité maximale lorsque le rat se trouve à une position donnée de l'environnement, généralement indépendamment de sa direction. De plus, les zones d'activation des différents neurones sont distribuées sur l'ensemble de l'environnement accessible au rat [O'Keefe and Conway, 1978]. Un tel type d'activité conduit à penser que le rat possède une représentation interne de son environnement, au sein de



Cellule 1 Cellule 2

FIG. *1.8: Deux exemples d'enregistrement de cellules de lieu sur des rats se déplaçant dans un environnement carré. Le niveau de gris indique l'activité d'une cellule de lieu en fonction de la position du rat dans l'environnement, du plus faible pour le noir au plus élevé pour le blanc. L'activité de chaque cellule est caractéristique d'une zone de l'environnement (repris de [Arléo, 2000]).*

laquelle il est capable d'estimer sa position. Ainsi, l'ensemble des cellules de lieu représenterait les différentes positions que le rat peut atteindre dans son environnement, tandis que leur activité indiquerait si le rat se trouve ou non à la position correspondante.

En plus de cette représentation de la position, des neurones dont l'activité dépend de la direction absolue de la tête du rat par rapport à l'environnement ont été découverts dans d'autres zones du cerveau. L'activité de ces *cellules de direction de la tête* ne dépend pas de la position de la tête du rat relativement à son corps. Il s'ensuit donc que les codages de la position et de la direction absolue du rat dans l'espace sont réalisés par des circuits différents.

1.5.2 Informations utilisées

L'expérience montre que l'activité des cellules de lieu et de direction de la tête dépend d'amers visuels situés dans l'environnement, qui permettent au rat de reconnaître une position ou une direction [Muller and Kubie, 1987]. Tous les points perceptibles dans l'environnement ne sont cependant pas utilisés pour ce faire. L'activité des neurones semble dépendre de préférence des amers stables et lointains de l'environnement. Certaines expériences montrent que les rats utilisent également d'autres types de perceptions, comme le toucher, ou l'odorat.

De plus, ces neurones conservent leur activité reliée à la position durant le déplacement du rat dans le noir, ou après suppression de tout amer de l'environnement [Quirk et al., 1990]. Ce comportement indique que le cerveau du rat utilise des informations idiothétiques sur le déplacement pour maintenir l'activité des cellules de lieu, en plus des informations visuelles. Ces informations peuvent provenir de l'oreille interne qui mesure l'accélération du rat par rapport à son environnement, ou de la mesure du mouvement des pattes.

1.5.3 Représentations biologiquement plausibles

Différents systèmes de navigation modélisant le fonctionnement de l'hippocampe des rats ont été conçus [Arleo and Gerstner, 2000, Balakrishnan et al., 1999, Burgess et al., 1994, Sharp, 1991, Touretzky et al., 1994]. Ces modèles utilisent tous une carte topologique de l'environnement dans laquelle chacun des nœuds de la carte correspond à une cellule de lieu. Cette représentation semble en effet très adaptée à une telle modélisation car, comme nous l'avons mentionné dans le paragraphe précédent, les cartes topologiques réalisent une fusion des données de différents capteurs. Cette fusion se réalise au niveau de chaque nœud sans utilisation d'un cadre de référence global qui semble difficile à concevoir dans un cadre biologique. Cette méthode de fusion des données par association semble, de plus, cognitivement similaire à d'autres fonctions de l'hippocampe, telle que de la mémoire [Marr, 1971].

De plus, l'utilisation de concepts de relativement haut niveau des cartes métriques semble inadaptée aux capacités cognitives d'un rat. La mémorisation brute des données utilisées dans les cartes topologiques semble, au contraire, plus réalisable par de simples processus d'association et de mémorisation.

La définition non supervisée des nœuds (cf. section 1.4) permet la création autonome de la carte avec très peu d'a priori sur l'environnement, d'une manière similaire à ce que peut faire un rat. Les cartes obtenues permettent de reproduire certaines des caractéristiques des cellules de lieu. Elles couvrent par exemple l'environnement de manière uniforme et il est possible d'associer par des méthodes relativement simples une activité à chacun des nœuds qui reproduit l'activité des cellules de lieu. De plus, cette activité dépend de différents facteurs, tels que la vision et l'odométrie, de manière similaire à la dépendance des cellules de lieu.

Notre modèle, sans prétendre à la plausibilité biologique, utilisera des structures similaires. Il sera donc compatible avec un certain nombre de données biologiques, mais s'écartera des modèles biologiquement plausibles, notamment à propos de la méthode utilisée pour le calcul des activités des nœuds de la carte.

1.6 Conclusion

La carte que nous avons choisi d'utiliser dans notre modèle est une carte topologique. Ce choix découle du fait que nous souhaitons pouvoir utiliser, au sein d'une même représentation, les informations provenant de nombreux types de capteurs, sans nous limiter aux capteurs pour lesquels nous disposons d'un modèle métrique. Les cartes topologiques, en associant les données de différents capteurs perçues en un même lieu permettent une telle intégration. De plus, nous souhaitions éviter le recours aux modèles métriques en raison des difficultés de mise en œuvre et de calibration auxquelles ils sont soumis.

Un autre avantage des cartes topologiques est qu'elles représentent de manière distincte les données idiothétiques et allothétiques, sans les fusionner dans un cadre unique comme le font les cartes métriques. Nous verrons dans le chapitre 3 que cette propriété est intéressante pour permettre une cartographie efficace.

Pour permettre une représentation relativement précise de la position du robot au sein de son environnement, la carte topologique de notre modèle est une carte très dense dont les nœuds correspondent à des lieux qui sont très proches les uns des autres. Cela permet de limiter la perte de précision due à l'utilisation d'une carte topologique par rapport à une carte métrique. Enfin, les arêtes de cette carte mémorisent une information sur la position métrique relative des nœuds qu'elles relient, sans qu'aucune information métrique de position globale, qui est difficile à estimer, ne soit mémorisée.

Tous ces choix découlent également de la volonté de créer un système dans lequel des éléments conceptuellement similaires aux cellules de lieu de l'hippocampe des rats sont identifiables. Les nœuds des cartes topologiques, qui représentent des *positions* de l'environnement - contrairement aux cartes métriques qui représentent des *objets* de l'environnement - sont particulièrement pertinents dans cette logique. La forte densité de la carte topologique utilisée rappelle, de plus, la densité élevée des cellules de lieu observées chez les rats.

Chapitre 2

Localisation

Dans ce chapitre, nous allons présenter plus précisément le problème de la localisation et différentes méthodes possibles pour le résoudre. Nous nous attacherons uniquement à ce problème et nous le découplerons de celui de la cartographie en considérant qu'une carte complète et précise de l'environnement est disponible pour l'animat. Le problème est alors uniquement de déterminer quelle position au sein de cette carte correspond le plus précisément à la position de l'animat dans son environnement réel. Nous verrons que le terme «localisation» recouvre en fait deux catégories d'algorithmes de capacités très différentes. Nous verrons également qu'il est possible d'utiliser de manière plus ou moins optimale les deux sources d'information accessibles à l'animat pour résoudre cette tâche. Les modèles mentionnés dans ce chapitre sont présentés plus en détail dans une publication [Filliat and Meyer, 2001a].

2.1 Différentes capacités de localisation

Il existe deux types de capacités regroupées sous le terme «localisation». La première, appelée *suivi de position*, est la capacité de mettre à jour une estimation existante de la position au vu de données idiothétique ou allothétiques nouvellement acquises. Dans le cas des données idiothétiques, cette mise à jour concerne un déplacement du robot et va en général diminuer la précision de l'estimation courante de la position. Dans le cas de données allothétiques, au contraire, cette mise à jour va en général permettre d'améliorer cette estimation grâce au lien avec l'environnement fourni par ces données. L'utilisation de cet ancrage dans l'environnement est fondamental pour assurer que l'estimation de la position reflète correctement la position de l'animat dans l'environnement réel. Cette mise à jour intégrant les deux types de données permet de combiner les avantages inhérents aux deux types d'information afin d'estimer au mieux la position de l'animat. En pratique, toutefois, le suivi de position est problématique car il repose sur une estimation initiale de la position qui doit souvent être fournie par une source extérieure. De plus, si la position estimée s'écarte trop de la position réelle du robot, l'animat peut très bien ne pas parvenir à corriger l'erreur et être définitivement perdu.

La seconde capacité de localisation, la *localisation globale*, est plus générale et permet de retrouver la position de l'animat sans qu'aucune estimation initiale ne soit fournie. Cette capacité est très importante du point de vue de l'autonomie, car elle permet au robot de retrouver sa position, dans toutes les conditions, sans intervention extérieure. Elle permet, par exemple, de couper l'alimentation d'un robot à des fins de maintenance, puis de remettre ce robot dans une position quelconque de l'environnement sans se soucier d'initialiser correctement son estimation de la position.

Ces deux capacités de localisation ont des propriétés duales. Comme le note Piasecki [Piasecki, 1995], dans le contexte d'une carte métrique, le suivi de position est une méthode locale, continue, qui effectue régulièrement de petites corrections à l'estimation de la position de l'animat. Cette méthode effectue de telles corrections en se basant sur des objets de l'environnement et la manière dont ils ont été perçus par le robot. L'identification de ces objets est de plus simplifiée grâce à l'estimation initiale de la position qui permet, en cas de perceptual aliasing, de décider quel est l'objet qui a été perçu parmi les différents objets correspondant aux perceptions.

Au contraire, la localisation globale est une méthode globale, discontinue, qui effectue exceptionnellement des corrections de grande ampleur de la position estimée. Sa première tâche, avant d'estimer une position, est de déterminer à quels objets de l'environnement correspond chacune des perceptions du robot. Cette tâche peut être très simple dans le cas où la carte ne contient que des amers différents, mais est en général assez complexe car plusieurs amers sont identiques à cause du *perceptual aliasing*.

Comme nous l'avons souligné, le suivi de position permet de fusionner et de tirer parti des informations idiothétiques et allothétiques disponibles pour le robot. Cette méthode ne conduit toutefois qu'à une estimation qui est localement la meilleure approximation possible de la position. En effet, la recherche est contrainte par l'estimation précédente de cette position. La position estimée sera donc celle qui est la plus en accord avec les données recueillies, *dans le voisinage de cette estimation précédente*. L'estimation résultante peut très bien ne pas correspondre à la position qui, sur l'ensemble de la carte, correspond le mieux aux données.

En principe, la localisation globale permet une telle estimation optimale. Au niveau de l'utilisation des données disponibles pour le robot, il existe de nouveau deux classes de méthodes de localisation globale :

- La première, qui ne fonctionne que dans des environnements où il n'existe aucun *perceptual aliasing*, fait appel uniquement aux données allothétiques disponibles en une position donnée. Nous l'appellerons *inférence directe de position*.
- La seconde, qui fonctionne dans tous les environnements, fusionne au contraire les informations idiothétiques et allothétiques, comme le fait le suivi de position. Toutefois, au lieu de restreindre la recherche par une estimation précédente de la position, elle estime parmi toutes les positions possibles au sein de la carte celle qui correspond le mieux aux données présentes et passées recueillies par le robot. Les méthodes de cette catégorie reposent, d'une façon ou d'une autre, sur le *suivi de plusieurs hypothèses* de position, ce qui permet

de généraliser le suivi de position en utilisant de manière plus efficace les informations idiothétiques et allothétiques disponibles.

Dans la suite de ce chapitre, nous allons présenter plus précisément les différentes méthodes de localisation. Nous allons d'abord voir comment il est possible d'estimer la position d'un animat au vu des seules données allothétiques. Puis, dans le cas de systèmes perceptifs soumis au *perceptual aliasing*, nous verrons comment il est possible d'intégrer les informations idiothétiques, de manière locale puis globale, afin de lever les ambiguïtés restantes.

2.2 Stratégies de localisation

2.2.1 Estimation de la position par les données allothétiques

Dans ce paragraphe, nous résumons les différentes méthodes qui peuvent être utilisées pour estimer la position d'un animat à l'aide des seules données allothétiques. Dans le cas où l'environnement est exempt de *perceptual aliasing*, cette étape suffit à déterminer la position de l'animat de manière unique. Cette méthode est alors la première méthode de localisation globale mentionnée précédemment. Dans le cas où le *perceptual aliasing* est présent, ces méthodes sont également utilisées mais elles serviront à repérer plusieurs positions possibles pour l'animat au sein de l'environnement. Le suivi de position ou la seconde méthode de localisation globale qui seront présentés dans les paragraphes suivants doivent alors être utilisés en sus pour sélectionner la position correcte.

Dans le cas des cartes topologiques, estimer la position à partir des seules données allothétiques est extrêmement simple. En effet, parmi tous les lieux représentés dans la carte, la position du robot est celle d'un des nœuds qui correspond le mieux aux perceptions courantes. La recherche de ce nœud passe donc par la comparaison des perceptions de l'animat avec les perceptions mémorisées dans chacun des nœuds de la carte. Les nœuds qui sont identiques ou suffisamment similaires sont alors reconnus comme positions possibles du robot.

Lorsque les animats ne sont pas soumis au *perceptual aliasing*, tous les nœuds de la carte correspondent à des situations différentes. Cette étape est alors suffisante pour la localisation complète de l'animat car le nœud reconnu est unique. Différents systèmes perceptuels ont été utilisés pour implémenter de tels modèles. Certains auteurs utilisent des images panoramiques de l'environnement pour définir les nœuds de la carte [Kortenkamp and Weymouth, 1994, Franz et al., 1998]. D'autre modèles utilisent les directions ou les distances d'amers ponctuels tous discernables, soit en simulation [Burgess et al., 1994, Levitt and Lawton, 1990, Trullier and Meyer, 2000, Sharp, 1991], soit sur des robots réels [Bachelder and Waxman, 1994, Gaussier et al., 2000].

Lorsqu'une position dans un espace métrique est associée à chacun des nœuds de la carte, la localisation permet en outre de déterminer la position métrique du robot. Cette position peut simplement être la position du nœud reconnu, mais il est souvent possible d'obtenir une précision supplémentaire. En effet, au lieu de tenir simplement compte du nœud le plus conforme aux perceptions courantes, il est possible de tenir compte de chacun des nœuds, selon son degré de similarité avec ces perceptions. La méthode mise en œuvre dans de tels modèles pour réaliser cette estimation de position est le *codage par population de vecteurs* [Georgopoulos et al., 1986]. Cette méthode consiste à estimer la position de l'animat par la moyenne des positions des différents nœuds, pondérées par le degré de similarité de chaque nœud avec les perceptions de l'animat. Cette méthode donne une estimation précise de la position du robot, mais suppose une relative continuité de l'environnement. Elle suppose en effet que des lieux similaires seront proches les uns des autres pour que la moyenne des positions ait un sens. Les données allothétiques doivent donc varier de manière continue avec la position.

Lorsque les modèles permettent la gestion du *perceptual aliasing* (par une des méthodes décrites dans les paragraphes suivants), les lieux peuvent également être définis par des images panoramiques de leur environnement [Arleo and Gerstner, 2000, Egerton and Callaghan, 2000, Hafner, 2000, Ulrich and Nourbakhsh, 2000, Radhakrishnan and Nourbakhsh, 1999, Von Wichert, 1998], ou par la configuration des positions d'amers distants [Balakrishnan et al., 1999, Touretzky et al., 1994]. Mais, puisque le *perceptual aliasing* sera géré par ailleurs, des définitions plus simples des nœuds peuvent également être adoptées, au prix d'une moins grande discrimination. Certains modèles utilisent ainsi les valeurs brutes de capteurs de distance [Nehmzow and Owen, 2000, Mataric, 1992, Kuipers and Byun, 1991, Hertzberg and Kirchner, 1996, Hafner, 2000, Oore et al., 1997], ou la configuration des murs autour du robot afin de détecter des angles de couloirs ou des embranchements [Kunz et al., 1997, Dedeoglu et al., 1999, Shatkay and Kaelbling, 1997, Cassandra et al., 1996, Nourbakhsh et al., 1995, Simmons and Koenig, 1995, Theocharous et al., 2001].

Dans le cas des cartes métriques, diverses méthodes d'estimation de la position existent. Lorsque les données allothétiques sont constituées d'amers ponctuels, une méthode de triangulation peut être utilisée [Betke and Gurvits, 1994, Greiner and Isukapalli, 1996, Levitt and Lawton, 1990, Madsen et al., 1997]. Cette méthode repose sur la mesure de la direction et de la distance d'amers ponctuels connus. La perception de trois amers de ce type permet en effet de définir la position du robot de manière unique. Un simple calcul mathématique permet donc de calculer cette position à partir des positions des amers. Ce calcul peut également être approximé par des réseaux de neurones, [Prescott, 1995], ou par des méthodes heuristiques qui permettent une meilleure résistance au bruit [Wijk and Christensen, 2000]. Lorsque cette méthode est utilisée avec des cartes ne comportant pas de *perceptual aliasing*, chaque amer est unique et cette méthode permet d'estimer directement de manière non ambiguë la position du robot. En cas de *perceptual aliasing*, certains amers ne peuvent être distingués et il faut tenir compte de l'estimation précédente de la position afin de pouvoir identifier correctement les différents amers et estimer correctement la position.

Certains types d'objets fournissent plus d'information que des amers ponctuels, sans toutefois permettre une estimation non ambiguë de la position. Par exemple, c'est le cas des murs dont la

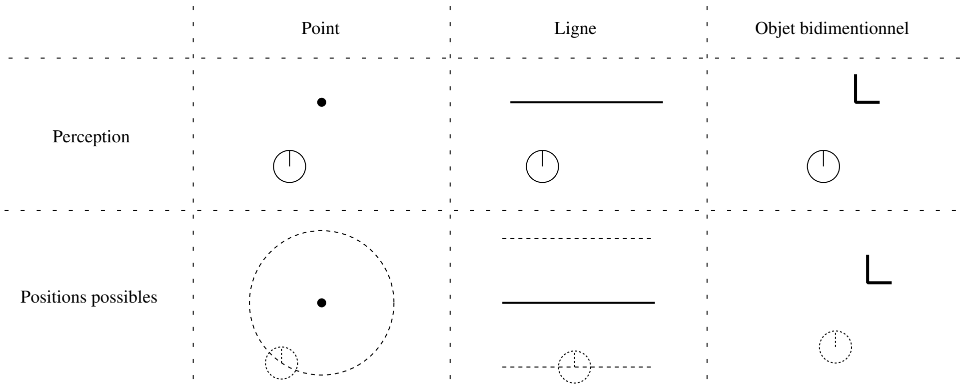


FIG. *2.1: Les amers que peut repérer un robot fournissent plus ou moins d'information sur sa position en 2 dimensions. La détection d'un amer ponctuel permet de savoir que le robot se trouve sur un cercle entourant l'amer. Un amer rectiligne permet de connaître la distance du robot perpendiculairement à cet amer, mais pas sa position le long de cet amer. Enfin, un amer ayant une étendue spatiale en deux dimensions permet de définir la position du robot de manière unique.*

perception fournit une information sur la distance du robot à ce mur, mais pas sur sa position le long de ce mur (cf. figure 2.1). Certains modèles utilisent de tels types d'objets, qui permettent d'affiner une estimation précédente de la position, mais pas d'estimer directement cette position [Castellanos et al., 1999, Jensfelt and Kristensen, 1999, Saffiotti and Wesley, 1995].

Lorsque les objets mémorisés dans la carte ont une certaine étendue spatiale en deux dimensions, il est par contre possible d'utiliser la perception d'un seul objet afin d'estimer directement la position du robot. Les amers utilisés peuvent alors être des objets tridimensionnels détectés par une caméra [Sim and Dudek, 1999], les angles des obstacles détectés par un télémètre laser [Arsenio and Ribeiro, 1998, Borghi and Brugali, 1995, Gomes-Mota and Ribeiro, 2000, Jensfelt and Kristensen, 1999] ou un capteur à ultrasons [Leonard et al., 1992, Saffiotti and Wesley, 1995], des segments détectés en utilisant une caméra [Ayache and Faugeras, 1989] ou un télémètre laser [Castellanos et al., 1999, Cox, 1991, Moutarlier and Chatila, 1990].

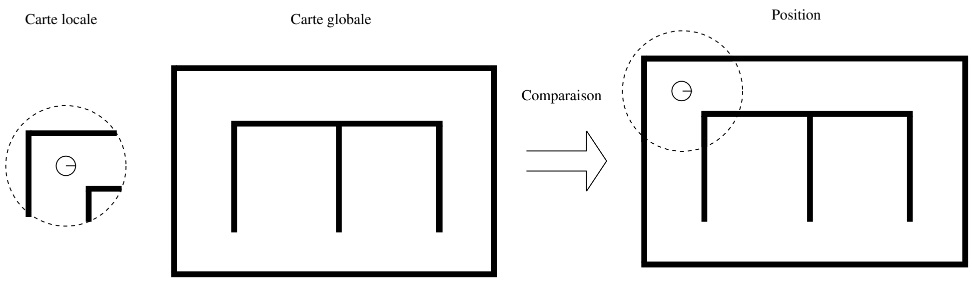


FIG. *2.2: Pour estimer la position d'un animat, il est possible de construire une carte locale représentant l'environnement proche de la position courante. La comparaison de cette carte locale et de la carte globale de l'environnement permet alors de trouver la position de l'animat.*

D'autre modèles, enfin, n'estiment pas directement la position du robot au vu des données allothétiques, mais reposent sur la comparaison d'une carte métrique locale avec la carte métrique globale (cf. figure 2.2). La carte métrique locale est construite soit à partir des seules données allothétiques courantes, soit à partir des données idiothétiques et allothétiques recueillies sur un court laps de temps. Le problème est alors de trouver la portion de carte globale qui correspond le mieux à la carte locale. Cette méthode est très souvent utilisée avec les grilles d'occupation [Olson, 2000, Schiele and Crowley, 1994, Schultz and Adams, 1998, Thrun, 1999], ainsi qu'avec des données brutes de télémètres laser [Lu and Milios, 1997, Gutmann and Schlegel, 1996, Einsele, 1997]. Le *polygone de visibilité*, qui entoure la zone d'espace libre visible depuis la position courante du robot peut aussi être utilisé [Guibas et al., 1995, Karch and Wahl, 1999]. Comme nous le verrons dans la section suivante, ces méthodes sont souvent utilisées sur un espace de recherche restreint par une estimation initiale de la position. Elles peuvent cependant être utilisées pour la localisation globale [Olson, 2000, Guibas et al., 1995, Karch and Wahl, 1999].

L'hypothèse d'un environnement sans *perceptual aliasing* est relativement forte, car beaucoup de capteurs en robotique sont limités et fortement bruités. De plus, les environnements intérieurs, de type bureaux, peuvent être très réguliers et présenter de nombreuses zones apparemment similaires pour le robot. Toutefois, les environnement courants contiennent souvent suffisamment d'informations accessibles à des capteurs précis et efficaces. Un être humain, par exemple, n'a aucun mal à se repérer dans un immeuble de bureaux, en lisant les numéros écrits sur les portes (trouver son chemin jusqu'à la sortie est un autre problème !). Il est donc théoriquement possible de concevoir des systèmes suffisamment discriminants pour être capables de se repérer grâce aux seules données allothétiques. Il est de plus possible d'aménager l'environnement afin de simplifier la tâche de perception pour le robot (comme le montre l'exemple des numéros de porte). Cette solution nuit toutefois fortement à l'autonomie du robot puisqu'il est alors limité aux environnements bien définis qui ont été préparés à l'avance. Le rejet de cette dernière solution et la difficulté de réaliser des capteurs suffisamment discriminant conduit donc la plupart des systèmes de navigation robotique à prendre en compte le *perceptual aliasing* et à utiliser les données idiothétiques pour déterminer leur position de manière unique. Nous allons décrire les différentes méthodes qui peuvent être utilisées dans la suite de ce chapitre.

2.2.2 Suivi d'une hypothèse unique

Lorsque les données allothétiques ne suffisent pas pour estimer la position de manière unique, une seconde source d'estimation de la position du robot est nécessaire pour lever l'ambiguïté. Cette seconde estimation provient, d'une part de la position déterminée lors de la précédente phase de localisation et, d'autre part, des données idiothétiques recueillies depuis cet instant. Les méthodes présentées dans cette section utilisent cette seconde estimation pour sélectionner, à chaque instant, la position qui est la plus cohérente vis-à-vis de cette estimation. Les positions estimées grâce aux données allothétiques qui ne sont pas compatibles avec la position précédente sont simplement ignorées.

Dans une carte topologique, sélectionner le nœud correct parmi les nœuds correspondant aux données allothétiques peut reposer simplement sur l'adjacence avec le nœud précédent. Dans ce cas, le nœud sélectionné est celui qui est connecté au nœud représentant la position précédente. Cette information est toutefois rarement suffisante et les relations métriques mémorisées dans les arêtes entre nœuds sont souvent utilisées en complément. Le nœud sélectionné est donc celui dont la position relative par rapport au nœud précédent correspond le mieux aux données idiothétiques [Kuipers and Byun, 1991, Kunz et al., 1997, Dedeoglu et al., 1999, Nehmzow and Owen, 2000]. Lorsqu'une position métrique est associée à chaque nœud, c'est le nœud dont la position est la plus proche de la position estimée par l'odométrie qui est sélectionné [Balakrishnan et al., 1999, Kurz, 1995, Yamauchi and Beer, 1996].

Certains modèles fonctionnent dans le sens opposé. Au lieu d'utiliser les données idiothétiques pour sélectionner un nœud parmi les nœuds possibles, ils utilisent ces données pour restreindre l'ensemble des nœuds possibles et utilisent ensuite les données allothétiques pour sélectionner le nœud correct parmi ceux-ci. Les données allothétiques sont, par exemple, utilisées pour choisir un nœud parmi tous les nœuds adjacents au nœud précédent [Ulrich and Nourbakhsh, 2000], ou parmi les nœuds suffisamment proches de la position estimée par l'odométrie [Von Wichert, 1998].

Enfin, certains modèles intègrent les deux étapes en une seule. Un paramètre mesure la probabilité que chaque nœud représente la position courante. Ce paramètre intègre, d'une part, la similarité du nœud avec la situation allothétique courante, et d'autre part sa proximité avec la position estimée par l'odométrie. Le nœud ayant la plus forte probabilité peut alors être reconnu [Mataric, 1992], ou la position peut être estimée par codage par population de vecteurs en utilisant les probabilités calculées [Arleo and Gerstner, 2000, Touretzky et al., 1994].

Dans une carte métrique, l'estimation initiale de la position est utilisée pour restreindre l'espace de recherche de la position correspondant aux données allothétiques. Dans le cas où la carte contient des objets, une estimation de la position permet de simplifier le problème de l'appariement. En effet, dans le cas où les senseurs sont soumis à un fort *perceptual aliasing*, de nombreux objets identiques, situés à des positions différentes, sont présents dans la carte. Lorsque le robot perçoit un objet, déterminer quel objet a été perçu exige d'examiner un grand nombre de possibilités. L'estimation de la position du robot permet donc d'estimer la position des objets perçus et donc de déterminer à quels objets de la carte ils correspondent. Ce choix se fait en général en appariant simplement chaque objet perçu à l'objet mémorisé le plus proche [Ayache and Faugeras, 1989, Castellanos et al., 1999, Cox, 1991, Dudek and MacKenzie, 1993, Gomes-Mota and Ribeiro, 2000, Leonard et al., 1992, Moutarlier and Chatila, 1990, Smith et al., 1988, Wijk and Christensen, 2000]. Une fois l'appariement effectué, les objets sont identifiés sans ambiguïté et permettent donc d'estimer la position de manière unique.

Lorsque la position est déterminée par la mise en correspondance d'une carte locale et d'une carte globale, la position estimée est utilisée pour restreindre la recherche de la position donnant la meilleure correspondance entre les deux cartes [Schiele and Crowley, 1994, Schultz and Adams, 1998, Thrun, 1999, Yamauchi et al., 1999]. La recherche du maximum de correspondance est simplement effectuée sur une zone limitée autour de la position estimée précédemment. La zone étant de faible étendue, le problème de *perceptual aliasing* ne se pose plus et la recherche conduit à une position unique.

Lorsque la position correspondant aux données allothétiques a été identifiée de manière unique, elle peut être considérée directement comme la nouvelle estimation de la position du robot [Yamauchi et al., 1999, Gomes-Mota and Ribeiro, 2000, Wijk and Christensen, 2000]. Cependant la plupart des modèles considèrent que cette estimation est entachée d'erreur, de la même manière que l'estimation initiale. La nouvelle position du robot est donc en général une combinaison de ces deux positions. La plupart des modèles [Ayache and Faugeras, 1989, Smith et al., 1988, Moutarlier and Chatila, 1990, Cox, 1991, Leonard et al., 1992, Schiele and Crowley, 1994, Betke and Gurvits, 1994, Borghi and Brugali, 1995, Lu and Milios, 1997, Castellanos et al., 1999] utilisent un *filtre de Kalman* [Maybeck, 1979] pour réaliser cette combinaison. Ce filtre permet de calculer une estimation optimale de la position du robot, connaissant les deux positions et leurs variances respectives. D'autres méthodes sont également utilisables pour combiner ces deux informations, par exemple la minimisation d'une fonction de coût reliée à ces deux positions [Thrun, 1999], ou l'utilisation de la méthode des *moindres carrés récursifs* [Boley et al., 1996].

La limitation principale de toutes ces méthodes de suivi de position est qu'elles ne garantissent une bonne estimation de la position que localement, autour de l'estimation initiale de la position. En pratique, si cette estimation initiale est trop éloignée de la position réelle, ces modèles ne pourront pas estimer correctement la position du robot (cf. figure 2.3). Ces modèles ne garantissent donc pas que la position calculée soit la position de la carte qui corresponde globalement le mieux aux données recueillies par le robot et donc la position réelle la plus probable.

Ce problème prend toute son importance lorsque l'estimation de la position est perturbée à la suite d'informations idiothétiques ou allothétiques erronées. En effet, de telles informations erronées peuvent faire diverger l'estimation de la position de telle manière que le système soit par la suite incapable de retrouver une estimation correcte de la position. L'animat est alors perdu tant qu'une nouvelle estimation correcte de la position ne lui est pas fournie.

La position correcte de l'animat pourra cependant être retrouvée par l'une des méthodes de localisation globale décrites dans le paragraphe précédent. Cette position pourra ensuite être utilisée comme nouvelle position initiale dans le processus de suivi de position. Toutefois, il est également possible d'utiliser une des méthodes de localisation globale décrites dans le paragraphe suivant qui permettent de ne plus dépendre d'une estimation initiale correcte de la position. Cette seconde solution, qui ne requiert pas l'utilisation séparée de deux méthodes de localisation semble plus performante.

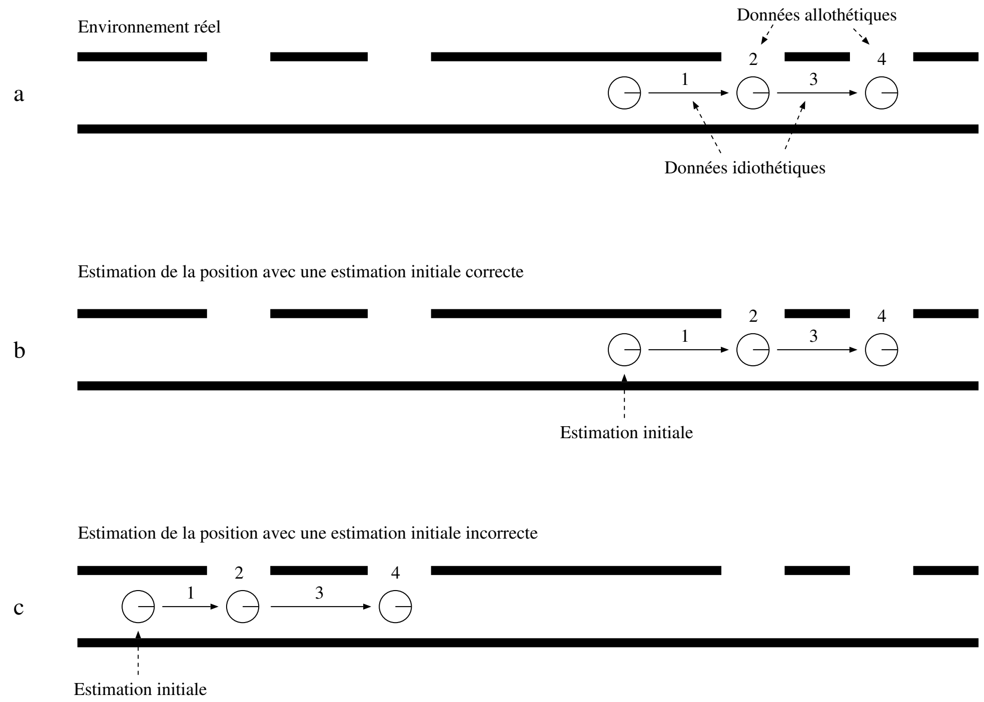


FIG. *2.3: La position estimée par le suivi de position dépend fortement de l'estimation initiale de la position. Dans cet exemple, le robot mesure son déplacement dans un couloir (1 et 3) dans lequel il détecte des portes (2 et 4) sans être capable de les reconnaître individuellement (Partie a). Le système de localisation va estimer la position du robot en intégrant ces données. Si l'estimation initiale de la position est proche de la position correcte, le système de localisation sera capable d'estimer précisément la position réelle du robot (Partie b). Toutefois, si l'estimation initiale de la position est trop éloignée de la position réelle, le système fournira une estimation de la position qui n'est que localement optimale et ne correspondra pas à la position réelle (Partie c).*

2.2.3 Suivi de plusieurs hypothèses

La localisation globale, lorsque l'animat est soumis au *perceptual aliasing*, ne peut se faire qu'en utilisant de manière optimale les informations idiothétiques et allothétiques. Contrairement au suivi de position qui utilise l'estimation précédente de la position pour sélectionner l'une des positions caractérisée par les données allothétiques et ignorer les autres, il faut tenir compte à chaque étape de toutes ces positions possibles. Ces positions conduisent à des hypothèses qui peuvent être mises à jour en fonction des données idiothétiques et qu'il faut comparer, afin de choisir la plus pertinente à chaque étape (cf. figure 2.4).

Ce suivi de plusieurs hypothèses peut être réalisé de manière explicite, en gérant une liste des hypothèses en question. Lorsque des données idiothétiques sont disponibles, chaque hypothèse est simplement mise à jour de manière à refléter le déplacement du robot (cf. figure 2.4c). Lorsque de nouvelles données allothétiques sont disponibles, l'ensemble des positions de la carte susceptibles de correspondre à ces perceptions est déterminé. Cet ensemble est ensuite comparé

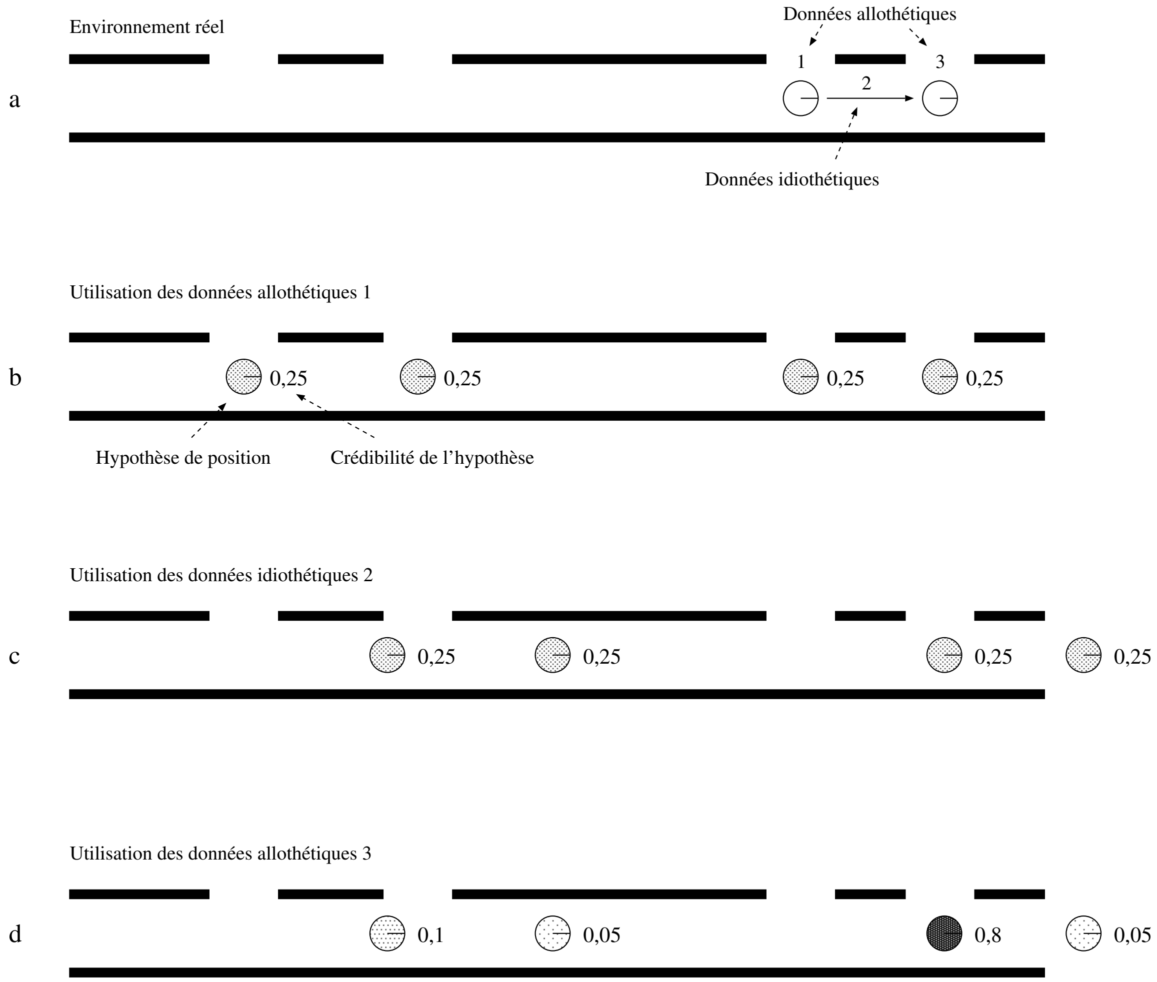


FIG. *2.4: Le suivi de plusieurs hypothèses permet de déterminer la position au sein de la carte qui globalement correspond le mieux aux données recueillies par le robot. Dans cet exemple, le robot est capable d'estimer son déplacement dans un couloir (2) dans lequel il détecte des portes (1 et 3) sans être capable de les reconnaître individuellement (Partie a). La perception d'une porte sans aucune estimation préalable de la position permet simplement de créer plusieurs hypothèses de position pouvant correspondre à cette perception. Il est, à ce stade impossible de décider quelle hypothèse est correcte (Partie b). L'intégration des données idiothétiques permet de mettre à jour la position de chacune des hypothèses, mais ne permet pas de les discriminer (Partie c). Des nouvelles perceptions permettent d'estimer la crédibilité relative de chacune des hypothèses en rendant plus crédibles les hypothèses dont la position correspond aux perceptions courantes et moins crédible les autres. (Partie d). L'hypothèse ayant alors la plus forte crédibilité correspond à la position qui rend le mieux compte des données recueillies par le robot.*

à l'ensemble des hypothèses. Si une hypothèse correspond à une position perçue, cette hypothèse est alors mise à jour en utilisant les données allothétiques par une méthode similaire à celle permettant le suivi de position, par exemple un filtre de Kalman. Les positions perçues qui ne correspondent à aucune hypothèse sont utilisées pour créer de nouvelles hypothèses associées à la position correspondante. La crédibilité de chacune des hypothèses est ensuite évaluée, généralement en fonction de la proximité de l'hypothèse avec une position correspondant aux perceptions du robot (cf. figure 2.4d). Ainsi, une hypothèse verra sa crédibilité augmenter si elle est proche d'une des positions correspondant aux perceptions du robot, elle la verra diminuer dans le cas contraire. De tels modèles ont été implémentés en utilisant des cartes topologiques [Duckett and Nehmzow, 1998, Donnart and Meyer, 1996] ainsi que des cartes métriques [Piasecki, 1995, Jensfelt and Kristensen, 1999].

Une seconde méthode permet d'intégrer de manière similaire les deux types d'informations, mais ne gère pas explicitement les hypothèses de position. Les différentes hypothèses sont ici remplacées par une distribution de probabilité de présence sur l'ensemble des positions possibles de la carte. Cette représentation permet donc de considérer chacune des positions au sein de la carte comme une position possible du robot dont il faut évaluer la crédibilité. Dans des cartes topologiques, ces positions possibles sont simplement les différents nœuds de la carte [Cassandra et al., 1996, Hafner, 2000, Hertzberg and Kirchner, 1996, Kortenkamp et al., 1994, Nourbakhsh et al., 1995, Oore et al., 1997, Shatkay and Kaelbling, 1997, Simmons and Koenig, 1995, Theocharous et al., 2001]. Pour les cartes métriques, il est possible de discrétiser l'ensemble des positions, à la manière des grilles d'occupation [Burgard et al., 1996, Fox et al., 1998b, Thrun et al., 1999], et d'associer une probabilité de présence du robot à chacun des éléments de la discrétisation. Il est également possible d'utiliser une méthode qui permette d'approximer directement une distribution de probabilité continue sur l'espace de la carte [Fox et al., 1999]. Enfin, il est possible d'utiliser des ensembles flous pour représenter la distribution de crédibilité des différentes positions [Saffiotti and Wesley, 1995].

Une telle distribution de probabilité permet d'intégrer la totalité des informations recueillies par le robot. Elle est mise à jour, d'une part à chaque déplacement du robot, et donc à chaque nouvelle donnée idiothétique, et d'autre part à chaque nouvelle donnée allothétique. De manière imagée, les données idiothétiques permettent de déplacer les probabilités d'une position à une autre pour refléter le déplacement du robot. Les données allothétiques permettent de moduler les probabilités de chaque position. Ainsi, les positions pour lesquelles les données allothétiques sont similaires aux données perçues voient leurs probabilités augmentées, tandis que les autres voient leurs probabilités diminuées. Ces opérations peuvent être modélisées très naturellement dans le cadre des Processus de Décision Markoviens Partiellement Observables (PDMPO). Nous décrirons ce formalisme en détail dans la prochaine section.

Quoi qu'il en soit, lorsqu'on utilise une telle distribution de probabilité, la position du robot est en général donnée par l'hypothèse ayant la plus forte probabilité [Hafner, 2000, Hertzberg and Kirchner, 1996, Shatkay and Kaelbling, 1997, Simmons and Koenig, 1995, Burgard et al., 1996, Fox et al., 1998b]. Cependant d'autres estimations peuvent être utilisées, similaires au codage par population de vecteurs [Cassandra et al., 1996].

La distribution de probabilités obtenue contient donc implicitement toutes les hypothèses possibles sur la position du robot. La probabilité de chacune des positions est, de plus, estimée en utilisant toutes les informations disponibles. Cette probabilité dépend faiblement des conditions initiales et peut donc être initialisée à une distribution uniforme lorsqu'aucune information n'est disponible sur la position du robot. La position sera alors retrouvée, même si le robot est soumis à un très fort *perceptual aliasing*, assurant ainsi la localisation globale du robot dans tous les environnements. Cette méthode est extrêmement robuste en pratique et met en œuvre un système de localisation complètement autonome, ne dépendant d'aucune intervention extérieure [Thrun et al., 1999].

Ces qualités reposent toutefois de manière importante sur le fait que la carte de l'environnement est complète. En effet, les systèmes de suivi de plusieurs hypothèses nécessitent une estimation correcte des crédibilités des différentes positions possibles. Or, comme nous l'avons expliqué au chapitre précédent, une carte partielle de l'environnement rend une telle estimation à partir des données allothétiques difficile. Pour cette raison, ces systèmes ne sont en général utilisés que pour la localisation sur des cartes construites dans une phase séparée.

2.3 Les modèles markoviens pour la localisation

Les processus de décision markoviens partiellement observables (PDMPO) permettent de modéliser des systèmes à états internes dont l'évolution peut être en partie contrôlée et dont il est impossible de mesurer directement l'état (cf. figure 2.5). Cependant, il est possible d'effectuer des mesures sur ce système qui fourniront une certaine information sur l'état courant. Pour ce faire, une probabilité d'effectuer chaque mesure est associée à chaque état. Lorsque le système est dans un certain état, toute mesure fournira une perception aléatoire selon les probabilités associées à cet état. De plus, l'évolution du système est elle aussi, en général, probabiliste de telle sorte que, à partir d'un état donné, effectuer une action pourra conduire dans différents états, avec une probabilité spécifiée pour chaque transition.

Les PDMPO sont principalement utilisés pour modéliser des processus de décision. A cette fin, une récompense est associée à chaque action effectuée dans chaque état. Le problème est alors de trouver, en fonction des informations disponibles, quelle action effectuer pour maximiser la récompense à long terme. Tout le problème vient, bien sûr, du fait qu'il est impossible de savoir avec certitude dans quel état se trouve le système au moment où l'on effectue une action. Les algorithmes pour résoudre ces problèmes sont extrêmement lourds en temps de calcul et inutilisables pour des systèmes comportant plus de quelque dizaines d'états. Toutefois, la procédure permettant d'estimer l'état courant du système est extrêmement simple et peu coûteuse en temps de calcul. Cette estimation prend la forme d'une distribution de probabilités sur l'ensemble des

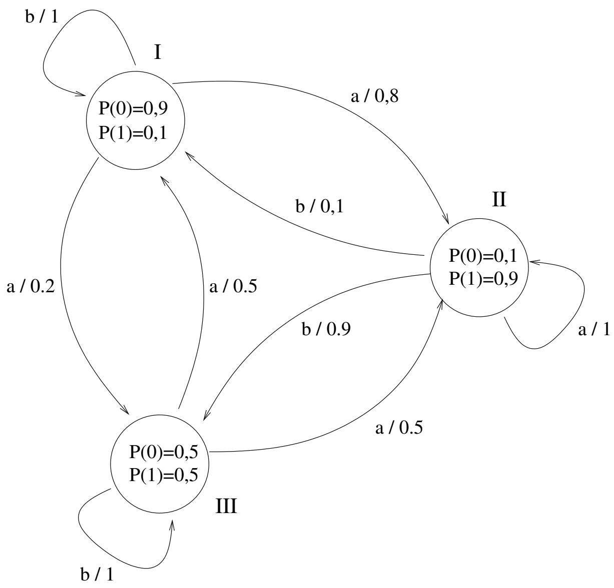


FIG. *2.5: Les PDMPO permettent de modéliser des systèmes à états dont les transitions entre état sont probabilistes et dont les mesures que l'on peut effectuer sur les états sont également probabilistes. Dans cet exemple, le système possède trois états (I, II et III) dans lesquels deux observations (0 et 1) sont possibles avec des probabilités dépendant de l'état. Par exemple, une mesure sur le système dans l'état I aura pour résultat 0 (avec une probabilité 0,9) ou 1 (avec une probabilité 0,1). De plus, deux actions (a et b) peuvent faire passer le système d'un état à un autre avec des probabilités associées. Par exemple, l'action a dans l'état I conduira le système dans l'état II avec une probabilité 0,8 ou dans l'état III avec une probabilité 0,2.*

états possibles du système, distribution reflétant la croyance que l'observateur peut avoir sur l'état courant du système. C'est cette distribution de probabilité, qui intègre les contrôles fournis au système et les mesures faites sur ce système, qui est utilisée pour la localisation en robotique.

L'utilisation des PDMPO pour la localisation est relativement simple. L'environnement est modélisé sous forme d'un PDMPO dont les états correspondent à l'ensemble des positions possibles du robot. Un état peut donc être, soit un nœud d'une carte topologique [Cassandra et al., 1996, Hertzberg and Kirchner, 1996, Shatkay and Kaelbling, 1997, Simmons and Koenig, 1995, Theocharous et al., 2001], soit une cellule d'une grille d'occupation [Burgard et al., 1996, Fox et al., 1998b, Thrun et al., 1999]. Les données idiothétiques correspondent aux actions possibles sur le système et fournissent des probabilités de transition d'un état à un autre correspondant aux déplacements du robot. Les données allothétiques correspondent aux mesures possibles sur le système. A chaque état est donc associée une distribution de probabilité sur l'ensemble des perceptions possibles. Le fait que le système soit partiellement observable correspond au fait que les mesures faites par les capteurs sont entachées de bruit, mais aussi au fait que le robot est soumis au *perceptual aliasing*. Dans cette situation, en effet, les perceptions ne permettent pas de déterminer de manière unique la position du robot mais fournissent simplement une indication sur cette position.

Formellement, un PDMPO se décrit grâce aux éléments suivants :

- Un ensemble d'états $\mathcal{S}$
- Un ensemble d'actions possibles $\mathcal{A}$
- Un ensemble de perceptions possibles $\mathcal{O}$
- Une fonction de transition $T : \{\mathcal{S}, \mathcal{A}, \mathcal{S}\} \to [0, 1]$
- Une fonction d'observation $O : \{\mathcal{S}, \mathcal{O}\} \to [0, 1]$

L'ensemble d'états $\mathcal{S}$ est l'ensemble de toutes les positions possibles du robot au sein de son environnement. Comme nous l'avons mentionné, ces états peuvent correspondre aux nœuds d'une carte topologique ou aux cellules d'une grille d'occupation. L'ensemble des actions possibles $\mathcal{A}$ regroupe les mouvements possibles du robot. Il peut contenir soit des actions discrétisées (avancer de 1 mètre, tourner de 90 degrés [Simmons and Koenig, 1995]), soit des actions dans un domaine continu (avancer de x mètres, tourner de θ degrés [Burgard et al., 1996]). Ces deux ensembles constituent le support de la fonction T qui, à tout triplet (s, a, s') de $\{\mathcal{S}, \mathcal{A}, \mathcal{S}\}$, associe une valeur réelle comprise entre 0 et 1. Cette valeur correspond à la probabilité pour le robot d'atteindre l'état s' après avoir effectué l'action a dans l'état s. Cette fonction modélise à la fois la structure de l'environnement (en déterminant la positions relative des états) et l'odométrie du robot (en prenant en compte l'imprécision de cette information).

Ces trois premiers éléments déterminent la structure spatiale de l'environnement. Les deux derniers, $\mathcal{O}$ et O permettent de modéliser les effets du système perceptif du robot dans l'environnement. L'ensemble $\mathcal{O}$ regroupe toutes les perceptions possibles du robot. Comme pour les déplacements, il peut être soit discrétisé (présence ou absence d'un obstacle devant, à droite ou à gauche du robot [Simmons and Koenig, 1995]), soit continu (présence d'un obstacle à x mètres du robot dans la direction θ par rapport à l'avant du robot [Burgard et al., 1996]). Cet ensemble sert de support à la fonction O qui, à chaque couple (s, o) de $\{\mathcal{S}, \mathcal{O}\}$, associe une valeur réelle comprise entre 0 et 1. Cette valeur correspond à la probabilité pour le robot de réaliser la perception o lorsqu'il se trouve dans l'état s. Comme pour la fonction de transition, cette fonction O modélise à la fois les propriétés de l'environnement et le fonctionnement des capteurs du robot.

Étant donné cette structure, l'estimation de l'état courant du système est représentée par une distribution de probabilité $\mathcal{B}$ sur l'ensemble des états $\mathcal{S}$. Chaque élément $b(s)$ de cette distribution représente la probabilité pour le robot de se trouver à la position représentée par s. Cette distribution permet d'intégrer toutes les informations idiothétiques et allothétiques recueillies par le robot grâce à une opération effectuée en deux temps.

Dans un premier temps, les déplacements du robot sont intégrés. Ainsi, chaque fois que le robot effectue un déplacement correspondant à l'action a, $\mathcal{B}$ est mise à jour par la formule :

$$b_{t+1}(s) = K_{t+1} \times \sum_{s' \in \mathcal{S}} T(s, a, s') \times b_t(s') \tag{2.1}$$

où K est un facteur de normalisation permettant d'assurer que la somme des probabilités $b(s)$ après application de cette équation reste bien égale à 1. Cette équation représente simplement le fait que la probabilité d'être dans l'état s est la somme, pour tous les états s', de la probabilité d'avoir été dans l'état s' avant le déplacement, multipliée par la probabilité de s'être déplacé de l'état s' à l'état s.

Dans un deuxième temps, chaque perception réalisée par le robot est intégrée suivant l'équation :

$$b_{t+1}(s) = L_{t+1} \times 0(s,o) \times b_t(s) \tag{2.2}$$

où L est un facteur de normalisation permettant d'assurer que la somme des probabilités $b(s)$ après application de cette équation reste égale à 1. Cette équation multiplie simplement la probabilité d'être dans un état par la probabilité d'avoir effectué la perception dans cet état. Le résultat est une augmentation de la probabilité des états pour lesquels la perception est très probable, et une diminution pour les autres.

Les systèmes basés sur les PDMPO se révèlent extrêmement robustes en pratique. Ils permettent, en effet, de gérer des environnements soumis à un fort *perceptual aliasing*, ainsi que des perceptions fortement bruitées, notamment dans des environnement très dynamiques. Par exemple, ils ont été utilisés avec succès sur des robots guides dans des musées, parmi des visiteurs nombreux qui produisent un bruit très élevé sur les capteurs de distance des obstacles [Burgard et al., 1996, Thrun et al., 1999]. Ces systèmes sont toutefois limités au niveau de l'apprentissage de la carte car ils ne peuvent fonctionner qu'avec une carte complète de l'environnement. Comme nous le verrons dans les chapitres suivants, l'apprentissage de la carte se déroule donc en général de manière supervisée ou dans une phase préalable à l'utilisation.

2.4 Perception active

Nous avons mentionné dans le chapitre précédent les différences entre capteurs omnidirectionnels et directionnels relatives à la comparaison des données allothétiques recueillies en différentes positions. Ces différences ont des conséquences sur l'utilisation des capteurs : les capteurs omnidirectionnels peuvent être utilisés directement, tandis qu'il est nécessaire de choisir une direction pour les capteurs directionnels. Ce choix de direction peut être figé, ou donner lieu à des procédures de *perception active* qui sélectionnent la direction des capteurs en fonction de l'état courant du processus de localisation, dans le but de faciliter ce processus.

D'une manière générale, la perception active met l'accent sur le fait que les actions d'un observateur peuvent lui permettre d'obtenir plus facilement de l'information sur une scène que s'il reste passif face à cette scène [Aloimonos, 1993]. Par exemple, il est beaucoup plus simple de reconstruire un modèle tridimensionnel d'une scène à partir d'images prises par une caméra si l'on peut la déplacer ou modifier certains de ses paramètres, tel que le point de focalisation.

Dans le cadre de cette thèse, le terme perception active est utilisé dans une acception assez restreinte qui ne concerne que la localisation dans le cadre du suivi de plusieurs hypothèses. Nous nous limitons donc au contrôle des capteurs directionnels dans le but d'améliorer l'estimation des crédibilités relatives des différentes hypothèses. Dans le cas où plusieurs hypothèses de localisation existent, en effet, il est intéressant d'orienter le capteur dans une direction pour laquelle les perceptions ne seront pas identiques pour les différentes hypothèses, puisque des perceptions identiques ne permettront pas de les distinguer.

De telles procédures sont particulièrement pertinentes à implémenter dans le cadre des systèmes de localisation basés sur des PDMPO. En effet, il existe un critère simple permettant de mesurer la qualité d'une estimation de la position. Cette mesure est l'entropie de la distribution de probabilité $\mathcal{B}$ représentant la position du robot :

$$H(\mathcal{B}) = -\sum_{s} b(s)\log(b(s)) \tag{2.3}$$

L'entropie d'une distribution de probabilité est minimale et vaut 0 lorsqu'une seule position de $\mathcal{B}$ a une probabilité de 1, tandis que les autres ont une probabilité nulle. Cette configuration correspond à une certitude sur la position du robot. Au contraire, l'entropie est maximale lorsque l'incertitude est maximale, c'est-à-dire lorsque toutes les positions de $\mathcal{B}$ ont une probabilité égale.

Ce critère est donc utilisable pour sélectionner une direction dans laquelle une perception permettra en moyenne de réduire l'incertitude de la position [Fox et al., 1998a]. Une telle procédure prend en compte la distribution courante des probabilités de position et recherche quelle direction de perception conduira en moyenne à une diminution de l'entropie. Cela revient à rechercher la direction α qui minimise la fonction :

$$E(\alpha) = \sum_{o_\alpha} p_{o_\alpha} H(\mathcal{B}_{o_\alpha}) \tag{2.4}$$

où les o_α sont les perceptions possibles au sein de l'environnement dans la direction α, p_{o_α} la probabilité de cette perception sur tout l'environnement et B_{o_α} la distribution de probabilité résultant de la perception o_α.

L'utilisation du critère 2.4 permet un gain important dans la précision de la localisation [Fox et al., 1998a]. Nous verrons que ce critère peut être transposé dans notre modèle, où nous le comparerons à d'autres méthodes de sélection de la direction.

2.5 Conclusion

La recherche de la plus grande autonomie possible pour un robot utilisant notre système de navigation nous a conduit à utiliser une méthode de localisation globale qui intègre de manière efficace les informations proprioceptives et perceptives. Cette méthode, qui ne nécessite pas d'estimation initiale de la position, permet d'estimer la position de l'animat dans tous les cas et dans tous les types d'environnements. L'animat peut ainsi être déplacé ou déposé en un lieu quelconque de son environnement sans qu'il soit nécessaire de fournir une estimation de cette position au système de navigation, ni d'utiliser un système de localisation auxiliaire pour estimer cette position.

La méthode de localisation globale retenue est similaire à celle utilisée dans les modèles basés sur des PDMPO. Ainsi la position de l'animat est représentée par une distribution de probabilité sur l'ensemble des nœuds de la carte topologique de notre modèle. Cette distribution permet d'intégrer toutes les informations disponibles pour l'animat en s'inspirant des équations de mise à jour de l'estimation de l'état des PDMPO.

Notre modèle permet également une construction simultanée de la carte, ce qui est difficile en utilisant une méthode de localisation globale. Nous verrons en détail dans le chapitre 5 que cette capacité repose sur une gestion de l'incomplétude de la carte lors de l'estimation des probabilités relatives des différentes hypothèses (cf. paragraphe 1.3.2). Cette capacité repose également sur une estimation du fait que la distribution de probabilité représente ou non de manière correcte la position du robot. Dans le cas où cette représentation est correcte, la localisation globale est utilisée tandis que, dans le cas contraire, un suivi de position est utilisé temporairement.

Notre modèle permet également la gestion des capteurs directionnels, d'une part, en s'accommodant de l'incomplétude de l'information lors de l'intégration à l'estimation de la position et, d'autre part, en utilisant des procédures de perception active pour sélectionner la direction des capteurs la plus pertinente.

Il est enfin possible de tracer un parallèle entre la probabilité associée à chaque nœud de la carte de notre modèle et le taux de décharge des cellules de lieux dans l'hippocampe des rats. Dans ces deux cas, en effet, cette valeur sera la plus élevée pour le nœud ou la cellule correspondant à la position courante de l'animat ou du rat. La question de savoir si le taux de décharge des cellules de lieu des rats pourrait représenter une probabilité de présence en une position donnée reste cependant ouverte. Nous reviendrons plus en détail sur ce point dans le chapitre 7.

Chapitre 3

Cartographie

Après avoir présenté le problème de l'estimation de la position d'un animat au sein d'une carte complète de son environnement, nous allons maintenant voir comment il est possible de construire cette carte. Nous verrons que la construction de la carte impose des limitations sur les méthodes de localisation utilisables. Nous montrerons également qu'il existe deux types de méthodes de cartographie, qui permettent d'utiliser de manière plus ou moins complète les données disponibles pour l'animat. Les modèles mentionnés dans ce chapitre sont également présentés plus en détail dans une publication [Filliat and Meyer, 2001b].

3.1 Les problèmes de la cartographie

3.1.1 Limitation des méthodes de localisation

La tâche de cartographie est intrinsèquement plus complexe que celle de localisation. En effet, la localisation revient à rechercher, parmi les positions possibles représentées dans la carte, celle qui correspond le mieux à la position courante de l'animat. Cette recherche se déroule donc dans un espace fermé, car on postule que la position recherchée se trouve parmi les positions enregistrées dans la carte. Dans le cas de la cartographie, une difficulté importante provient du fait que l'estimation de la position du robot et la construction de la carte se déroulent dans un espace de recherche ouvert. A chaque instant, il faut déterminer si le robot se trouve dans une position déjà mémorisée dans la carte et, dans ce cas, déterminer laquelle, ou s'il se trouve dans un lieu qu'il n'a jamais visité. Dans ce second cas, il faut être capable de positionner ce lieu nouveau par rapport à la carte, et modifier celle-ci pour y incorporer les informations relatives à ce nouveau lieu.

De plus, l'utilisation de capteurs directionnels lors de la construction de la carte conduit à une incomplétude dans la représentation des lieux visités. Comme nous l'avons déjà mentionné, cette incomplétude de la représentation pose des problèmes pour la comparaison des données allothétiques de toutes les positions possibles au sein de la carte. Notamment, il est difficile d'utiliser des méthodes de localisation globale qui suivent plusieurs hypothèses car l'incomplétude de la carte rend l'estimation des crédibilités relatives des différentes hypothèses complexe, voire impossible.

En conséquence, un certain nombre de méthodes de localisation précédemment évoquées ne sont plus utilisables lorsque la carte est incomplète. La plupart des systèmes reposent donc sur une méthode de localisation qui réalise un suivi de position. En effet, si le robot atteint un lieu qui n'est pas représenté dans la carte, il est possible, grâce à ces méthodes locales de localisation, de définir sa position par rapport à une position précédente connue au sein de la carte.

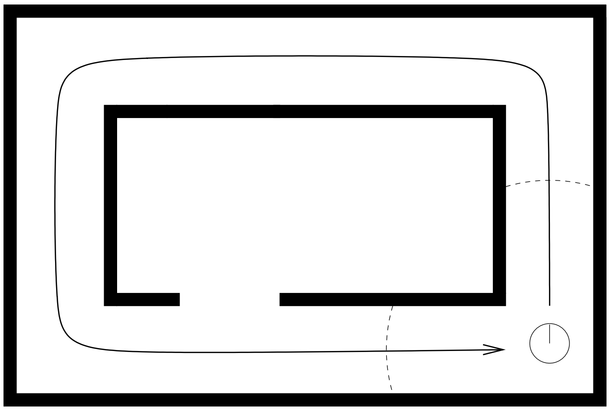


FIG. *3.1: Un exemple simple d'environnement contenant un cycle.*

L'utilisation de systèmes de localisation effectuant un suivi de position pose cependant certains problèmes car, comme nous l'avons vu au chapitre précédent, ces méthodes peuvent diverger et conduire à une estimation erronée de la position sans possibilité de retrouver la position réelle du robot. Ce problème est particulièrement important au cours du processus de cartographie car les erreurs de localisation conduisent à des mises à jour erronées de la carte, ce qui peut conduire à des erreurs durables dans les futures tentatives de localisation. Ces erreurs sont cruciales dans les environnements *cycliques*, c'est-à-dire dans des environnements contenant des boucles dont les différentes parties ne sont pas toutes visibles les unes à partir des autres (cf. figure 3.1). En effet, dans de tels environnements, les erreurs de cartographie durant le parcours du cycle peuvent empêcher la reconnaissance de la fermeture du cycle et conduire à des cartes à la topologie erronée (cf. figure 3.2).

3.1.2 Cohérence de la carte

Dans le cas des cartes topologiques, un autre problème est de plus susceptible de se produire, même si le système de localisation est suffisamment performant et permet de reconnaître une position déjà visitée après la fermeture d'un cycle. Dans ce cas, le système de cartographie risque de créer une carte localement fausse qui ne sera plus cohérente. Une carte incohérente est une carte dans laquelle la position relative de deux lieux accessibles au robot aura des valeurs différentes en fonction du chemin utilisé pour son calcul (cf. figure 3.3). Cette incohérence n'est pas toujours problématique. Par exemple, elle ne pose aucun problème si les positions relatives des lieux ne

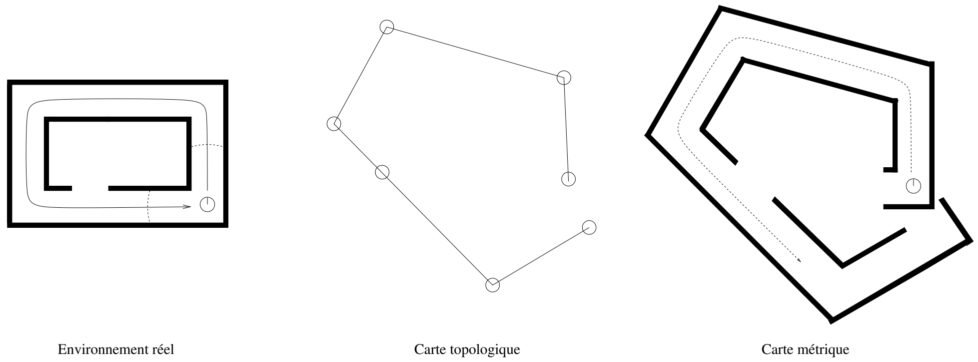


FIG. *3.2: Lorsque la position du robot est estimée par une méthode de suivi de position, les erreurs s'accumulent pendant le parcours d'un cycle. Lors de la fermeture de ce cycle, le système peut alors être incapable de reconnaître la position initiale du robot, ce qui conduit a une carte incorrecte, dans le cas topologique (partie gauche) et dans le cas métrique (partie droite).*

sont utilisées que localement, entre deux nœuds reliés par un lien, pour résoudre le problème de *perceptual aliasing* au cours de la localisation. Dans ce cas en effet, les informations d'une seule connexion sont utilisées. Le fait que la somme des informations de plusieurs connexions donne des informations différentes selon le chemin choisi ne pose alors aucun problème.

Toutefois, si l'on cherche à estimer la position relative de deux nœuds quelconques de la carte, il est primordial que la cohérence de la carte soit assurée car, dans une carte incohérente, cette position relative est définie de manière ambiguë.

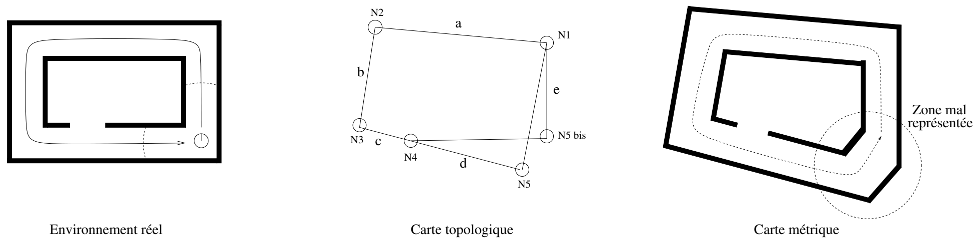


FIG. *3.3: Lors de la fermeture d'un cycle si, malgré les erreurs de localisation, la position est correctement estimée, la carte résultante risque néanmoins d'être de mauvaise qualité. Dans le cas des cartes topologiques, la carte risque d'être incohérente. Dans une carte incohérente, la position relative de deux nœuds sera différente si deux chemins différents sont utilisés pour son calcul. Dans cet exemple, la position du nœud 5 estimée par rapport au nœud 1 via les connexions a, b, c et d (position $N5$) sera différente de la position estimée en utilisant la connexion e directe entre les nœuds 1 et 5 (position $N5bis$). Dans le cas des cartes métriques, la carte risque de contenir une zone ne correspondant pas du tout à l'environnement réel.*

Dans le cas où cette cohérence globale est utile, ce qui est le cas dans notre thèse, elle peut être assurée de trois manières différentes.

- La première manière consiste à associer une position métrique dans un cadre de référence global à chacun des nœuds de la carte, plutôt que de mémoriser les informations de positions relatives dans les arêtes reliant les noeuds (cf. paragraphe 1.4.1). La position relative des nœuds est ainsi définie de manière non ambiguë et rend la carte naturellement cohérente. Cette méthode, qui rapproche fortement la carte topologique d'une carte métrique, présente l'inconvénient de supprimer certains avantages inhérents à la carte topologique. En premier lieu, l'utilisation d'un cadre de référence global oblige à assurer en permanence une estimation fiable de la position par l'odométrie. La force des cartes topologiques, qui peuvent ne dépendre de l'odométrie que localement entre deux nœuds, est alors perdue. De plus, la modification d'un seul lien dans une carte topologique permet de modifier potentiellement la position relative de tout les nœuds, afin de mieux rendre compte de l'ensemble de l'environnement si la carte ne contient qu'une erreur localisée. Cela est impossible si une position est associée à chaque nœud (cf. figure 3.4).

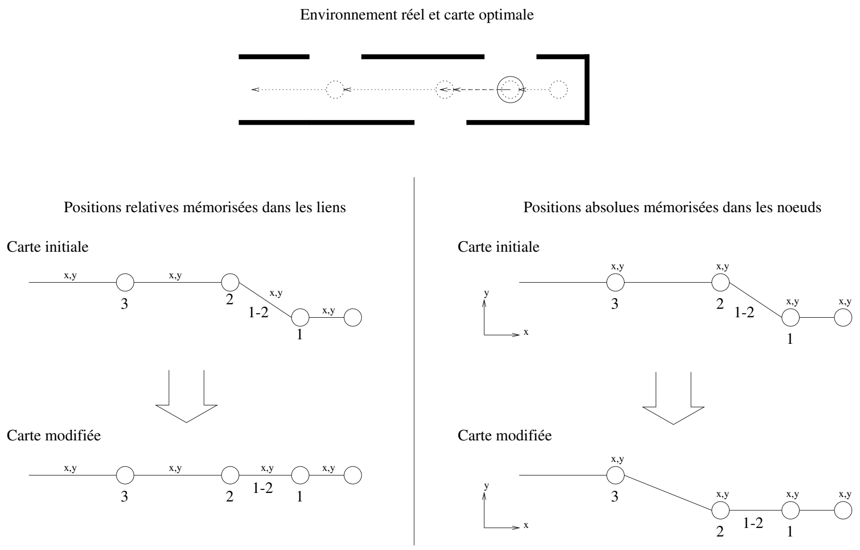


Fig. *3.4: Lorsque la cohérence d'une carte topologique est assurée en associant une position à chacun des nœuds de la carte, des propriétés intéressantes des carte topologiques sont perdues. Dans cet exemple, la carte ne contient qu'une erreur locale (le lien* $1-2$*). Si cette erreur est détectée et corrigée lors d'un nouveau passage du nœud* 1 *au nœud* 2*, la carte deviendra correcte sur l'ensemble de l'environnement dans le cas où les relations métriques sont mémorisées localement (Partie gauche de la figure). Si au contraire, les informations métriques sont mémorisées de manière absolue pour chaque nœud, une telle modification déplacera l'erreur, sans la supprimer (Partie droite de la figure).*

– La seconde méthode de maintien de la cohérence permet de conserver les avantages des cartes topologiques en modifiant les données métriques mémorisées dans les connexions proches de la position courante. La cohérence est donc assurée par des adaptations locales des données des connexions, ce qui conduit des cartes où l'essentiel des erreurs est concentré sur une petite zone (cf. figure 3.5). Cette zone correspond alors très mal à l'environnement réel. Cette méthode, tout en assurant la cohérence, conduit à des erreurs locales de cartographie très similaires à celles décrites lors de la fermeture des cycles dans les cartes métriques (cf. figure 3.3).

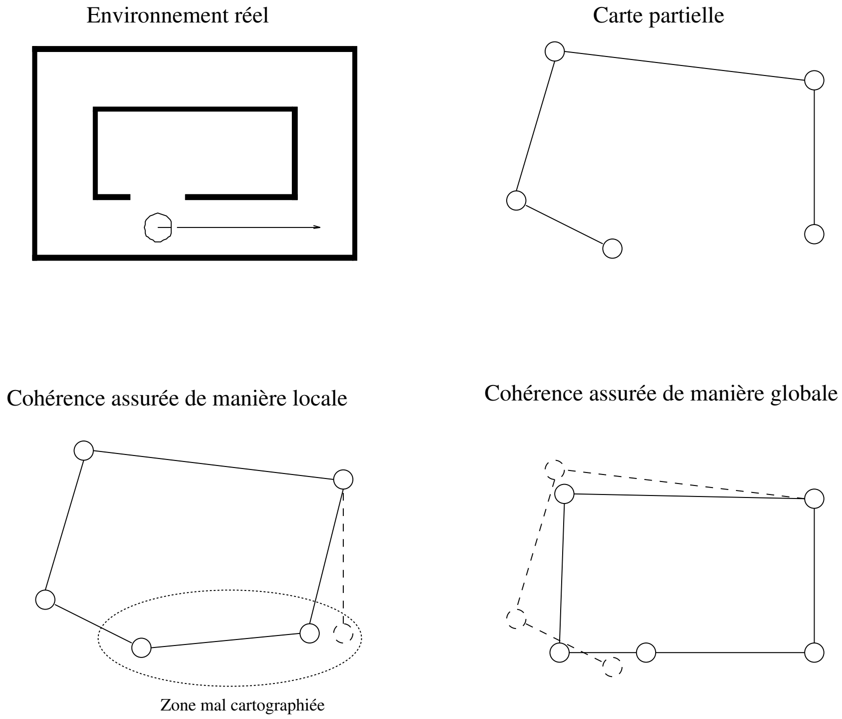


FIG. *3.5: Lorsque la cohérence est assurée uniquement par des adaptations locales, la carte résultante contient des erreurs qui sont concentrées dans une zone limitée. Toutefois, dans cette zone, l'environnement réel peut être extrêmement différent de ce qui est représenté dans la carte (Partie en bas-gauche de la figure). Si la cohérence est assurée de manière globale par des modifications de l'ensemble de la carte, les erreurs seront réparties sur l'ensemble de la carte qui correspondra globalement mieux à l'environnement réel (Partie en bas-droite de la figure).*

– Enfin, la troisième méthode permet de conserver les avantages des cartes topologiques et d'assurer globalement la cohérence en modifiant l'ensemble de la carte lorsqu'un problème de cohérence est détecté (cf. figure 3.5). En général, cette méthode permet de construire une carte globalement plus précise de l'environnement. Nous verrons par la suite comment une telle correction globale peut être réalisée.

3.1.3 Cartographie incrémentale et retour en arrière

Les méthodes de maintien de la cohérence d'une carte que nous venons de citer correspondent à deux catégories de systèmes de cartographie. La première catégorie contient des systèmes qui effectuent une cartographie incrémentale, c'est-à-dire qui modifient la carte uniquement autour de la position courante du robot. Les deux premières méthodes de maintien de la cohérence correspondent à cette catégorie. La troisième catégorie, qui correspond au maintien global de la cohérence, regroupe des systèmes capables de revenir sur une modification passée de la carte au vu d'informations a posteriori sur les positions précédentes du robot.

La *cartographie incrémentale* [Thrun et al., 2000], permet simplement d'ajouter de nouvelles informations dans la carte à partir de l'estimation courante de la position du robot. Cependant, si cette estimation se révèle fausse a posteriori, il est impossible de revenir sur les modifications qui ont été effectuées. Cette limitation se révèle cruciale dans les environnements contenant des cycles parce que la fermeture d'un tel cycle donne une information importante sur les erreurs des estimations précédentes de la position du robot. Cette information est ignorée par les systèmes de cartographie incrémentale et conduit à des cartes dans lesquelles les erreurs vont se concentrer dans une petite zone (cf. figure 3.5). Cette méthode correspond aux deux premières méthodes de maintien de la cohérence des cartes dans lesquelles les incohérences, qui indiquent que la carte construite jusqu'ici est erronée, sont traitées uniquement par une modification locale de la carte et non par une remise en cause globale de la construction de la carte.

La cartographie incrémentale constitue donc la méthode de base des systèmes utilisant des cartes topologiques dans lesquels la cohérence est assurée de manière locale, ou simplement négligée. De plus, cette méthode regroupe de nombreux systèmes basés sur des cartes métriques. Cela semble dû à la fusion de toutes les informations dans un cadre de référence unique, qui complique la reprise des modifications passées. En effet, une erreur dans l'estimation de la position du robot se propage à tous les éléments de la carte perçus par le robot à ce moment-là. Or, comme l'historique des modifications des paramètres des objets de la carte n'est en général pas mémorisé, il est difficile, même si l'on recueille de nouvelles informations sur les positions passées du robot, de revenir sur ces modifications pour prendre en compte ces nouvelles informations.

La seconde catégorie de systèmes de cartographie regroupe donc les systèmes qui permettent d'intégrer des informations a posteriori sur les positions passées du robot. Pour les cartes topologiques, ces systèmes assurent la cohérence de manière globale. Il faut noter que l'intégration de données a posteriori sur les positions passées est relativement simple dans les cartes topologiques grâce à la séparation des données idiothétiques et allothétiques que cette représentation implique. En effet, des erreurs dans l'estimation de la position du robot n'influent que sur les informations mémorisées dans les liens de la carte, et non sur les définitions allothétiques des lieux qui sont mémorisées dans les nœuds. Ainsi, revenir sur une modification passée de la carte, lorsque des nouvelles informations sur la position du robot sont disponibles, requiert simplement de modifier les informations mémorisées dans les liens et ne concerne pas les données allothétiques mémorisées, comme c'est le cas dans les cartes métriques. Toutefois, certains types de cartes métriques, qui s'attachent à mémoriser les informations idiothétiques et les informations allothétiques de manière différente, peuvent aussi donner lieu à de telles adaptations. Nous verrons notamment que des systèmes de cartographie basés sur la construction de PDMPO, qui prennent en compte toutes les informations passées et futures sur la position du robot afin de construire la carte de l'environnement, atteignent ce but.

3.2 Cartographie incrémentale

Les modèles de cartographie incrémentale utilisent donc une méthode de suivi de position pour estimer la position du robot par rapport à la carte existante. Au vu de cette position, si la carte ne représente pas le lieu ou les objets perçus par le robot, ceux-ci sont ajoutés à la carte. En revanche, si ce lieu est déjà représenté, la carte est adaptée en fonction des nouvelles perceptions.

3.2.1 Estimation de la position

Suivi de position

Pour estimer la position dans les cartes topologiques sans *perceptual aliasing* [Sharp, 1991, Burgess et al., 1994, Levitt and Lawton, 1990, Franz et al., 1998, Yamauchi and Langley, 1997, Gaussier et al., 1998, Bachelder and Waxman, 1995], il suffit de comparer les données allothétiques courantes avec les données mémorisées dans chacun des nœuds de la carte. Si aucun des nœuds ne correspond suffisamment bien aux données courantes, cela signifie que le lieu n'est pas représenté dans la carte et qu'il devra donc être ajouté. Au contraire, si la similitude des données courantes avec un nœud de la carte est suffisante, ce nœud sera reconnu comme la position courante du robot. Le choix entre ces deux alternatives est le point difficile du processus de cartographie. En effet, dans le cas de la simple localisation, il suffit de trouver le nœud correspondant le mieux à la situation courante pour trouver la position du robot. Dans le cadre de la cartographie, il faut de plus utiliser un seuil pour décider si le lieu le plus similaire est la position courante du robot ou non : si la similitude est supérieure à ce seuil, le nœud est reconnu, sinon un nouveau lieu est créé. Cette utilisation d'un seuil rend le processus de cartographie potentiellement plus instable que la localisation seule.

Lorsque le *perceptual aliasing* est pris en compte, percevoir un lieu différent de tous les lieux mémorisés dans la carte permet toujours de conclure que ce lieu est nouveau. Mais percevoir des données allothétiques qui correspondent à un lieu déjà mémorisé n'est pas suffisant pour déterminer si le lieu est nouveau ou s'il est connu. En effet, un lieu nouveau peut être similaire, du point de vue allothétique, à un lieu déjà visité. La position précédente du robot doit donc être prise en compte pour déterminer si un lieu est nouveau ou s'il correspond à un nœud mémorisé. Si la position prédite par l'odométrie depuis le lieu précédent ne correspond pas au lieu déjà mémorisé, le lieu est considéré comme nouveau et ajouté à la carte [Kuipers and Byun, 1991, Engelson and McDermott, 1992, Kurz, 1995, Donnart and Meyer, 1996, Yamauchi and Beer, 1996, Kunz et al., 1997, Von Wichert, 1998, Nehmzow and Owen, 2000, Balakrishnan et al., 1999]. Certains modèles intègrent directement les informations allothétiques et la position pour la reconnaissance des nœuds, se ramenant ainsi au cas où il n'y a pas de *perceptual aliasing* [Arleo and Gerstner, 2000, Touretzky et al., 1994, Dedeoglu et al., 1999, Mataric, 1992].

Dans le cas des cartes métriques représentant des objets, la position du robot est estimée par une approche de suivi de position. Cela est possible car les données allothétiques se réfèrent à des objets distants du robot. Ainsi, le robot est en général capable d'estimer sa position, même pour des lieux qu'il n'a jamais visités, par rapport à des objets qu'il a perçus depuis d'autres positions. La position du robot est donc estimée en utilisant les objets visibles qui sont déjà dans la carte, ainsi que les données idiothétiques [Smith et al., 1988, Ayache and Faugeras, 1989, Moutarlier and Chatila, 1990, Leonard et al., 1992, Hébert et al., 1996, Feder et al., 1999, Castellanos et al., 1999, Gasós and Martín, 1997]. Pour les grilles d'occupation, la position du robot est estimée par les techniques de mise en correspondance mentionnées au chapitre précédent, en utilisant les données idiothétiques afin de restreindre l'espace de recherche [Moravec and Elfes, 1985, Thrun, 1999, Koren and Borenstein, 1991].

Correction de l'odométrie

Les techniques de suivi de position mentionnées précédemment permettent de limiter la dérive de l'odométrie en utilisant les données allothétiques en conjonction avec la carte partielle de l'environnement qui a déjà été construite. Toutefois, cette carte a été construite à partir de ces mêmes données idiothétiques et ce processus circulaire (voir le paragraphe 1.1) peut rendre la cartographie instable.

Pour résoudre ce problème, de nombreux systèmes utilisent des méthodes supplémentaires de correction de l'odométrie, indépendantes du processus de localisation. Deux grandes catégories de méthodes sont utilisées.

La première catégorie regroupe les méthodes utilisant des capteurs supplémentaires, tels que les compas magnétiques ou les gyroscopes [Hafner, 2000, Nehmzow and Owen, 2000]. Ces capteurs sont utilisés principalement pour corriger l'estimation de la direction qui, comme nous l'avons déjà mentionné, est l'information la plus difficile à estimer à partir des données idiothétiques. L'intérêt de ces capteurs est qu'ils fournissent une mesure directe de la direction, indépendante d'un processus d'intégration. La qualité de cette mesure ne se dégrade donc pas au cours du temps et permet d'obtenir une meilleure estimation de la position du robot.

La seconde catégorie regroupe les méthodes qui se basent sur des hypothèses sur l'environnement, afin de permettre le recalage de l'odométrie. L'hypothèse la plus couramment utilisée suppose que les murs de l'environnement sont rectilignes et orthogonaux [Thrun, 1999, Kunz et al., 1997], ce qui permet de corriger facilement l'estimation de la direction du robot.

3.2.2 Modifications de la carte

Une fois la position du robot identifiée par rapport à la carte, cette dernière est mise à jour en fonction des données allothétiques. Dans le cadre de la cartographie incrémentale, cette mise à jour ne concerne que la position courante, sans retour sur les modifications effectuées précédemment.

Dans le cas des cartes topologiques, les données allothétiques sont utilisées pour modifier les données mémorisées dans le nœud reconnu. Cela permet d'avoir une meilleure estimation des données allothétiques caractérisant le lieu grâce au filtrage du bruit sur ces données. Si la position du robot correspond à un lieu nouveau, un nouveau nœud est simplement ajouté à la carte, pour mémoriser les données allothétiques courantes. Les données idiothétiques recueillies depuis le nœud précédent sont utilisées pour créer ou modifier l'arête qui joint le nœud précédent au nœud courant. Comme nous l'avons mentionné précédemment, dans le cas où des informations métriques sont mémorisées entre les lieux, la carte obtenue peut alors se révéler incohérente. Dans les méthodes de cartographie incrémentale, la cohérence peut être assurée par l'association d'une position à chacun des nœuds de la carte [Arleo and Gerstner, 2000, Touretzky et al., 1994, Dedeoglu et al., 1999, Mataric, 1992, Kurz, 1995, Donnart and Meyer, 1996], ou par une adaptation locale des valeurs des liens. Dans le cas où ces valeurs sont simplement utilisées de manière locale, sans chercher à estimer les relations métriques entre lieux distants, le maintien de la cohérence peut être simplement négligé [Kuipers and Byun, 1991, Engelson and McDermott, 1992].

Dans le cas des cartes métriques contenant des objets, les paramètres des objets reconnus sont mis à jour en fonction de la position du robot et des données allothétiques. Cette mise à jour s'effectue en général par l'utilisation d'un filtre de Kalman qui fournit un compromis optimal entre l'information précédente et l'information nouvelle, en prenant en compte leurs incertitudes respectives. Si des objets détectés ne correspondent à aucun objet de la carte, ils sont ajoutés à leur position mesurée [Smith et al., 1988, Ayache and Faugeras, 1989, Leonard et al., 1992, Moutarlier and Chatila, 1990, Castellanos et al., 1999]. Dans le cas des grilles d'occupation, les probabilités d'occupation des cellules autour du robot sont mises à jour en fonction de cette position et des données allothétiques. Cette mise à jour peut se faire de différentes manières, en utilisant la théorie des probabilités [Moravec and Elfes, 1985, Thrun, 1999], la théorie des possibilités [Hughes and Murphy, 1992], ou une méthode heuristique [Koren and Borenstein, 1991].

3.2.3 Stratégies d'exploration

Pour limiter les erreurs de cartographie, il est possible d'utiliser une exploration active de l'environnement, plutôt que de passivement mémoriser les données recueillies par le robot. Dans le cadre des cartes topologiques, il est par exemple possible, lorsqu'un nœud a été reconnu, de chercher à atteindre un des nœuds voisins mémorisé dans la carte [Kuipers and Byun, 1991]. Si ce second nœud est correctement détecté, il permet de confirmer la détection du nœud précédent qui est alors mis à jour. D'autre modèles utilisent une exploration active de l'environnement pour diriger le robot vers les zones pour lesquelles l'incertitude de la carte est grande [Engelson, 1995]. Enfin certains modèles sont capables de générer des hypothèses sur des portions non visitées de l'environnement qui sont ensuite vérifiées grâce à une procédure qui dirigera le robot dans les zones où de telles hypothèses ont été faites [Kunz et al., 1997]. Des stratégies d'explorations peuvent également être utilisées afin de garantir un exploration rapide et exhaustive de l'environnement [Thrun, 1999, Kunz et al., 1997, Yamauchi et al., 1999].

D'une manière générale, de telles procédures permettent donc, d'une part, de limiter les erreurs de cartographie en insistant sur les zones incertaines et en évitant que l'estimation de la position ne devienne trop mauvaise et, d'autre part, de garantir une exploration exhaustive de l'environnement. Nous verrons que notre modèle utilise deux procédures simples pour réaliser ces deux tâches.

3.3 Retour sur les modifications passées

Les modifications apportées à la carte au cours du processus de cartographie se font donc en supposant que l'estimation de la position du robot au moment de cette modification est correcte, mais il arrive que cette estimation se révèle fausse a posteriori. Dans ce cas, les modifications de la carte ont été effectuées de manière incorrecte, et il serait souhaitable de pouvoir revenir sur ces modifications pour prendre en compte les nouveaux indices sur les positions passées du robot. Nous avons vu que les systèmes réalisant une cartographie incrémentale n'effectuent pas de telles corrections. Un certain nombre d'autre modèles, que nous présentons dans ce paragraphe, en sont capables.

3.3.1 Découplage des données idiothétiques et allothétiques

Pour pouvoir remettre en cause une modification passée, il faut pouvoir retrouver tous les éléments de la carte qui ont été influencés par cette modification. Nous avons vu précédemment que cela est relativement simple pour les cartes topologiques, grâce à la séparation de la mémorisation des données idiothétiques et allothétiques. Dans le cas de cartes métriques, le cadre de référence commun qui permet de fusionner les deux types de données peut se révéler être un handicap important. Pour surmonter ce handicap, les modèles de cartographie métrique découplent eux aussi la mémorisation des données issues de l'odométrie et des perceptions.

Cela peut être fait en ne fusionnant pas les données perçues depuis différentes positions par le robot [Einsele, 1997, Einsele, 1997, Lu and Milios, 1997, Thrun et al., 2000]. La carte est alors une collection de groupes de données allothétiques, chacun de ces groupes étant mémorisé avec la position du point depuis lequel ces données ont été perçues. Ces positions sont simplement mesurées par les données idiothétiques (cf. figure 3.6). La carte ainsi représentée est similaire à une carte topologique, à l'exception du fait que les données allothétiques mémorisées sous forme métrique permettent d'estimer la position relative des différents groupes de données, ce qui est impossible dans une carte topologique. Dans de telles cartes, il est alors possible de modifier simplement les positions de ces différents groupes si de nouvelles données apportent des informations supplémentaires sur les positions du robot au moment où ces groupes ont été mémorisés.

Il est également possible d'enregistrer la carte sous forme d'une collection des cartes locales dont les positions relatives sont également mesurées par l'odométrie [Hébert et al., 1996, Borghi and Brugali, 1995, Burgard et al., 1999, Gutmann and Konolige, 2000]. Chacune des sous-cartes contient des objets dont les positions relatives ont été mesurées par les données allothétiques seulement. Par rapport à l'approche précédente, les sous-cartes regroupent toutes les données allothétiques qui ont au moins un point commun, même si celles-ci ont été acquises

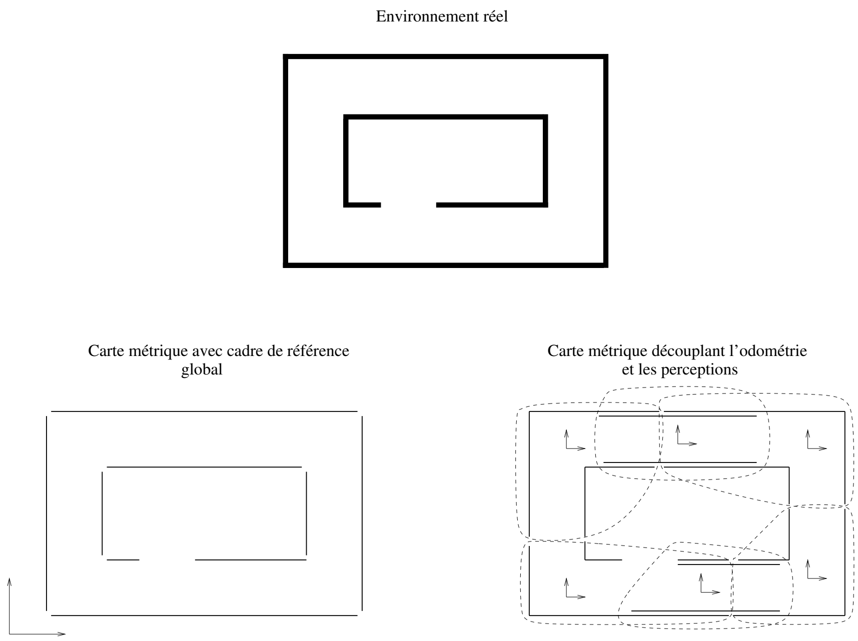


FIG. 3.6: *Afin de permettre l'intégration de données a posteriori sur les positions passées du robot, les modèles utilisant des cartes métriques doivent découpler la mémorisation de l'odométrie et des perceptions. Pour cela, au lieu de fusionner toutes ces informations dans un cadre de référence global (Partie gauche de la figure), les données allothétiques perçues depuis une position unique sont mémorisées ensemble, conjointement avec la position depuis laquelle elles ont été mesurées (Partie droite de la figure).*

depuis des positions différentes (cf. figure 3.7).

3.3.2 Cartographie «hors-ligne»

En utilisant une carte topologique ou une carte métrique sous une des formes décrites précédemment, l'algorithme de cartographie va donc chercher à estimer au mieux la position relative des différentes parties de la carte, que ce soient les nœuds de la carte topologique, ou les portions de la carte métrique. Cette estimation pourra prendre en compte toute nouvelle information sur les positions passées du robot, et modifier la carte en conséquence. Notons que le résultat d'une modification d'un choix passé en fonction de nouvelles données est le même que celui qui serait obtenu si ce choix passé avait été effectué en connaissant le futur. Ces algorithmes de cartographie peuvent donc être considérés, d'un certain point de vue, comme tenant compte du futur. Ils seront donc particulièrement efficaces s'ils connaissent le futur du robot. Pour parvenir à cela, certains d'entre eux fonctionnent «hors-ligne» sur un ensemble de données recueillies par le robot. Cet ensemble de données, qui contient les informations concernant un intervalle de temps étendu, contient en effet une partie du futur pour les instants inclus dans cet intervalle. Chaque position du robot pourra donc être estimée en prenant en compte, non seulement les informations qui la précèdent dans l'ensemble de données, mais aussi les informations qui lui font suite.

La plupart des systèmes de cette catégorie utilisent une carte sous forme de PDMPO, ainsi

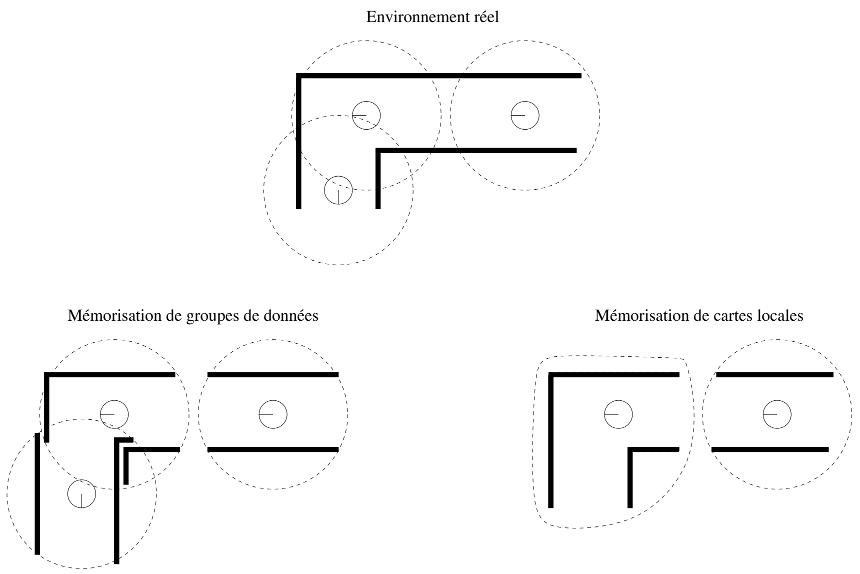


FIG. *3.7: Deux stratégies sont possibles pour séparer l'odométrie et les perceptions dans les cartes métriques. La première enregistre séparément toutes les données qui ont été perçues en des positions différentes (Partie gauche de la figure), la seconde fusionne les données allothétiques ayant des point communs sous forme de cartes locales (Partie droite de la figure).*

qu'un algorithme qui permet de reconstituer un PDMPO à partir d'un ensemble d'actions et de perceptions provenant du système modélisé. Cet algorithme est un algorithme de *maximisation de l'espérance* [McLachlan and Krishnan, 1997], c'est-à-dire qu'il recherche le PDMPO qui permet de générer les données qui ont été recueillies par le robot avec la plus grande certitude possible.

Cet algorithme est itératif et fonctionne en deux phases. La première phase utilise une carte partielle de l'environnement et estime la position du robot au vu des données de l'ensemble d'apprentissage. Cette estimation est similaire à l'estimation de position décrite pour les systèmes utilisant les PDMPO. Elle est toutefois plus puissante car elle permet de prendre en compte les données de l'ensemble d'apprentissage acquises après la position estimée (cf. figure 3.8). En utilisant ces positions estimées, la seconde phase de l'algorithme produit, grâce aux données allothétiques de l'ensemble d'apprentissage, une nouvelle carte qui rend mieux compte des données acquises. L'utilisation itérée de ces deux phases permet de converger vers une carte qui rend compte au mieux des données recueillies. Cet algorithme est utilisable pour des cartes topologiques [Simmons and Koenig, 1996, Theocharous et al., 2001, Shatkay and Kaelbling, 1997] et pour des cartes métriques [Thrun et al., 1998, Burgard et al., 1999]. Il est toutefois extrêmement sensible aux conditions initiales et n'est garanti de converger que vers un minimum local. Il est, de plus, relativement lourd en temps de calcul et demande une quantité importante de données. Diverses optimisations sont donc nécessaires pour qu'il soit utilisable en pratique [Simmons and Koenig, 1996, Burgard et al., 1999]. Son principal inconvénient toutefois est qu'il ne peut être utilisé «en-ligne» pour estimer incrémentalement une carte à chaque nouvelle donnée

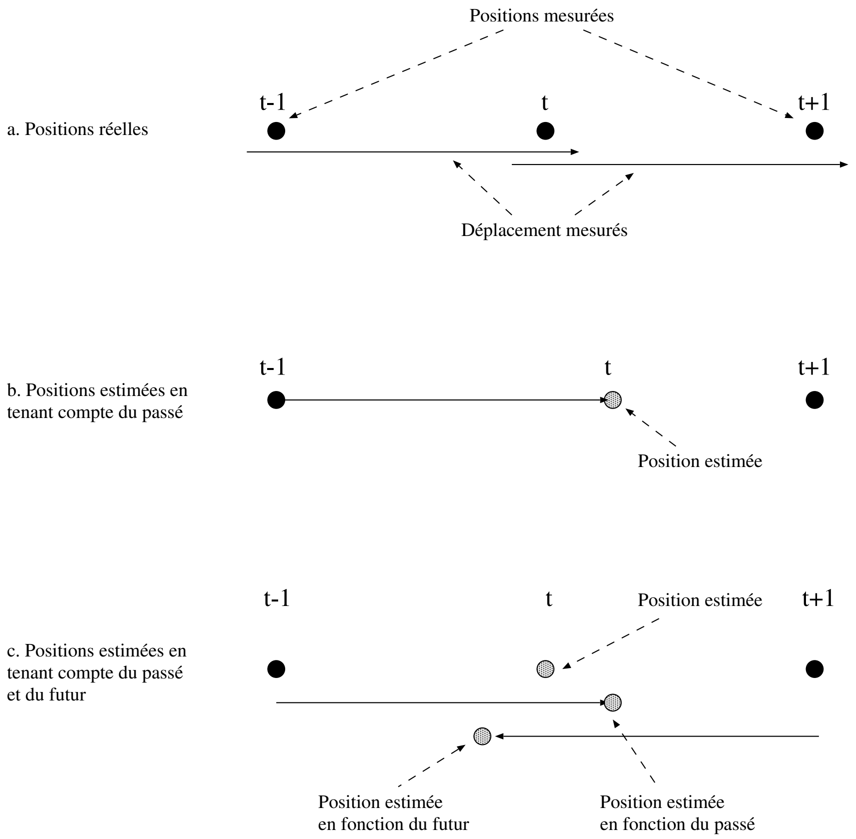


Fig. *3.8: Exemple de l'intérêt de la prise en compte du futur pour l'estimation de la position. Dans cet exemple, la position au temps t doit être estimée, connaissant les positions au temps $t-1$ et $t+1$ et le déplacement de $t-1$ à t et de t à $t+1$. La mesure de ces déplacement est surestimée. L'utilisation du passé seul conduira donc à une mauvaise estimation de la position dûe à la surestimation du premier déplacement (b). L'utilisation du futur, par contre, permet de réaliser un compromis entre les erreurs de mesure des deux déplacements et d'estimer la position de manière plus précise (c).*

fournie par le robot.

3.3.3 Cartographie «en-ligne»

La cartographie «hors-ligne» à partir d'un ensemble de données permet d'utiliser toute l'information disponible à chaque instant pour estimer au mieux la carte de l'environnement. Toutefois, du point de vue de l'autonomie robotique, il est souhaitable de pouvoir estimer «en-ligne» une carte correcte de l'environnement, quitte à effectuer des modifications plus profondes si de nouvelles informations sur les positions passées du robot sont disponibles. Des approximations des modèles précédents peuvent être utilisés pour ce faire.

Comme les informations sur les positions passées du robot sont principalement disponibles lors de la fermeture de cycles dans l'environnement, les modèles correspondant fonctionnent comme les algorithmes de cartographie incrémentale précédemment évoqués, mais présentent un

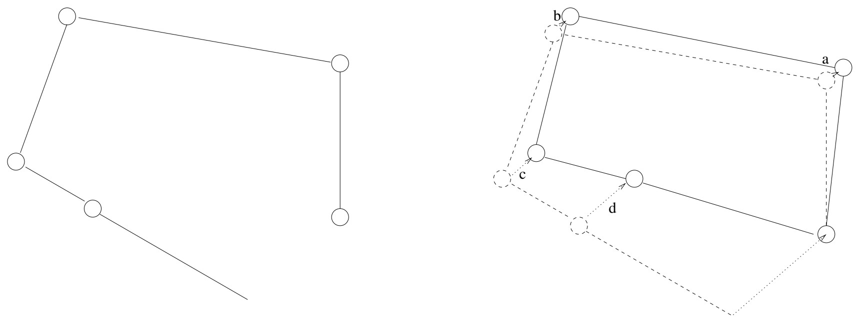


FIG. *3.9: Pour assurer la cohérence globale d'une carte, il est possible de modifier chacune des relations le long du cycle incohérent afin de le rendre cohérent. Cette méthode est similaire à une modification assurant localement la cohérence de la carte, mais la modification est propagée avec une amplitude décroissante en fonction de l'éloignement de la connexion dans le cycle. Ainsi, les modifications d, c, b et a qui permettent la fermeture du cycle sont de plus en plus faibles.*

comportement différent lorsqu'un tel cycle est bouclé. Ce comportement vise à assurer de manière globale la cohérence de la carte, ce qui revient à modifier les informations de position du passé pour les rendres cohérentes avec les informations courantes. Dans un tel cas, les informations sur la position du robot sont donc propagées le long du cycle afin de rendre compte au mieux de la forme de ce cycle.

Une première méthode pour effectuer ces modifications consiste à propager le long du cycle la modification due à la fermeture du cycle avec une amplitude décroissante en fonction de la distance [Thrun et al., 2000] (cf. figure 3.9). Toutefois, cette méthode ne permet d'assurer la modification et la cohérence de la carte le long d'un cycle que lors de la fermeture de ce cycle. Elle peut se révéler problématique si plusieurs cycles ayant des parties communes existent dans l'environnement puisque la fermeture d'un second cycle peut détruire la cohérence du premier.

La seconde méthode consiste à considérer le graphe composant la carte comme un ensemble de ressorts dont les longueurs au repos sont données par la distance mesurée par l'odométrie du robot entre les éléments de la carte [Gutmann and Konolige, 2000, Lu and Milios, 1997, Duckett et al., 2000, Hafner, 2000] (cf. figure 3.10). La fermeture d'un cycle non cohérent conduit donc à un système de ressorts qui n'est pas en équilibre. Pour estimer au mieux la forme du cycle, il suffit alors de calculer la position de repos de l'ensemble de ressorts. Cette position d'équilibre correspond en effet au minimum d'énergie potentielle du réseau de ressorts et donc à la position des ressorts dans laquelle leur écart à la longueur de repos est la plus faible. La carte cohérente ainsi obtenue est celle dont les différences avec les longueurs mesurées sont les plus faibles. L'avantage de cette seconde méthode est que la relaxation permet de garantir à tout moment la cohérence d'un nombre arbitraire de cycles enchevêtrés.

Cette capacité de cartographie «en-ligne», qui vise à prendre en compte toutes les informations disponibles sur la position du robot, est clairement la plus appropriée au cas d'un robot complètement autonome.

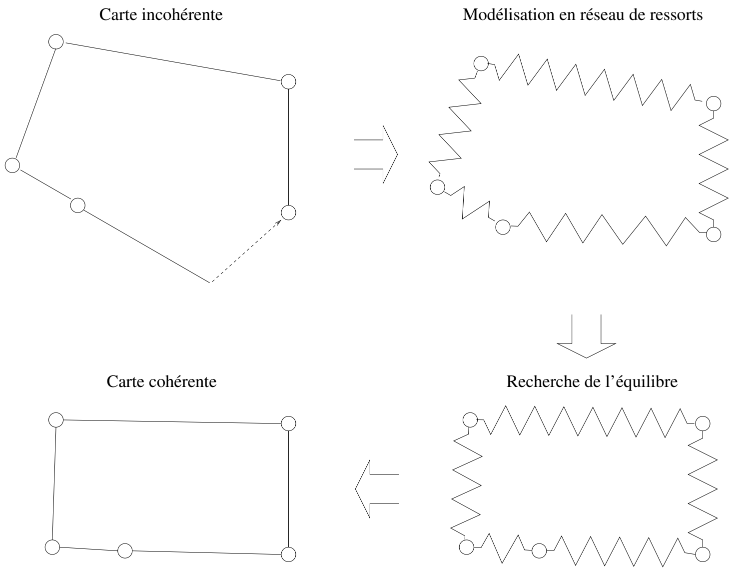


FIG. *3.10: Pour assurer la cohérence d'une carte, il est possible de modéliser le graphe des positions mémorisées dans la carte comme un réseau de ressorts, la longueur à vide des ressorts étant donnée par la distance mesurée par le robot entre les positions. Une carte incohérente se traduit alors par un réseau de ressorts en déséquilibre. Calculer la position d'équilibre du réseau permet d'obtenir une carte cohérente qui minimise les différences entre les longueurs des connexions de la carte et la longueur mesurée des connexions.*

3.4 Conclusion

La recherche de l'autonomie pour le système de navigation décrit dans cette thèse nous a donc conduit à développer un système de cartographie «en-ligne». De plus, pour permettre une cartographie efficace d'environnements cycliques, ce système doit être capable d'intégrer des informations a posteriori sur les positions passées du robot. Ce type de système s'intègre assez naturellement au sein d'une carte topologique, par le simple maintien de la cohérence globale. Nous avons choisi d'assurer cette cohérence globale à l'aide d'une modélisation du graphe de la carte par un système de ressorts car cette méthode permet de maintenir la cohérence de la carte quels que soient la forme et le nombre des cycles enchevêtrés.

Toute la difficulté du processus de cartographie, associé à la méthode de localisation globale que nous avons choisie, tient dans l'estimation de la nouveauté ou non du lieu courant du robot. En effet, si ce lieu a déjà été visité, il faut simplement le retrouver dans la carte pour le mettre à jour. Si, au contraire, il n'a jamais été atteint précédemment, il faudra l'y ajouter. Dans ce second cas, un suivi de position est utilisé temporairement pour estimer la position de ce lieu nouveau par rapport à la carte en cours de construction. L'estimation de la nouveauté ou non d'un lieu est réalisée dans notre modèle grâce à l'heuristique mentionnée à la fin du chapitre précédent qui permet de décider si la distribution de probabilité représente de manière correcte ou non la position courante de robot.

De plus, notre système utilise une stratégie d'exploration qui lui permet, d'une part, d'explorer et de cartographier l'environnement de manière exhaustive et, d'autre part, de limiter la distance que parcourt le robot en terrain inconnu. Cette deuxième fonction à pour but de garantir une estimation correcte de la position en limitant le temps pendant lequel le système ne peut pas utiliser la carte pour cette estimation.

Chapitre 4

Planification

Dans ce court chapitre, nous allons rapidement présenter le processus de planification afin de justifier la méthode que nous avons choisie dans notre modèle. Cette partie du processus de navigation n'a toutefois pas été centrale dans notre thèse, et nous resterons donc à un niveau très général d'autant que la planification est un vaste domaine de recherche relativement séparé de la cartographie et de la localisation. De fait, les problèmes importants de la planification surviennent lors de l'utilisation d'espaces de recherche de grandes dimensions ou de robots non holonomes [Latombe, 1991, Laumond, 1998]. Or en robotique mobile holonome, l'espace considéré est en général à deux dimensions, ce qui permet d'utiliser des techniques de planification relativement simples. Les méthodes présentées ici s'appuient sur une carte connue de l'environnement, et sur une position de l'animat correctement estimée au sein de cette carte. Les modèles mentionnés dans ce chapitre sont détaillés dans une publication [Filliat and Meyer, 2001b].

4.1 Exécution d'un plan

Avant de chercher à calculer une séquence d'actions pour rejoindre un but, il convient de préciser comment ces actions peuvent être exécutées. Cette exécution, en effet, est confrontée à deux problèmes. Le premier problème vient des différences qui existent toujours entre le monde réel et le modèle qui en est fait. Ces différences, dues par exemple aux éléments dynamiques de l'environnement, peuvent conduire à des plans qui seront irréalisables en pratique. Le second problème vient directement de l'exécution des actions par le robot. En effet, ces actions peuvent ne pas produire les déplacements escomptés, par exemple en cas de glissement des roues du robot. Ainsi, même un plan correct pourra ne pas conduire au but recherché.

Ces problèmes ont empêché les premiers robots utilisant des *contrôleurs hiérarchiques* (cf. figure 4.1) de fonctionner de manière efficace dans les environnements réels [Murphy, 2000]. En effet, ces architectures permettent peu de contrôles sur l'exécution des actions et reposent fortement sur l'hypothèse que le modèle du monde est correct.

Brooks [Brooks, 1991] a proposé une solution radicale à ces problèmes sous la forme d'une *architecture réactive*. Dans cette architecture, un ensemble de comportements réactifs, fonction-

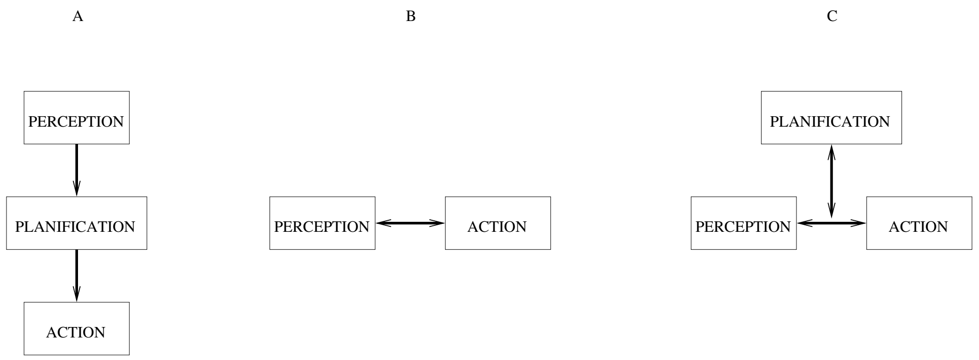


FIG. *4.1: Architectures classiques des contrôleurs pour les robots mobiles : Hiérarchique (A), Réactive (B) et Hybride (C) (Voir le texte pour les détails).*

nant en parallèle, contrôle le robot sans utiliser de modèle du monde. Cette architecture supprime évidemment les problèmes cités dûs aux différences entre la réalité, d'une part, et le modèle de l'environnement du robot, d'autre part, mais limite clairement les tâches que peut effectuer le robot (cf. [Kirsh, 1991] pour une critique). En effet, sans représentation interne de l'état de l'environnement, il est très difficile de planifier une suite d'actions en fonction d'un but à atteindre. Les robots utilisant cette architecture sont donc en général efficaces pour la tâche précise pour laquelle ils ont été conçus, dans l'environnement pour lequel ils ont été prévus, mais sont souvent difficiles à adapter à une tâche différente.

La plupart des contrôleurs actuellement utilisés choisissent une solution intermédiaire entre ces deux approches sous la forme d'une *architecture hybride* [Murphy, 2000, Arkin, 1989]. Cette architecture se compose de deux niveaux. Le premier est chargé des tâches de navigation de haut niveau, telles que la localisation, la cartographie et la planification. Pour cela, il s'appuie sur un second niveau réactif qui est chargé d'exécuter les commandes avec le plus de précision possible et de gérer les éléments non modélisés de l'environnement tels que les obstacles dynamiques. L'action conjointe de ces deux niveaux permet de réagir rapidement face aux variations imprévues de l'environnement, tout en permettant la réalisation d'actions planifiées à plus long terme. C'est ce type architecture que nous utiliserons dans notre modèle.

4.2 Discrétisation de l'espace de recherche

Nous nous concentrons dans cette thèse sur les techniques de planification dans des espaces discrets car, comme nous l'avons mentionné dans le paragraphe 1.4.1, les cartes topologiques fournissent directement une discrétisation de l'environnement réel utilisable par ces techniques. Nous pourrons donc les utiliser directement dans notre modèle.

Dans le cas des cartes métriques, qui représentent l'espace de manière continue, ces techniques ne sont utilisables qu'après discrétisation de l'espace libre représenté dans la carte. Pour ce faire, certains modèles intègrent directement cette décomposition au niveau de la cartographie, en construisant une carte topologique parallèlement à la carte métrique [Arleo et al., 1999, Chatila and Laumond, 1985, Thrun, 1999, Buhmann et al., 1995]. D'autres modèles font appel à des décompositions de l'espace libre spécifiques à la planification. De plus, différentes techniques, tels les champs de potentiel [Latombe, 1991, Dudek and Jenkin, 2000, Murphy, 2000], permettent de calculer des chemins directement dans le domaine continu, sans phase préalable de discrétisation. Nous ne décrirons cependant pas ces techniques dans cette thèse.

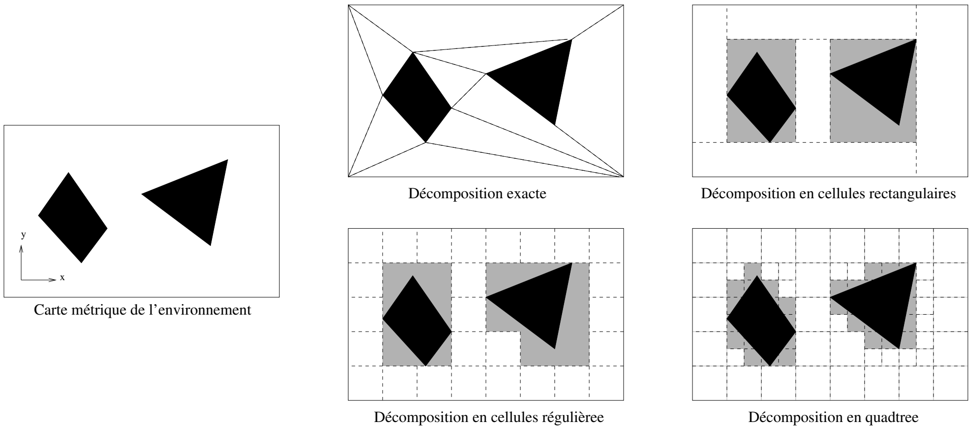


FIG. *4.2: Exemples de décompositions en cellules de l'espace libre dans les cartes métriques. La décomposition exacte permet de représenter l'ensemble de l'espace libre, à l'aide de cellules de formes irrégulières. La décomposition en cellules rectangulaires ne représente qu'un sous-ensemble de l'espace libre suffisant pour la planification. Ces cellules peuvent être de taille régulière ou non. Enfin une représentation hiérarchique telle que le «quadtree» permet d'utiliser des cellules de taille variable en fonction de la complexité locale de l'environnement.*

Il existe deux catégories de méthodes pour discrétiser l'espace de recherche des cartes métriques. Les méthodes de la première catégorie font appel à des décompositions en cellules, de différents types, qui permettent de reproduire la topologie de l'espace libre [Latombe, 1991, Murphy, 2000] (cf. figure 4.2). Les cellules obtenues sont alors utilisées de manière similaire aux nœuds des cartes topologiques dans le processus de planification, les cellules adjacentes étant considérées comme reliées par une arête.

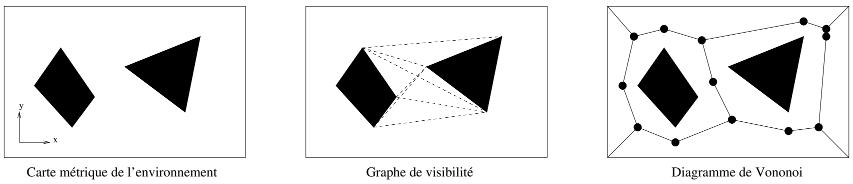


FIG. *4.3: Exemples de décompositions en chemins pré-calculés dans les cartes métriques. Ces chemins relient des points particuliers utiles pour la navigation et répartis dans l'environnement. Différents choix de points sont possibles. Le graphe de visibilité utilise les angles d'obstacles qui sont les points que le robot devra contourner pour éviter ces obstacles. Le diagramme de Voronoï utilise les points équidistants de plusieurs obstacles qui permettent de générer des chemins passant le plus loin possible des obstacles.*

Les méthodes de la seconde catégorie font appel au pré-calcul de chemins entre des points répartis dans l'environnement [Latombe, 1991, Laumond, 1998] (cf. figure 4.3). Les point seront utilisés comme les nœuds d'une carte topologique, tandis que les chemins pré-calculés reliant les nœuds seront utilisés comme les arêtes de cette carte.

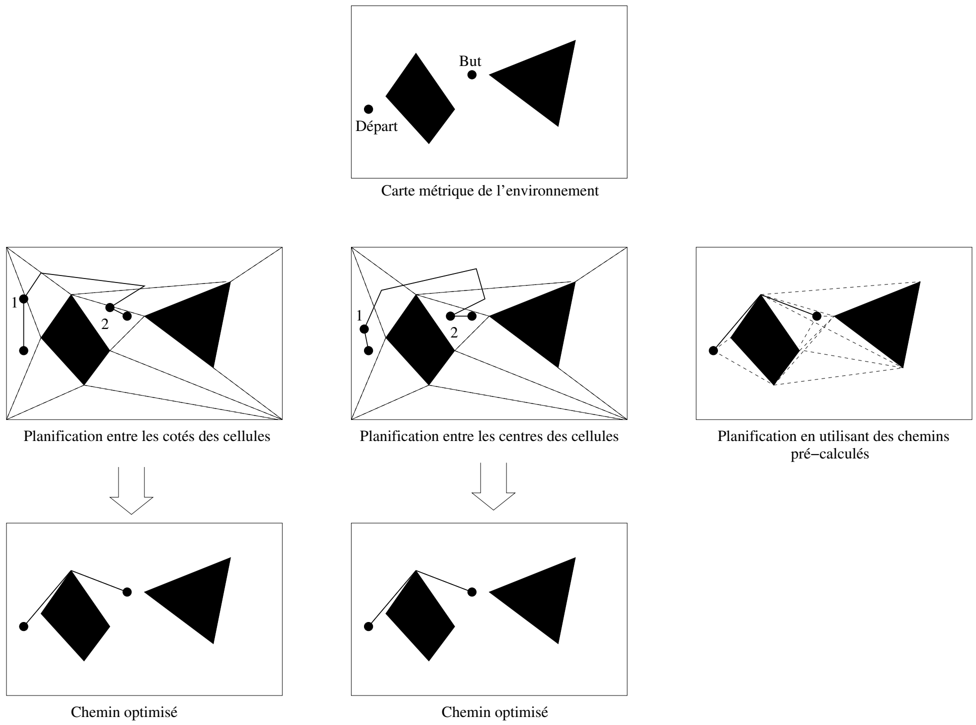


FIG. *4.4: Exemple de planification de chemin dans une carte métrique. Deux portions de trajectoire sont calculées pour relier le point de départ et le but à des points de l'espace discrétisé (points 1 et 2 dans cet exemple). Un chemin est ensuite calculé dans l'espace discrétisé entre ces deux points. La trajectoire résultante peut ensuite être optimisée pour supprimer les effets de la discrétisation.*

La planification du chemin entre deux points de l'environnement se réalise alors en deux étapes. La première étape permet de calculer un chemin direct entre, d'une part, le point de départ et le point le plus proche dans l'espace discrétisé et, d'autre part, le point de l'espace discrétisé le plus proche du but et le but en question. La seconde étape permet ensuite de calculer un chemin entre ces deux points de l'espace discrétisé, en utilisant une des méthodes décrites dans le prochain paragraphe. Ces trois parties de trajectoires sont ensuite assemblées pour obtenir le chemin reliant le point de départ au but. Une phase d'optimisation supplémentaire peut être utilisée pour limiter les effets de la discrétisation et lisser la trajectoire (cf. figure 4.4). Dans le cas de la décomposition de l'espace libre en cellules, les points de l'espace discrétisé utilisés peuvent être les centres des cellules ou les milieux des côtés des cellules. Dans le cas de l'utilisation de chemins pré-calculés, ces points sont simplement les points de passage de ces chemins.

4.3 Planification

A partir d'une carte métrique discrétisée ou d'une carte topologique, il existe différentes méthodes pour calculer un chemin entre la cellule de départ et la cellule but. Nous distinguons ici les méthodes selon le type de plans qu'elles génèrent.

4.3.1 Deux types de plan

Le premier type de plan qui peut être généré contient une suite d'actions à effectuer par le robot, ou une suite de points à atteindre afin de rejoindre le but. Les algorithmes classiques de recherche dans les graphes, tels que l'algorithme de Dijkstra $A^{\star}$, ou l'une de ses nombreuses variantes, peuvent être utilisés pour calculer ce type de plan [Levitt and Lawton, 1990, Kortenkamp et al., 1994, Kuipers, 2000, Scholkopf and Mallot, 1995, Nourbakhsh et al., 1995, Arleo et al., 1999, Dudek and Jenkin, 2000]. La taille raisonnable des cartes topologiques classiquement utilisées en robotique rend ces algorithmes suffisamment efficaces en pratique. Ce type de plan pose toutefois des problèmes lors de son exécution si le robot ne parvient pas à atteindre l'un des points du chemin calculé, ou s'il s'éloigne de la trajectoire et que sa position correspond à un nœud qui ne fait pas partie du chemin planifié. La solution à ces problèmes est alors de recommencer le processus de planification en prenant en compte la nouvelle position de départ. Ce processus de replanification est souvent inutilement coûteux en calcul car un grand nombre des opérations nécessaires auront déjà été effectuées lors de la planification précédente.

Un second type de plan, qui associe à chacune des positions possibles du robot au sein de la carte l'action qu'il doit effectuer pour atteindre son but, peut être utilisé. Ce type de plan est appelé *politique* ou *plan universel* [Schoppers, 1987]. Le résultat est alors une stratégie de déplacement similaire à la stratégie *d'action associée à un lieu* mentionnée dans l'introduction. L'enchaînement de reconnaissances de positions et de réalisations des actions associées à ces positions permet donc de générer une route joignant le but. Ce type de plan présente l'avantage de permettre au robot d'atteindre le but, aussi longtemps qu'il possède une estimation correcte de sa position. En effet, le chemin précis rejoignant le but n'est pas spécifié et le robot peut donc s'écarter du chemin direct entre la position initiale et le but sans entraîner de replanification.

Une politique est plus lourde à calculer que les plans du type précédent car toutes les positions de la carte doivent être envisagées, sans utiliser les heuristiques des algorithmes précédents qui permettent de restreindre l'exploration de l'espace de recherche. Toutefois, cette augmentation est rapidement compensée si le robot s'écarte du chemin direct vers le but. Dans ce cas, en effet, la planification doit être reprise pour le premier type de plan, alors que c'est inutile pour une politique. Le calcul d'une politique reste donc en général praticable pour les cartes de taille limitée typiques de la robotique mobile.

Pour calculer une telle politique, une simple recherche en largeur dans le graphe en partant du but peut être utilisée. Cette méthode se retrouve sous le nom de *breadth first search*, *spreading activation* [Mataric, 1992, Bachelder and Waxman, 1995] ou *wavefront propagation* [Murphy, 2000].

Ces deux derniers noms viennent de l'analogie entre l'ordre de parcours du graphe et la manière dont un fluide progresserait s'il s'échappait du but pour se répandre dans le graphe.

Une seconde méthode pour calculer une telle politique est l'utilisation de l'algorithme de *value iteration* [Buhmann et al., 1995, Thrun, 1999, Burgard et al., 1998] qui est un algorithme de programmation dynamique. Un tel algorithme associe une valeur V_i représentant la distance du but à chaque nœud de la carte. Initialement, la valeur associée au but est donc ajustée à 0, tandis que celle des autres cellules est mise à $+\infty$. L'algorithme met alors à jour itérativement la valeur de chaque nœud en lui associant la valeur la plus faible parmi ses voisins plus une valeur représentant le coût du déplacement d'un nœud à son voisin. Après convergence, la valeur associée à chaque nœud augmente en fonction de la distance du nœud au but. L'action associée à chacun des nœuds est alors calculée par une simple descente de gradient sur ces valeurs, en associant à chaque position l'action menant à la position connexe ayant la valeur V_i la plus faible.

Les plans sous forme de politique semblent, de plus, particulièrement adaptés à une implémentation s'inspirant de la biologie. En effet, il est relativement simple d'imaginer des méthodes permettant d'associer une action à chaque cellule de lieu, plutôt que de calculer et mémoriser un plan sous forme d'une suite explicite d'actions à exécuter. Après une telle association, le choix de l'action à exécuter dépend alors simplement de l'activation des cellules de lieu qui représente la position courante de l'animat. Ce type de plan est donc utilisé par la plupart des systèmes de navigation inspirés de la biologie [Arleo and Gerstner, 2000, Burgess et al., 1994, Trullier and Meyer, 2000].

4.3.2 Choix de l'action avec une position incertaine

Lorsque la position estimée par le système de localisation est non ambiguë, l'utilisation d'une politique se résume simplement au choix de l'action associée avec la position courante. Toutefois, les systèmes réalisant le suivi de plusieurs hypothèses, particulièrement ceux basés sur les PDMPO, fournissent également une estimation de la probabilité de présence du robot en différentes positions. Il peut donc se révéler utile de tenir compte de ces probabilités pour sélectionner l'action à exécuter.

La méthode permettant de prendre en compte ces probabilités de manière optimale serait de calculer la politique optimale pour l'utilisation du PDMPO correspondant. Une telle politique associe une action, non pas à chaque état du PDMPO, mais à chaque distribution de probabilité sur cet ensemble d'états. Les politiques résultantes sont extrêmement efficaces, car elles prennent en compte l'incertitude de l'estimation de la position, en plus de la position la plus probable. Elles peuvent, par exemple, contenir des actions qui écartent temporairement le robot de son but si ces actions permettent de réduire l'incertitude sur l'estimation de l'état et donc d'arriver globalement de manière plus efficace au but. Malheureusement, le calcul d'une telle politique est complexe et n'est pas réalisable en pratique pour des PDMPO de plus de quelques dizaines d'états [Kaelbling et al., 1998]. Ce calcul est en particulier irréalisable pour des systèmes tels que le nôtre qui contiennent plusieurs milliers d'états.

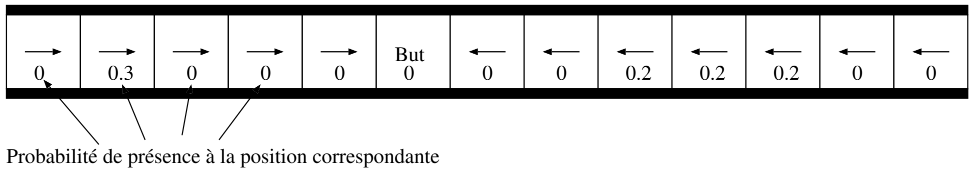


FIG. *4.5: Exemple de l'intérêt d'une procédure de vote dans le cas où la situation du robot est incertaine. Si l'on choisit l'action associée à la position la plus probable, le robot ira à droite, ce qui correspondra à l'action correcte avec une probabilité* $0,3$. *En utilisant une méthode de vote, l'action choisie sera d'aller à gauche, ce qui sera correct avec une probabilité* $0,6$.

Différentes méthodes permettent toutefois de prendre ces probabilités en compte de manière utilisable en pratique. Une première méthode consiste à utiliser une méthode de vote [Simmons and Koenig, 1995, Cassandra et al., 1996]. Pour cela, une action est simplement associée à chacun des nœuds de la carte, en utilisant une des méthodes décrites au paragraphe précédent. Un score est alors calculé pour chaque action. Ce score est la somme des probabilités des nœuds auxquels chaque action est associée. L'action ayant le score le plus élevé est alors exécutée. Cette méthode est efficace dans les cas de grande ambiguïté dans la localisation, où la probabilité de la position la plus probable est seulement très légèrement supérieure aux autres. Dans ce cas, en effet, si la direction associée à la position la plus probable est incorrecte, cette méthode permet de l'ignorer et de choisir une direction associée à plusieurs autres hypothèses de position qui se révélera correcte dans un plus grand nombre de cas (cf. figure 4.5). C'est ce type de stratégie que nous avons choisi d'utiliser dans notre modèle.

Les politiques optimales de PDMPO mentionnées en début de paragraphe possèdent l'avantage de permettre de planifier des actions dans le seul but de diminuer l'incertitude de l'estimation de la position. De telles actions ne seront pas planifiées dans l'approche précédente. Toutefois, le choix de telles actions peut également être réalisé au moment du choix de l'action et non au moment de la planification. Il est ainsi possible de choisir de se diriger vers le but si la confiance en la position courante est suffisamment élevée, ou de chercher à améliorer l'estimation de sa position dans le cas contraire. Cette confiance peut être simplement mesurée par l'entropie de la distribution de probabilité [Cassandra et al., 1996, Thrun, 2000] (cf. paragraphe 2.4). Ainsi, si l'entropie de la distribution de probabilité représentant la position est trop élevée, une action permettant de diminuer cette entropie sera sélectionnée. L'utilisation de telles stratégies permet par exemple d'éviter des zones dans lesquelles l'incertitude de localisation est plus grande (par exemple les larges espaces ouverts), et de privilégier les zones plus favorables à l'estimation de la position (par exemples les zones où se trouvent des points de repère fiables).

4.4 Conclusion

L'utilisation de plans sous forme de politique associant une action à chaque position de la carte permet une exécution robuste du plan aussi longtemps que la position du robot est estimée correctement. Nous avons donc choisi ce type de plan pour le modèle de navigation développé dans cette thèse. Le plan est calculé en utilisant un simple algorithme de *spreading activation*.

Notre modèle permettant également de calculer une probabilité de présence du robot en chaque nœud de la carte, nous utiliserons une méthode de vote pour sélectionner l'action à exécuter à chaque pas de temps.

Enfin, l'exécution de ce plan sera assurée par une architecture hybride dans laquelle des procédures de bas-niveau permettront d'éviter les obstacles dynamiques de l'environnement. Nous avons de plus utilisé une méthode qui permet de vérifier l'exécution correcte du plan, qui permettra de reprendre la planification si un plan se révèle impossible à exécuter.

Deuxième partie

Notre modèle

Chapitre 5

Notre modèle

Après avoir présenté les différents problèmes qui surviennent lors de la conception d'un système de navigation et les différents choix que nous avons effectués, nous allons maintenant présenter notre modèle en détails. Nous commencerons par résumer les caractéristiques dont nous avons souhaité doter ce modèle.

5.1 Choix stratégiques

Deux idées ont présidé aux choix que nous avons réalisés lors de la conception de notre modèle. La première est la volonté de se situer à mi-chemin entre une approche d'inspiration purement biologique et une approche purement robotique de type ingénieur. La seconde est la recherche d'une autonomie maximale pour le robot dans toutes les phases de navigation.

Notre modèle utilise une carte topologique dense de l'environnement, c'est-à-dire que les lieux associés aux différents nœuds sont très proches les uns des autres. Les arêtes de cette carte mémorisent une information sur la position métrique relative des nœuds qu'elle rejoint, sans qu'aucune information métrique de position globale ne soit mémorisée. La cohérence de la carte est assurée par une méthode globale, utilisant un modèle de réseau de ressorts.

La méthode de localisation retenue est similaire à celle utilisée dans les modèles basés sur des PDMPO (cf. section 2.3). Ainsi la position est représentée par une distribution de probabilité sur l'ensemble des nœuds de la carte qui permet d'intégrer toutes les informations disponibles pour le robot en s'inspirant des équations de mise à jour des PDMPO. Notre modèle permet, de plus, la gestion des capteurs directionnels, d'une part en s'accommodant de l'incomplétude de l'information lors de l'intégration à l'estimation de la position et, d'autre part en utilisant des procédures de perception active pour sélectionner la direction des capteurs la plus pertinente.

La méthode de cartographie utilisée est une méthode "en-ligne" qui est capable d'intégrer des informations a posteriori sur des positions passées du robot (cf. section 3.3).

Enfin, nous utiliserons une méthode simple de planification utilisant une politique associant l'action à effectuer pour rejoindre le but à chaque nœud. A partir de cette politique et de l'estimation courante de la position, nous choisirons la direction à prendre pour le robot par une méthode de vote. L'exécution correcte du plan résultant sera de plus contrôlée afin de détecter une éventuelle impossibilité à suivre le chemin prescrit. La planification pourra alors être reprise en tenant compte de cette impossibilité.

5.2 Structure du modèle

La carte implémentée dans notre modèle est une carte topologique dense, dont chaque nœud N_i permet de mémoriser les perceptions que le robot pourra obtenir au lieu correspondant de l'environnement (cf. figure 5.1). Chaque nœud contient des données allothétiques omnidirectionnelles O_i et des données allothétiques directionnelles D_i (cf. section 1.3.2). Le nombre de passages en ce lieu, V_i ainsi que la date du dernier passage, T_i, sont également mémorisés.

L'arête A_{ij}, reliant les nœuds N_i et N_j, contient des informations métriques sur la position relative des lieux représentés par les deux nœuds. Les arêtes joignant deux nœuds en sens inverse sont symétriques, c'est-à-dire que nous imposons $A_{ij} = -A_{ji}$. Deux types de relations sont mémorisées. La première, R_{ij}^{Brute}, est la position relative mesurée par l'odométrie du robot entre les deux lieux représentés par les deux nœuds ainsi reliés. Cette relation mémorise directement les valeurs mesurées et ne garantit donc pas la cohérence globale de la carte. La seconde relation, $R_{ij}^{Coherente}$, correspond à la position relative des deux nœuds qui garantit la cohérence de la carte. Ces valeurs sont calculées à partir des valeurs R_{ij}^{Brute}, à l'aide d'une méthode de relaxation décrite dans la section 5.4.2. La carte étant cohérente, il est possible de calculer la position Pos_{N_i} de chaque nœud N_i par rapport à un nœud choisi comme référence. Le nombre de passages, V_{ij}, ainsi que la date du dernier passage, T_{ij}, le long de chaque arête sont également mémorisés.

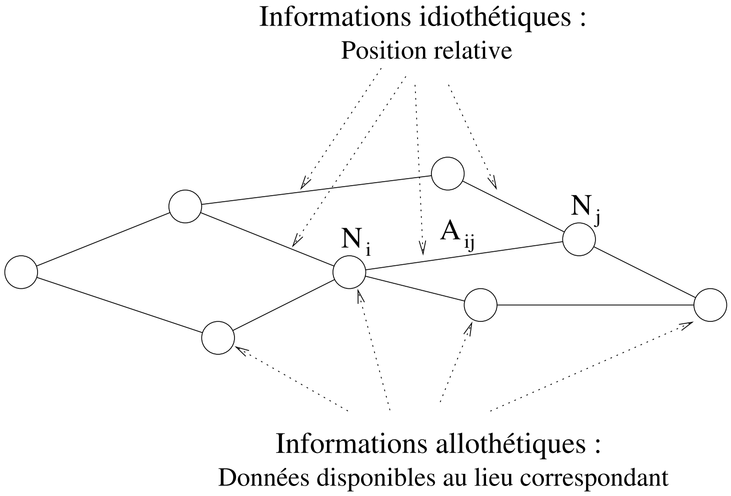


FIG. *5.1: Le type de carte utilisé dans notre modèle de navigation. Chaque nœud N_i de la carte mémorise les données allothétiques que le robot peut percevoir à la position correspondante de l'environnement. Les nœuds N_i et N_j sont reliés par l'arête A_{ij} qui permet de mémoriser les informations idiothétiques reliant ces deux positions.*

Une activité $Act(N_i)$ est associée à chaque nœud N_i de la carte. Cette activité permet d'estimer la probabilité que le robot se trouve à la position correspondante de l'environnement. Elles est calculée, comme nous le décrirons plus loin, en intégrant les données idiothétiques et allothétiques.

L'algorithme de localisation et de cartographie exécute les étapes suivantes chaque fois que la position du robot doit être estimée :

– Calcul de l'activité pour tous les nœuds de la carte.
– Reconnaissance du nœud courant ou création d'un nouveau nœud.
– Mise à jour des données allothétiques pour le nœud reconnu.
– Mise à jour des paramètres des connexions entre le nœud précédent et le lieu courant en utilisant les données odométriques.

Ces étapes vont être détaillées dans la suite, après une présentation des données utilisées par notre modèle.

5.2.1 Données allothétiques

Notre modèle traite différemment les capteurs de type directionnel et omnidirectionnel. Il permet de choisir une direction vers laquelle diriger les capteurs directionnels afin, d'une part, d'améliorer la qualité de l'estimation de la position et, d'autre part, de construire une carte la plus complète possible de l'environnement.

Capteurs omnidirectionnels

Les capteurs omnidirectionnels fournissent directement une représentation complète de l'environnement tel qu'il peut être vu par le robot, en couvrant toutes les directions autour du robot. Dans notre modèle, ces données sont orientées par rapport à un référentiel absolu. Puisque ces données sont complètes, la comparaison des données mémorisées et des données perçues permet d'estimer directement la similarité perceptuelle des lieux auxquels elles correspondent (cf. paragraphe 1.3.2). Notre modèle requiert seulement deux procédures pour utiliser de telles données.

La première procédure permet de mémoriser de nouvelles données. Elle permet également de mettre à jour des données anciennes lorsque de nouvelles informations provenant de la même position sont disponibles. Cette mise à jour, par l'intégration successive des données provenant d'un même lieu, doit permettre de filtrer un bruit blanc pour converger vers une représentation fiable des informations caractérisant la position considérée.

La seconde procédure est une procédure qui doit calculer la similitude de deux perceptions. Cette procédure $P_{Omn}(O_1, O_2)$, à valeurs entre 0 et 1, doit être maximale lorsque les données sont identiques, et tendre vers 0 lorsque les différences entre les données augmentent. Cette procédure est utilisée pour approximer la probabilité qu'une perception corresponde à chacun des nœuds de la carte.

Capteurs directionnels

Les capteurs directionnels fournissent une information uniquement pour une portion de l'environnement autour du robot. Notre modèle suppose cependant que la direction absolue de ces capteurs est connue. Nous supposons de plus que l'ensemble des directions possibles pour ces capteurs est discret. La comparaison de données n'est pas aussi simple que dans le cas de capteurs omnidirectionnels, car il faut tenir compte du fait que les données sont partielles. C'est pourquoi trois procédures sont utilisées pour traiter de telles données.

La première procédure permet la mémorisation des données. Comme pour les données omnidirectionnelles, cette procédure permet également de mettre à jour les données mémorisées lorsque de nouvelles données concernant la même position et la même direction sont disponibles. Le nombre de perceptions effectuées dans chacune des directions est également mémorisé. Cette information sera utilisée par le système de navigation afin de sélectionner la direction dans laquelle orienter le capteur.

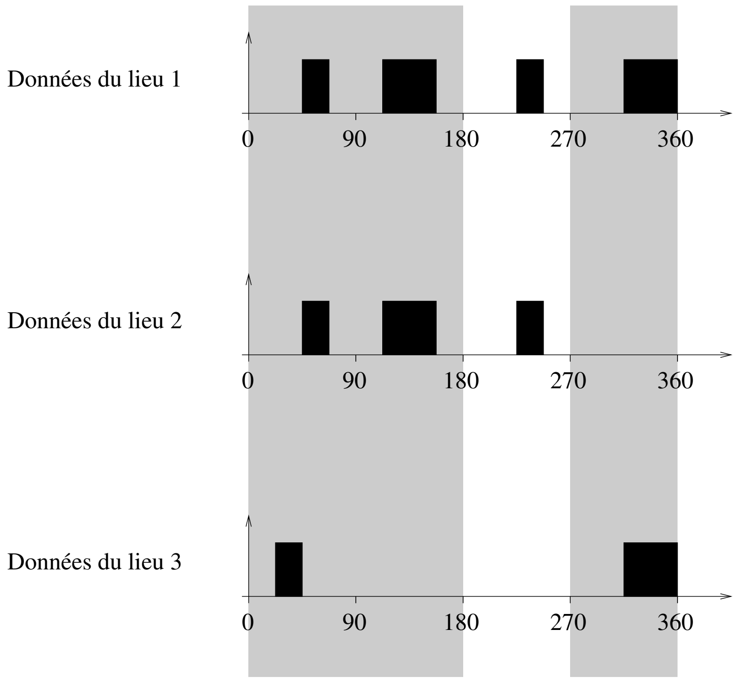


FIG. *5.2: Un exemple illustrant l'effet de la comparaison de données partielles. Les trois graphiques indiquent les données mémorisées en fonction de la direction pour trois lieux différents. Dans cet exemple, si la comparaison des lieux est effectuée sur les directions comprises entre 0 et 180 degrés, les lieux 1 et 2 seront identiques, tandis que le lieu 3 sera différent. Si la comparaison est effectuée entre 270 et 360 degrés, les lieux identiques seront les lieux 1 et 3, tandis que le lieu 2 sera différent.*

La seconde procédure permet de comparer des données et d'estimer la probabilité qu'elles proviennent de positions perceptuellement identiques. La similitude $P_{Dir}(D_1, D_2)$ de deux ensembles de données directionnelles D_1 et D_2 est calculée pour les directions dans lesquelles des données sont présentes dans les deux ensembles. Elle est maximale et vaut 1 lorsque toutes les données communes sont identiques, et décroît lorsque les différences entre les données augmentent. Dans le cas où il n'y a aucune donnée commune, la valeur de la similitude est fixée à -1. Il est très important de noter que cette similitude ne porte que sur des données partielles. Ainsi, il est impossible d'avoir une valeur qui reflète fidèlement la similarité globale entre un lieu perçu et les lieux mémorisés dans la carte. La seule information totalement fiable, dans ce cadre, est que deux lieux identiques ont une similarité égale à 1. Deux lieux différents ayant des données identiques dans quelques directions, cependant, pourront aussi avoir une similarité égale à 1 (cf. figure 5.2). La qualité de cette estimation dépend en fait très fortement de la proportion de données communes. Le fait que cette information soit partielle sera géré lors de la mise à jour de la distribution de probabilité représentant la position.

Enfin la troisième procédure, $I_{Dir}(D_1, D_2)$, permet de prendre en compte l'incomplétude de l'information. Elle mesure, pour deux ensembles de données allothétiques directionnelles D_1 et D_2, le nombre de directions pour lesquelles les deux ensembles de données contiennent des informations.

5.2.2 Données idiothétiques

Les capteurs odométriques fournissent des informations sur la position des lieux visités par le robot. Ces informations ne sont pas utilisées de manière globale, mais uniquement pour calculer les positions relatives des lieux. Cette approche permet de limiter l'influence du bruit de l'odométrie sur la cartographie, mais notre modèle suppose cependant que l'erreur sur la direction du robot reste bornée. Cela est difficile à obtenir par la seule intégration des mouvements des roues d'un robot et requiert donc une estimation séparée de la direction, en utilisant un compas magnétique ou un gyroscope.

Les données odométriques sont modélisées à l'aide d'une fonction qui permet d'estimer la probabilité du déplacement réel du robot. Cette fonction $P_{Depl}(A, od, B)$ estime la probabilité que le déplacement du robot, mesuré par l'odométrie od, corresponde au déplacement depuis le point A jusqu'au point B.

5.3 Estimation de la position

L'activité des nœuds de la carte est mise à jour en utilisant une procédure en plusieurs étapes. A chaque fois que la position du robot doit être estimée, une première étape prend en compte les déplacements du robot, puis des étapes successives traitent les informations provenant des différents systèmes de perception.

5.3.1 Choix des lieux à représenter

Dans les cartes topologiques, le choix des lieux à représenter se base en général sur les données allothétiques. Celles-ci sont utilisées pour détecter des lieux pertinents pour la navigation. Cette méthode requiert cependant un suivi constant des données allothétiques afin de pouvoir détecter ces lieux pertinents. Cette opération est en général simple pour des capteurs ne demandant pas beaucoup de traitements, comme les sonars, mais peut être difficile pour des capteurs plus complexes, comme les caméras. La mesure en temps réel des variations des données allothétiques peut alors se révéler difficile.

Pour cette raison, deux méthodes sont utilisées dans notre modèle pour décider à quel instant effectuer les opérations de localisation et de mise à jour de la carte. La première méthode calcule, lorsque cela est possible en temps réel, la similarité entre les données allothétiques courantes et les données recueillies lors de la dernière localisation. Lorsque cette similarité tombe en dessous d'un certain seuil, une nouvelle étape de localisation est effectuée.

La seconde méthode est utilisée lorsque les données allothétiques ne permettent pas un calcul rapide des similarités. Dans ce cas, la localisation est effectuée chaque fois que le robot a parcouru une distance fixe `Seuil_Localisation` depuis la dernière localisation. Cette méthode, indépendante des données allothétiques, est bien adaptée à notre modèle car la carte topologique utilisée est dense et couvre tout l'environnement. Elle ne serait pas utilisable si la carte ne représentait que des positions particulières de l'environnement, comme des intersections de couloirs.

5.3.2 Intégration des données idiothétiques

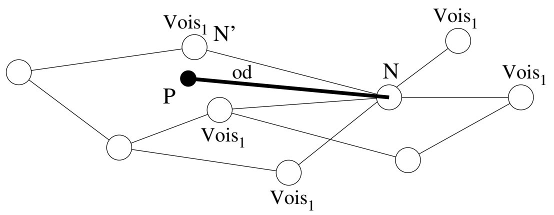


FIG. *5.3: Illustration de la méthode utilisée pour intégrer les données idiothétiques aux activités des nœuds. Pour chaque nœud N, la position P du robot au pas de temps précédent - position qui aurait conduit au nœud N - est calculée grâce aux relevés od de l'odométrie du robot depuis le pas de temps précédent. Le nœud N' pour lequel le produit de l'activité au pas de temps précédent par la probabilité que le déplacement de N' à N corresponde au déplacement od est alors recherché parmi l'ensemble $Vois_1$ des nœuds directement connectés au nœud N. L'activité du nœud N est alors calculée en fonction de l'activité de N' au pas de temps précédent et de la différence de position entre N' et P.*

Les informations idiothétiques récentes sont utilisées dans une procédure prenant en compte la structure locale de la carte. La position de chacun des nœuds est calculée en utilisant les données des connexions, relativement à un nœud de référence. Ces positions sont utilisées pour calculer la nouvelle activité des nœuds selon l'équation :

$$Act_t(N_n) = \max_{n' \in Vois_1(N_n)} (Act_{t-1}(N_{n'}) \times P_{Depl}(Pos_{N_{n'}}, od, PosN_n)) \qquad (5.1)$$

où $Act_t(N_i)$ est l'activité du nœud N_i au temps t, od la position relative du robot au pas de temps précédent mesurée par l'odométrie, $Vois_p(N_i)$ l'ensemble des nœuds de la carte connectés à N_i par un chemin contenant moins de p connexions, et $P_{Depl}(Pos_{N_i}, od, Pos_{N_j})$, la fonction permettant de modéliser la probabilité que le déplacement od mesuré par l'odométrie corresponde au déplacement du nœud N_i au nœud N_j.

Ainsi, pour chaque nœud N_n, la procédure détermine pour quel nœud $N_{n'}$, le produit de l'activité de ce nœud au pas de temps précédent par la probabilité que le déplacement depuis ce nœud corresponde à l'odométrie est le plus grand. La nouvelle activité du nœud N_n est alors l'activité précédente du nœud $N_{n'}$, modulée par la probabilité que le déplacement du robot corresponde effectivement à ce déplacement (cf. figure 5.3).

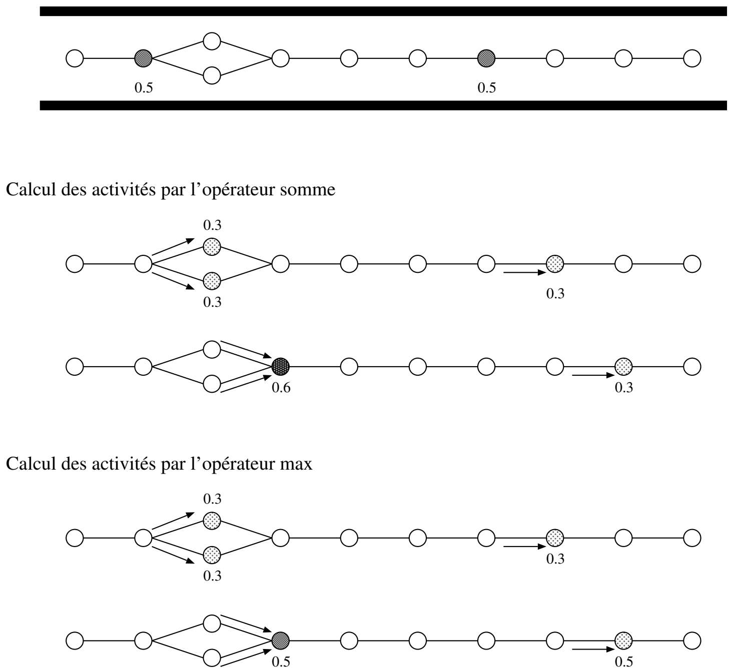


FIG. *5.4: Illustration des biais introduits par l'utilisation de l'opérateur* somme *dans notre modèle. Dans cet exemple, la carte représente un couloir rectiligne dans lequel une erreur de cartographie a dédoublé un nœud représentant une position. En utilisant l'opérateur* somme *et après normalisation des activités, cette erreur conduira à une plus grande confiance accordée à la position correspondante. L'utilisation de l'opérateur* max *permet de réduire l'influence de tels dédoublements de nœuds qui sont assez fréquents dans les cartes produites par notre modèle.*

Cette équation est le pendant de l'équation 2.1 (cf. chapitre 2) pour le calcul des probabilités après une action au sein des PDMPO. Dans notre cas, cependant, nous avons choisi d'utiliser l'opérateur *max*, au lieu de l'opérateur *somme*. En effet, l'utilisation de l'opérateur *somme* dans notre modèle conduit à des biais dans l'estimation des probabilités qui proviennent de la discrétisation irrégulière de l'environnement fournie par la carte topologique (cf. figure 5.4).

Nous montrerons expérimentalement dans le prochain chapitre que ces biais conduisent à une estimation moins précise de la position.

Cette équation de mise à jour des activités ne prend en compte que les nœuds directement connectés au nœud dont l'activité est calculée. Lorsque la connectivité de la carte est incomplète, c'est-à-dire lorsque les nœuds représentant des positions existent, mais ne sont pas connectés à tous les nœuds proches directement accessibles, ce choix conduit à des biais dans l'estimation des activités. Un tel effet est particulièrement sensible lors du parcours de cycles dans l'environnement et empêche alors une reconnaissance correcte du nœud représentant la position du robot (cf. figure 5.5). Comme nous le verrons dans la suite, la position du robot dans ce cas est néanmoins estimée de manière correcte grâce à l'utilisation directe des données idiothétiques.

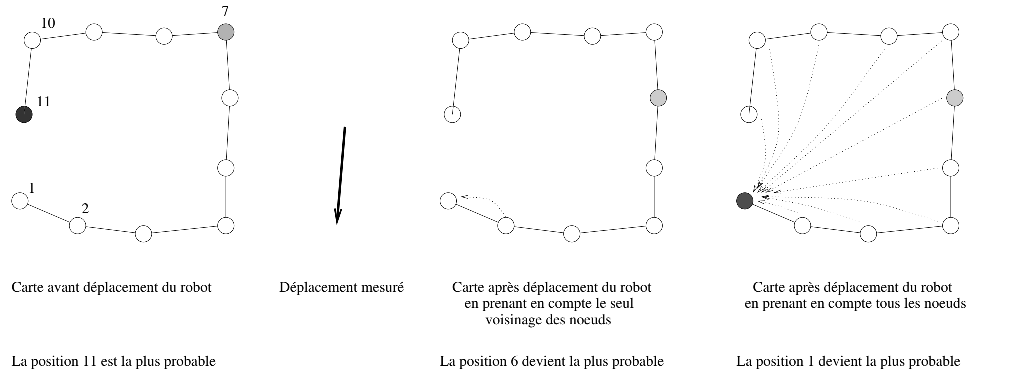


FIG. *5.5: Illustration de l'utilité de prendre en compte tous les nœuds pour la mise à jour de l'activité. Dans cet exemple, le robot se trouve initialement à la position du nœud* 11 *et se déplace directement jusqu'au nœud* 1 *pour boucler le cycle. Si seuls les nœuds proches sont pris en compte pour calculer l'activité du nœud* 1*, son activité sera faible car le seul nœud qui lui est connecté (le nœud* 2*) ne correspond pas à la position précédente du robot. Si au contraire tous les nœuds de la carte sont pris en compte, l'influence du nœud* 11 *permettra d'estimer correctement l'activité du nœud* 1*.*

Comme les positions relatives de tous les nœuds ont été calculées, il pourrait donc paraître intéressant, pour une meilleure estimation des activités, de prendre en compte tous les nœuds de la carte dont la position est proche du nœud considéré, indépendamment de l'existence ou non d'une connexion entre ces nœuds. Deux raisons, toutefois, s'opposent à ce choix. Premièrement, l'absence de connexions entre deux nœuds proches peut résulter de l'existence d'un obstacle entre les deux positions, aussi bien que de l'incomplétude de la carte. Transmettre l'activité aux nœuds proches reviendrait dans ce cas à «traverser les murs». La carte de notre modèle étant une carte topologique ne représentant pas les obstacles de manière explicite, il est impossible de déterminer pour chaque transition dans quel cas nous nous trouvons.

La seconde raison pour un calcul de l'activité ne prenant en compte que les nœuds connectés est la complexité algorithmique de la prise en compte de tous les nœuds. Cela conduit en effet à une complexité quadratique de la mise à jour en fonction du nombre de nœuds de la carte, ce qui devient problématique lorsque ce nombre de nœuds augmente. En pratique, la procédure est inutilisable en temps réel lorsque le nombre de nœuds de la carte atteint le millier. La complexité du calcul ne prenant en compte que les seuls nœuds connectés ne dépend, pour sa part que du facteur de branchement de la carte topologique et du nombre de nœuds de la carte. Comme nous le verrons par la suite, ce dernier est limité par les procédures de nettoyage de la carte. Ainsi la complexité globale de la mise à jour est linéaire en fonction du nombre de nœuds de la carte.

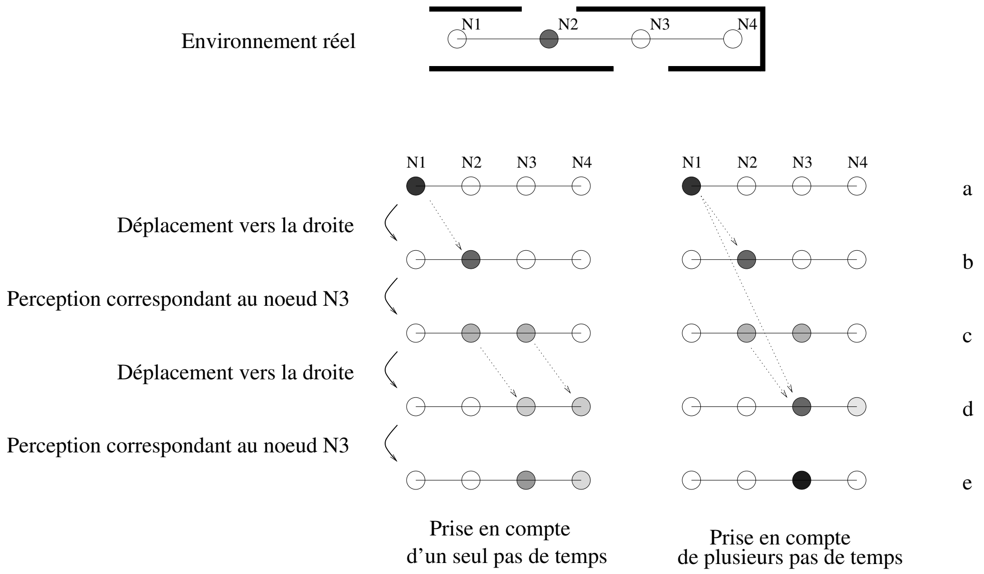


Fig. *5.6: Illustration de l'intérêt de la prise en compte de plusieurs pas de temps. Le robot se situe initialement au niveau du nœud $N1$ et sa position est correctement estimée (Partie a). Il se déplace ensuite vers le nœud $N2$ (Partie b). Au niveau du nœud $N2$, une erreur de perception survient et le nœud $N3$ est détecté, à la place du nœud $N2$. Cette information est toutefois intégrée par la localisation (Partie c). Lors du déplacement suivant, vers le nœud $N3$, la prise en compte de plusieurs pas de temps permet de retrouver directement une estimation correcte de la position, tandis que la prise en compte du seul pas de temps précédent conduit à une estimation moins précise (Partie d). La perception correcte du nœud $N3$ conforte alors ces estimations (Partie e). Au final, l'estimation prenant en compte plusieurs pas de temps se révèle plus précise.*

En revanche, dans le cas de cartes incomplètes, ou dans le cas d'erreurs de perceptions temporaires, il est possible de prendre en compte plusieurs pas de temps en arrière pour calculer l'activité des nœuds. Par exemple, lorsqu'une seule perception est erronée, si un seul pas de temps est pris en compte, l'activité du nœud correspondant à la position réelle du robot va fortement diminuer et restera faible pendant plusieurs pas de temps. Si, au contraire, plusieurs pas de temps sont pris en compte, dès la perception suivante, l'activité du nœud correct sera très élevée (cf. figure 5.6). L'utilisation de plusieurs pas de temps correspond à l'hypothèse que l'estimation de la position au sein de notre modèle ne vérifie pas l'hypothèse d'indépendance de Markov et dépend donc de plusieurs pas de temps passés. Cette hypothèse qui

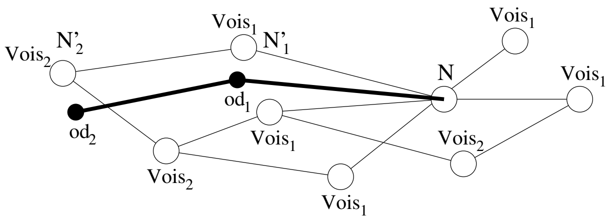


FIG. 5.7: *Illustration de la méthode utilisant plusieurs pas de temps dans l'intégration des données idiothétiques aux activités des nœuds. La procédure décrite par la figure 5.3 est utilisée pour plusieurs pas de temps précédant le pas de temps courant. L'activité de chaque nœud N est alors le maximum des activités calculées pour les différents pas de temps.*

se révèle utile dans notre modèle n'est toutefois pas nécessaire dans plusieurs autres systèmes de navigation qui ont montré que l'hypothèse de Markov permettait une localisation efficace [Simmons and Koenig, 1995, Burgard et al., 1996, Fox et al., 1998b, Thrun et al., 1999].

Cette prise en compte est réalisé par une modification de la fonction permettant la mise à jour des activités (cf. figure 5.7) :

$$Act_t(N_n) = \max_{p \in [1..P]} \left(\max_{n' \in Vois_p(N_n)} (Act_{t-p}(N_{n'}) \times P_{Depl}(Pos_{N_{n'}}, od_p, PosN_n)) \right) \tag{5.2}$$

où od_p est la position relative du robot p pas de temps auparavant, position mesurée par l'odométrie.

Après cette mise à jour, la somme des activités des nœuds de la carte est calculée. Nous verrons par la suite que cette valeur est utilisée au cours du processus de localisation.

5.3.3 Intégration des données allothétiques omnidirectionnelles

L'intégration des données allothétiques omnidirectionnelles $O_{percues}$ est relativement simple car toutes les données sont disponibles. La comparaison des données perçues avec les données mémorisées permet donc d'avoir directement une estimation de la probabilité d'avoir effectué cette perception à la position correspondant à chaque nœud. Cette probabilité est intégrée à la distribution de probabilité grâce à l'équation suivante :

$$Act_t(N_n) = Act_{t-1}(N_n) \times P_{Omn}(O_n, O_{percues}) \tag{5.3}$$

où $P_{Omn}(O_i, O_j)$ mesure la similarité entre les données O_i et O_j.

L'effet de cette équation est d'augmenter les activités des nœuds les plus similaires aux perceptions courantes et de diminuer celle des nœuds les plus différents. Cette équation est le pendant direct de l'équation 2.1 qui permet d'intégrer des observations dans l'estimation de l'état des PDMPO.

5.3.4 Intégration des données allothétiques directionnelles

Du fait de l'incomplétude des informations recueillies par les capteurs directionnels, il est nécessaire de choisir la direction de ces capteurs avant d'intégrer leurs informations. Le choix de cette direction peut dépendre de l'estimation courante de la position du robot et doit chercher à fournir l'information la plus pertinente possible dans le contexte courant. Ce choix est donc fait par une procédure de perception active qui distingue deux cas pour orienter les capteurs directionnels :

- Le premier cas est celui où l'estimation courante de la position du robot est jugée précise et fiable. Les informations des capteurs directionnels ne sont alors pas nécessaires à la localisation. La direction choisie est donc une direction qui n'a pas encore été explorée à partir de la position courante du robot, afin de servir à compléter la représentation mémorisée en cette position.
- Le second cas est celui où l'estimation courante de la position est imprécise. Le choix de la direction du capteur est alors importante pour obtenir une meilleure estimation de la position. Différentes procédures sont utilisables dans ce cas : elles sont détaillées dans la section 5.7.

Dans notre modèle, le choix entre ces deux alternatives est fait en fonction de la plus forte activité de la carte. Si cette activité est supérieure à un certain seuil, la localisation est considérée comme précise et la première alternative est choisie. Sinon, c'est la seconde alternative qui est choisie pour améliorer la précision de la localisation.

Lorsque les données ont été acquises, elles sont utilisées pour mettre à jour les activités des nœuds. Cette mise à jour doit cependant prendre en compte le fait que les informations sont partielles. En effet, se contenter de multiplier l'activité de chaque nœud par sa similarité avec les données perçues, comme dans le cas des capteurs omnidirectionnels, conduirait à de forts biais dans l'estimation de la position. Il faut, de plus, gérer le cas où il n'y a rien de commun entre les données mémorisées et les données perçues. En effet, ignorer les nœuds correspondants reviendrait à multiplier leur activité par 1, ce qui impliquerait une identité entre les données perçues et mémorisées.

Une solution à ce problème pourrait consister à ignorer les données qui n'apportent pas d'informations pour tous les nœuds de la carte et à ne pas mettre à jour les activités lorsque de telles données sont disponibles. Cette solution est envisageable lorsque la carte de l'environnement est presque complète, car il manque alors peu de données. Elle représente toutefois une grande perte d'information lorsque la carte est en cours de construction. En effet, dans une carte partielle, l'information concernant l'ensemble des nœuds ne sera disponible que dans très peu de directions, voir aucune. La solution retenue dans notre modèle consiste à toujours utiliser les informations disponibles, même lorsqu'elles ne concernent qu'une partie des nœuds. Cette solution repose sur deux éléments. Le premier concerne le choix des valeurs à affecter aux nœuds pour lesquels aucune information n'est disponible, le second concerne la manière d'intégrer les informations partielles à la carte.

La valeur que nous avons associée aux nœuds pour lesquels aucune information n'est disponible est la valeur moyenne des similarités mesurées pour les autres nœuds $P_{Dir}^{Moyenne}$. Cette valeur sera évidemment fausse, mais permet d'éviter un biais qui surestimerait ou sous-estimerait systématiquement la probabilité de se trouver au lieu correspondant. De plus, cette valeur aura une influence faible si la direction choisie permet d'avoir des informations sur la plupart des nœuds qui constituent les positions les plus probables du robot. Ainsi, ces valeurs fausses seront utilisées pour des zones qui correspondent à une faible probabilité de présence et joueront un rôle moins important pour l'estimation de la position.

Pour l'intégration des informations partielles à l'activité des nœuds de la carte, nous avons choisi de pondérer l'influence des données des capteurs directionnels en fonction de la proportion de nœuds sur lesquels ces données apportent de l'information. Ainsi, si des données allothétiques perçues n'apportent d'information sur aucun nœud, leur effet sera nul sur l'estimation de l'activité. Si, au contraire, elles apportent de l'information sur tous les nœuds de la carte, leur effet sera similaire à celui des données omnidirectionnelles. Le coefficient de pondération choisi est donc le rapport du nombre de nœuds ayant des données communes avec les données perçues sur le nombre total de nœuds de la carte, pour chacune des directions de perception possibles. Le coefficient α est donc donné par :

$$\alpha = \frac{\sum_{N_i \in Carte} I_{Dir}(D_i, D_{Percue})}{\sum_{N_i \in Carte} I_{Dir}(D_{Complete}, D_{Percue})} \tag{5.4}$$

L'équation de mise à jour utilisant ce paramètre est alors :

$$Act_t(N_n) = Act_{t-1}(N_n) \times P_{Dir}(D_n, D_{percues})^{\alpha} \quad si \quad P_{Dir}(D_n, D_{percues}) \neq -1 \tag{5.5}$$

$$Act_t(N_n) = Act_{t-1}(N_n) \times (P_{Dir}^{Moyenne})^{\alpha} \quad si \quad P_{Dir}(D_n, D_{percues}) = -1 \tag{5.6}$$

Dans le cas où la perception n'apporte aucune information ($\alpha = 0$), cette équation ne modifie pas l'activité des nœuds. Si, au contraire, la perception apporte des informations sur tous les nœuds, elle est identique à celle utilisée pour les capteurs omnidirectionnels. Cette équation est alors identique à l'équation 2.1.

L'activité des nœuds de la carte est ensuite normalisée, afin que leur somme reste égale à 1.

5.3.5 Estimation de la position par les données idiothétiques

Comme nous allons le voir dans la section suivante, la position du robot est essentiellement reconnue grâce à l'activité des nœuds de la carte. Cependant, lorsque la position réelle du robot ne se trouve plus dans la zone couverte par la carte, il est nécessaire d'utiliser directement les données allothétiques pour estimer la position du robot.

Les nœuds de la carte étant relativement espacés, il ne suffit pas d'estimer cette position en utilisant les données allothétiques depuis le dernier nœud reconnu. Cette méthode conduit en effet très rapidement à une dérive de l'estimation de la position du fait de la discrétisation des positions représentées. Cela a pour conséquence d'empêcher la construction de cartes correctes de l'environnement. Deux améliorations ont donc été apportées à cette première idée.

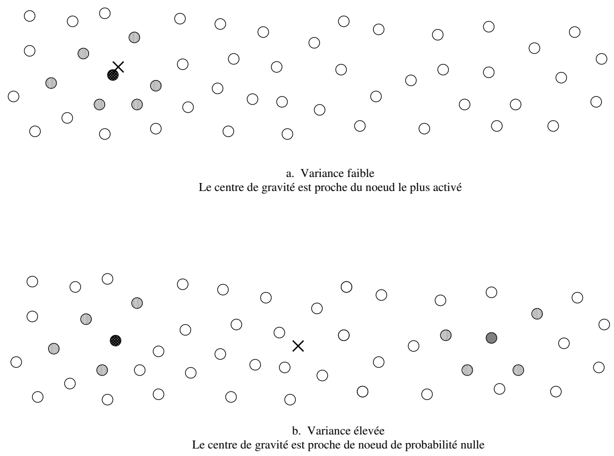


FIG. *5.8: Le codage par population de vecteurs n'a de sens que si les activités des nœuds de la carte sont centrées autour d'un point. Dans ce cas, en effet, la position calculée est proche d'un nœud qui représente la position du robot avec une forte probabilité (partie a). Si les activités sont dispersées, du fait que plusieurs hypothèses de position éloignées existent, le codage par population de vecteurs n'a plus de sens. En effet, il pourra alors donner une position qui se trouve dans une zone où la probabilité de présence du robot est nulle. Dans notre modèle, ces deux cas sont détectés grâce à la variance de l'estimation de la position.*

La première amélioration est l'utilisation du codage par population de vecteurs. Au lieu d'utiliser la position du nœud précédent comme point de départ des données idiothétiques, le centre de gravité des nœuds pondérés par leur activité au pas de temps précédent est utilisé. Cela permet de représenter la position du robot de manière continue, et non de manière discrète, ce qui permet d'augmenter la précision correspondante. Le codage par population de vecteurs n'est toutefois pas utilisable dans tous les cas. En effet, il faut pour cela que les activités des nœuds de la carte soient centrées autour d'un point (cf. figure 5.8). Un tel cas est détecté dans notre modèle à l'aide d'un seuil `Seuil_Variance` sur la variance du barycentre des nœuds. Si cette variance est supérieure à ce seuil, seule la position du nœud le plus activé est utilisée, sans codage par population de vecteurs.

La seconde amélioration est la prise en compte de plusieurs pas de temps dans le passé. En effet, la procédure décrite précédemment peut être utilisée pour tous les pas de temps précédents. La valeur retenue pour la position du robot est alors la moyenne des positions estimées depuis les pas de temps précédents, chaque position étant pondérée par un facteur qui décroît pour les estimations les plus anciennes.

5.3.6 Reconnaissance de la position

L'activité des différents nœuds ayant été calculée, il faut être capable de détecter si la position réelle du robot est déjà mémorisée dans la carte ou non. En effet, tous les calculs de l'activité se font en supposant que la position du robot se trouve au sein de la carte et visent à estimer cette position. Nous verrons plus loin comment notre modèle décide si le robot se trouve en un lieu connu ou non. Supposons pour le moment que cette question soit réglée.

Lorsque la position du robot se trouve effectivement dans la carte, le nœud reconnu comme la position courante du robot est simplement le nœud ayant la plus forte activité.

Dans le cas contraire, la position du robot est estimée en utilisant les données idiothétiques selon la méthode décrite dans la section précédente. Si la distance entre cette position et la position d'un nœud existant déjà dans la carte est inférieure à une distance fixe `Seuil_Reconnaissance`, celui-ci est reconnu comme le nœud courant. Sinon, un nœud est créé et devient le nœud représentant la position courante. Le cas dans lequel un nœud déjà existant est reconnu survient lorsque la position du robot est mal estimée par la distribution des activités des nœuds, mais lorsque le nœud représentant la position existe déjà. Cela se produit, par exemple, avant qu'un cycle de l'environnement ne soit terminé et mémorisé dans la carte (cf. figure 5.5). Cette méthode de reconnaissance des nœuds est donc extrêmement importante car elle permet de rattraper les erreurs d'estimation sur la nouveauté d'un lieu. En effet, si un lieu est incorrectement identifié comme nouveau, elle permet néanmoins de reconnaître le nœud qui le représente déjà, au lieu d'en créer un nouveau. En conséquence, la méthode de détection des lieux nouveaux pourra surestimer la nouveauté des lieux et détecter des «faux positifs», c'est-à-dire détecter comme nouveau des lieux déjà visités.

En gardant en mémoire le fait que nous devons plutôt surestimer la nouveauté des lieux, nous pouvons maintenant présenter la méthode de détection de cette nouveauté. Dans notre modèle, l'estimation de la nouveauté des lieux repose sur la détection du fait que le robot sort de la zone couverte par la carte ou non. En effet, si la position au pas de temps précédent est convenablement estimée par la distribution d'activités, cette activité est simplement «déplacée» dans la carte si la nouvelle position est au sein de la carte (cf. figure 5.9a). Si, au contraire, la nouvelle position se trouve en dehors de la carte, l'activité va «sortir» de la carte (cf. figure 5.9b). Dans notre modèle, cet événement peut être très simplement détecté en comparant la somme des activités des nœuds de la carte avant et après la prise en compte des données idiothétiques. Si cette somme augmente ou reste constante, le robot se trouve toujours dans une zone couverte par la carte ; si cette somme diminue, cela signifie que le robot est entré dans une zone inconnue.

L'irrégularité de la carte peut évidemment provoquer des détections abusives de zones nouvelles (cf. figure 5.10) mais, comme nous l'avons vu, cela est compensé par le processus de création de nœuds qui vérifie au préalable qu'aucun nœud n'existe à la position mesurée.

C'est cette heuristique qui nous permet d'utiliser une méthode de localisation globale tout en construisant la carte de l'environnement. Elle permet en effet de déterminer les cas dans lesquels

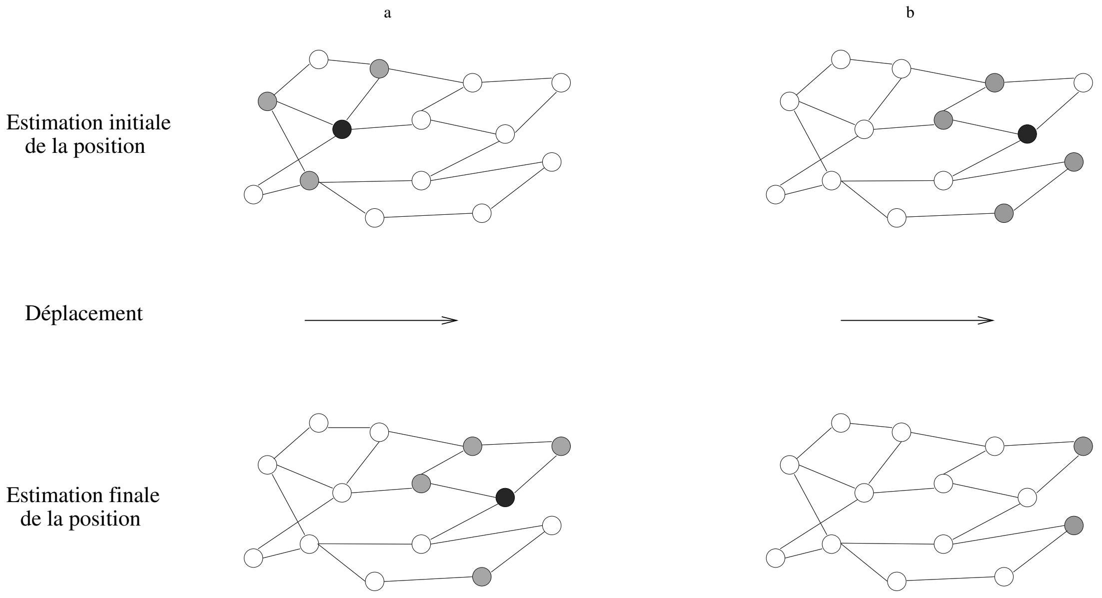


FIG. *5.9: La méthode d'estimation de la nouveauté d'un lieu repose sur la comparaison de la somme des activités des nœuds de la carte avant et après l'intégration des données idiothétiques. Lorsque le robot reste dans une zone couverte par la carte, cette somme reste approximativement constante (Partie a). Si le robot sort de la zone cartographiée pour arriver en un lieu nouveau, cette somme va, par contre, diminuer (Partie b).*

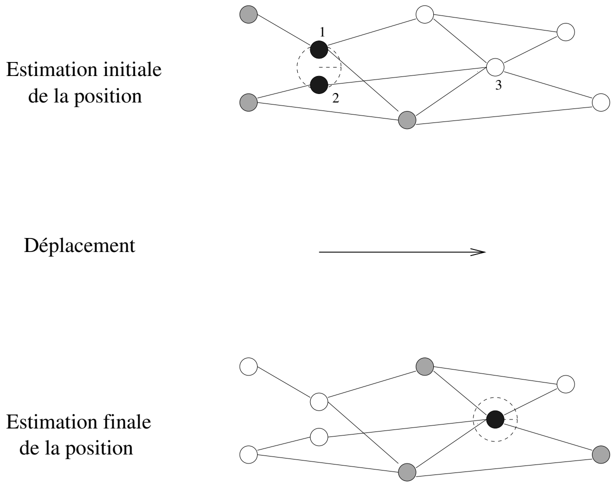


FIG. *5.10: Exemple de cas dans lequel l'irrégularité de la carte provoque des fausses détections de lieux nouveaux. Les nœuds* 1 *et* 2 *sont très proches et ont chacun une forte activité représentant la position du robot. Lorsque le robot se déplace, la méthode de déplacement des activités utilisée va conduire à une activité du même niveau, mais dans le seul nœud* 3*. En conséquence, la somme des activités des nœuds de la carte va diminuer, ce qui va provoquer la détection d'un lieu nouveau, alors que le nœud* 3 *aurait représenté parfaitement la nouvelle position du robot. De telles erreurs de détection sont compensées au moment de la création d'un nouveau nœud.*

ce modèle de l'environnement est pertinent et permet d'estimer correctement la position du robot. Dans les cas où ce modèle n'est pas pertinent, l'odométrie depuis les positions précédentes est alors utilisée pour pallier les manques du modèle et estimer néanmoins correctement la position. Cette heuristique joue donc un rôle central de «superviseur» pour estimer quand le modèle du monde peut être utilisé et quand il doit être amélioré.

5.4 Mise à jour de la carte

Lorsqu'un nœud de la carte a été reconnu ou créé comme position courante du robot, la carte est mise à jour en utilisant les données idiothétiques et allothétiques.

5.4.1 Mise à jour des nœuds

Les données allothétiques mémorisées dans le nœud reconnu sont simplement mises à jour grâce aux procédures qui leurs sont associées, en utilisant les données perçues par le robot. Si le nœud est nouveau, les données allothétiques qu'il mémorise sont simplement initialisées avec les données courantes.

5.4.2 Mise à jour des arêtes

Les données idiothétiques sont utilisées pour mettre à jour les informations brutes mémorisées dans les arêtes. Pour ce faire, le déplacement *od* mesuré par l'odométrie du robot depuis la reconnaissance du nœud précédent est calculé. Dans le cas où le nœud courant vient d'être créé, une connexion entre le nœud précédent et le nœud courant est créée et ses paramètres sont initialisés avec les valeurs de ce déplacement.

Dans le cas où le nœud courant existait déjà, la position relative rel du nœud courant et du nœud précédent sont calculées. Si la différence entre *od* et de rel est supérieure à un seuil `Seuil_Mise_A_Jour`, aucune mise à jour n'est effectuée dans la carte. Cela a pour but d'empêcher l'intégration dans la carte de données idiothétiques incohérentes lors de phases de relocalisation importantes. Dans un tel cas, en effet, la reconnaissance d'un nouveau nœud ne traduit pas simplement le déplacement du robot, mais aussi le changement d'hypothèse de position. Intégrer les données idiothétiques entre les deux nœuds reconnus successivement conduirait alors à une topologie fausse (cf. figure 5.11).

Ce même phénomène est cependant utile lorsque la relocalisation est due à une mauvaise cartographie, notamment lors de la fermeture d'un cycle de l'environnement (cf. figure 5.12). Dans ce cas, les informations idiothétiques doivent être intégrées pour corriger la carte. Un compromis est donc à trouver sur la valeur du seuil pour limiter l'intégration erronée de données idiothétiques mais permettre une cartographie efficace.

Lorsque la différence entre *od* et de rel est inférieure au seuil, la connexion directe entre le nœud précédent et le nœud courant est ajoutée si elle n'existait pas, ou mise à jour sinon. La valeur après la mise à jour est la moyenne vectorielle de l'ancienne valeur et de la valeur mesurée,

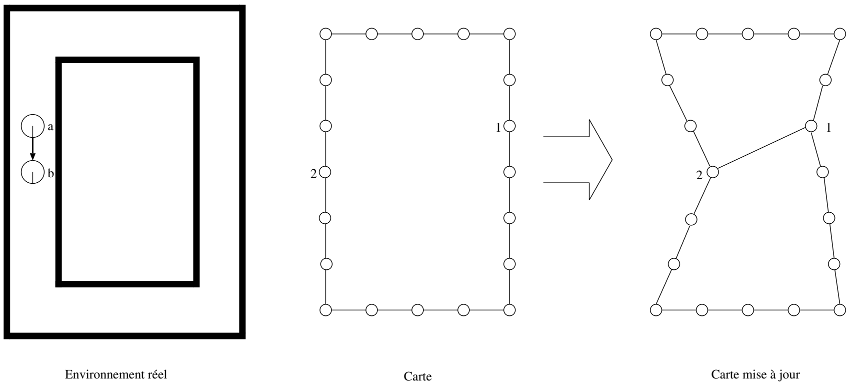


FIG. *5.11: Exemple de cas dans lequel les données idiothétiques ne doivent pas être intégrées à la carte. Le robot se déplace ici de la position* a *à la position* b*. L'estimation de cette position, qui était initialement fausse, passe du nœud 1 au nœud 2. Ajouter une connexion et les données idiothétiques relevées entre les nœuds 1 et 2 dans ce cas conduit à une carte à la topologie erronée. Pour limiter ce phénomène, l'ajout de connexion n'est réalisé que si la position relative des nœuds 1 et 2 et le déplacement mesuré par l'odométrie sont suffisamment proches.*

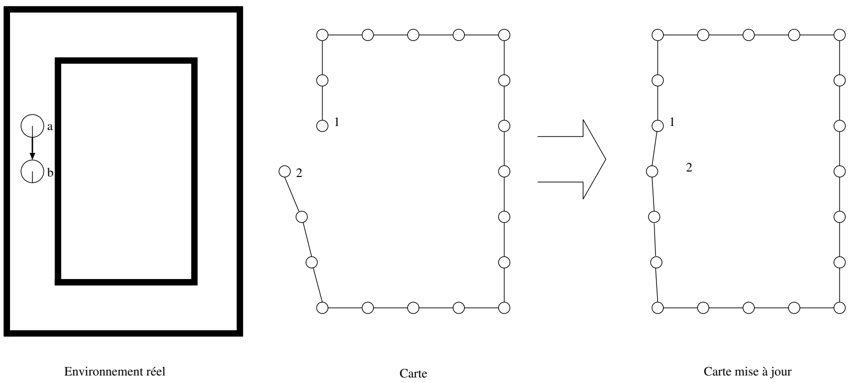


FIG. *5.12: Exemple de cas dans lequel les données idiothétiques doivent être intégrées à la carte. Le robot se déplace ici de la position* a *à la position* b*. L'estimation de cette position reste correcte et passe du nœud 1 au nœud 2. Dans ce cas, la position relative des nœuds 1 et 2 et le déplacement mesuré par l'odométrie sont différents à cause d'erreurs de la carte. Ajouter les données idiothétiques relevées entre les nœuds 1 et 2 permet donc d'améliorer la carte. Le seuil empêchant ce type de mises à jour doit donc être choisi pour permettre de tels phénomènes.*

l'ancienne valeur étant pondérée par le nombre de traversées de cette arête afin de stabiliser cette estimation :

$$R_{ij}^{Brute} = \frac{V_{ij} \times R_{ij}^{Brute} + od}{V_{ij} + 1} \tag{5.7}$$

où V_{ij} est le nombre de traversées de l'arête A_{ij}

A partir de ces données, les valeurs $R_{ij}^{Coherente}$ qui permettent d'assurer la cohérence de la carte sont calculées. Dans ce but, les arêtes de la carte sont considérées comme des ressorts dont la longueur à vide est la distance mesurée par l'odométrie du robot R_{ij}^{Brute}. Si la carte est incohérente, le système de ressorts ne sera pas en équilibre. Un algorithme permettant de calculer la position d'équilibre du système de ressorts est donc utilisé. Ces positions à l'équilibre seront prises comme valeurs de $R_{ij}^{Coherente}$ puisqu'elles correspondent au réseau cohérent dont les positions satisfont au mieux les contraintes données par les valeurs R_{ij}^{Brute}.

L'algorithme utilisé pour estimer la position d'équilibre est un algorithme itératif [Duckett et al., 2000]. Pour sa mise en œuvre, la position (x_i, y_i) de chaque nœud de la carte est calculée dans un cadre de référence global. Une variance est associée à la position de chaque nœud (v_i) ainsi qu'à la valeur R_{ij}^{Brute} de la position relative des nœuds (v_{ij}). La variance v_{ij} utilisée dans notre modèle est simplement fixée à dix pour cent de la longueur de la connexion correspondante, faisant donc l'hypothèse que la qualité de l'odométrie décroît linéairement avec la distance parcourue.

Le principe de cet algorithme est d'estimer la position de chaque nœud comme la moyenne des positions qu'il devrait avoir par rapport à chacun de ses voisins. L'application répétée de cette estimation conduit à la convergence de la position de chaque nœud vers la position qui minimise les écarts entre les position relatives réelles et les positions relatives voulues. Pour cela, l'algorithme itère donc les étapes suivantes :

– Estimation de la position relative (x_{ij}, y_{ij}) de chaque nœud N_i, par rapport à chacun de ses voisins N_j :

$$x_{ij} = x_j + x_{R_{ji}^{Brute}} \tag{5.8}$$

$$y_{ij} = y_j + y_{R_{ji}^{Brute}} \tag{5.9}$$

– Estimation de variance v_i de la position de chaque nœud :

$$\frac{1}{v_i} = \sum_j \frac{1}{v_{ji}} \tag{5.10}$$

– Estimation de la position de chaque nœud :

$$x_i = \sum_j \frac{x_{ij} v_i}{v_{ij}} \tag{5.11}$$

$$y_i = \sum_j \frac{y_{ij} v_i}{v_{ij}} \tag{5.12}$$

Lorsque l'algorithme a convergé, les valeurs $R_{ij}^{Coherente}$ sont calculées en fonction des positions obtenues pour chacun des nœuds.

5.4.3 Nettoyage de la carte

Afin de permettre de stabiliser la carte, certaines procédures additionnelles sont utilisées pour supprimer des nœuds ou des arêtes obsolètes.

Le critère que nous utilisons pour juger qu'un nœud est obsolète est un critère de fréquence de visite. Le système de cartographie supprime ainsi les nœuds rarement visités qui sont proches de nœuds fréquemment et récemment visités. Le fait qu'ils soient proches de nœuds récemment visités indique en effet que la zone correspondante de l'environnement a été visitée peu de temps auparavant. Si ces nœuds sont beaucoup moins visités que leurs proches voisins, cela correspond au fait qu'ils ne représentent pas de manière correcte cette portion de l'environnement. De tels nœuds sont donc supprimés de la carte si la date de leur dernière visite et le nombre de leur visites sont suffisamment inférieurs à la date de dernière visite et au nombre de visites d'un de leurs voisins. Le critère s'exprime de la façon suivante :

$$\textit{il existe } N_j \textit{ tel que } \begin{cases} Pos(N_i) - Pos(N_j) < \texttt{Seuil\_Localisation} \\ T_i < T_j - \texttt{Seuil\_Temps\_Oubli} \\ V_i < V_j \times \texttt{Seuil\_Visites\_Oubli} \end{cases} \tag{5.13}$$

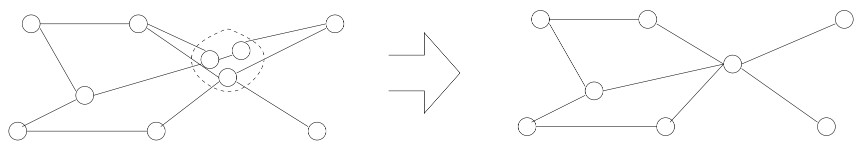


FIG. *5.13: Lorsque des nœuds de la carte se retrouvent trop proches, ils sont regroupés. Les données allothétiques associées au nouveau nœud sont les moyennes de celles associées aux nœuds qu'il remplace. Toutes les connexions aboutissant aux nœuds remplacés sont associées au nouveau nœud.*

La manière dont sont créés les nœuds de la carte garantit une distance minimale entre les nœuds de `Seuil_Reconnaissance`. En effet, avant de créer un nœud à une position donnée, le procédure de création des nœuds vérifie si un nœud n'existe pas déjà à une distance inférieure à `Seuil_Reconnaissance` de cette position. Si tel est le cas, ce dernier nœud est reconnu et aucun nœud n'est créé. Toutefois, comme les positions relatives de tous les nœuds sont re-estimées à chaque intégration de nouvelle donnée idiothétique pour garantir la cohérence, certains nœuds peuvent se retrouver à une distance d'un autre nœud inférieure à ce seuil. Dans une telle situation, notre modèle regroupe ces nœuds en un nœud unique (cf. figure 5.13). Cette opération fusionne les données allothétiques des nœuds et associe au nouveau nœud toutes les arêtes qui étaient reliées à l'un des nœuds regroupés.

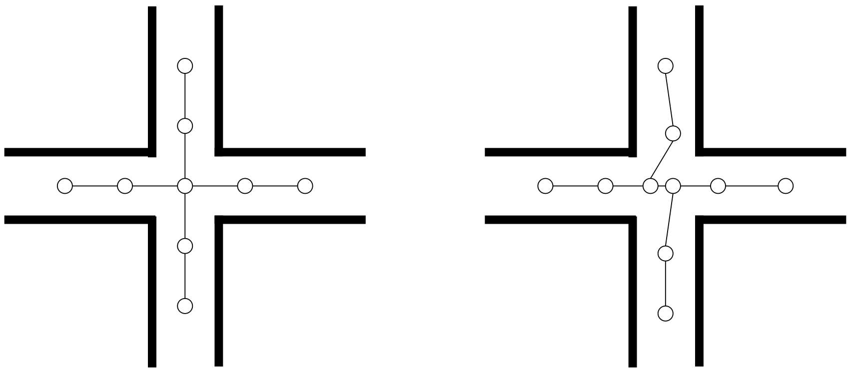


FIG. *5.14: Dans une carte topologique classique, limiter la connectivité de la carte entraîne une limitation des environnements qui peuvent être cartographiés. Dans cet exemple, limiter la connectivité à trois arêtes empêcherait la création de la carte de gauche. Dans notre modèle cependant, les nœuds peuvent avoir des positions quelconques et être très proches. Cela permet de cartographier toutes les situations, avec un nombre limité de connexions (partie droite de la figure).*

Les connexions inutiles sont également supprimées. En effet, comme nous l'avons mentionné dans la section 5.3.2, le nombre de connexions par nœuds joue un rôle important pour la complexité de l'estimation de l'activité des nœuds. Notre modèle limite donc le nombre de connexions par nœud à une valeur `Nombre_Connexions_Max`. Dans le cas où un nœud comporte plus de connexions que cette valeur, la connexion la plus ancienne est supprimée. Cette limitation de la connectivité de la carte, pourrait également limiter la complexité des environnements que le système serait capable de cartographier. Ce n'est pas de cas en pratique grâce à la densité de la carte. En effet, si un lieu devait être lié à plus de lieux que ne l'autorise cette limite, l'un des nœuds proches est simplement utilisé pour mémoriser les connexions supplémentaires (cf. figure 5.14).

5.5 Algorithme d'exploration

Les procédures d'exploration permettent de limiter les erreurs de localisation et de cartographier l'environnement de manière exhaustive. Pour limiter les erreurs de localisation, notre modèle limite la distance que peut parcourir le robot en terrain inconnu avant de revenir dans une zone cartographiée de l'environnement.

Pour estimer le temps parcouru en zone inconnue, notre modèle maintient le compte du nombre de nœuds créés consécutivement, sans aucune reconnaissance de nœud existant. Lorsque ce nombre dépasse un seuil `Seuil_Exploration`, le robot est arrêté et retourne vers une zone connue. Pour cela, la trajectoire relevée par l'odométrie depuis le dernier nœud reconnu est parcourue en sens inverse. Lorsque ce point est atteint, la procédure d'exploration normale est reprise.

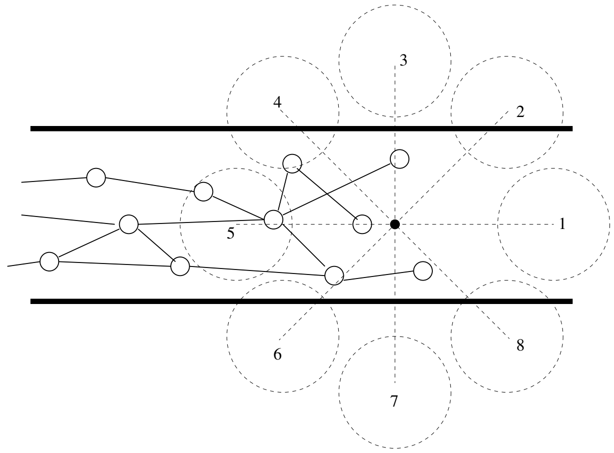


FIG. *5.15: La procédure d'exploration cherche, autour de la position du robot estimée par les données idiothétiques, la direction dans laquelle le moins de nœuds sont présents. Les directions dans lesquelles le robot perçoit des obstacles sont éliminées. L'évaluation du nombre de nœuds se fait dans une zone circulaire, à une distance fixée du robot, pour 8 directions réparties autour du robot. Dans cet exemple, le robot ira dans la direction* 1 *pour explorer son environnement.*

Lorsque le robot n'est pas guidé pour retourner vers une zone connue, notre modèle utilise une procédure d'exploration qui permet de parcourir l'environnement de manière exhaustive. Pour cela, la direction choisie à chaque instant est la direction dans laquelle aucun obstacle n'est détecté et qui est la moins bien cartographiée. La direction libre d'obstacles dans laquelle se trouvent le moins de nœuds est choisie (cf. figure 5.15). Si plusieurs directions comportent un nombre égal de nœuds, la direction pour laquelle les dates de dernière visite des nœuds sont les plus anciennes est choisie.

5.6 Planification

Nous avons utilisé une méthode de planification simple, afin de pouvoir utiliser les cartes créées pour rejoindre un but donné à partir de la localisation courante. Pour cela, nous avons calculé une politique en associant une direction à chacun des nœuds de la carte.

5.6.1 Calcul de la politique

Deux méthodes ont été utilisées pour calculer une telle politique. La première méthode utilise un algorithme classique de parcours de graphe en largeur d'abord, en partant du but (cf. figure 5.16). A chaque nœud connecté au but par n connexions, nous associons donc la direction de celui de ses voisins qui est connecté au but par $n-1$ connexions. Dans notre cas, cependant, l'incomplétude et la densité de la carte conduisent à des associations de direction qui peuvent être localement fausses (cf. figure 5.17). Cette technique ne donne donc pas de très bons résultats

```
- Pour chaque nœud N_i, index_{N_i} = 0
- index_But = 1
- index = 1
- (1) Tant que tous les nœuds ne sont pas parcourus
  - Mettre les nœuds N_i tels que index_{N_i} = index dans L
  - (2) Pour tous les nœuds N_j de L
    - (3) Pour tous les nœuds N_k directement connectés à N_j
      - Si index_{N_k} = 0, associer à N_k la direction de N_j
      - index_{N_k} = index + 1
    - fin (3)
  - Vider L
  - index = index + 1
  - fin (2)
- fin (1)
```

FIG. *5.16: Algorithme de planification utilisant le parcours de graphe au sein de la carte. Cet algorithme est un simple parcours du graphe en largeur d'abord en partant du but, associant à chaque nœud la direction de son voisin le plus proche du but.*

en pratique tant que la carte n'est pas suffisamment complète.

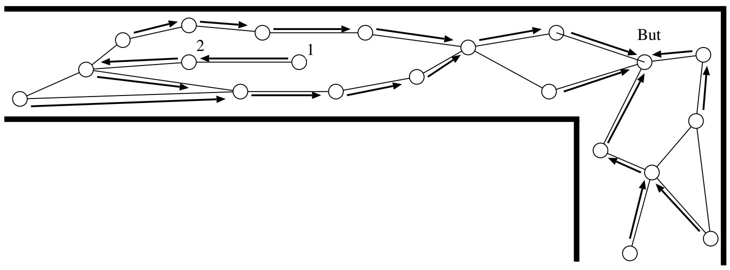


FIG. *5.17: Exemple de planification utilisant un algorithme de parcours de graphe. L'utilisation de cet algorithme dans une carte incomplète conduit à des direction qui sont localement fausses dans les zones où la connectivité est incomplète. Ainsi, pour rejoindre le but à partir des lieux 1 et 2, le robot choisira de partir vers la gauche du couloir, c'est-à-dire à l'opposé de la bonne direction. Cela rend cet algorithme très inefficace en cours de cartographie.*

Pour pouvoir utiliser la planification, même dans des cartes très partielles, nous avons donc utilisé une seconde méthode de planification. Cette méthode effectue un parcours de la carte similaire au précédent, mais en utilisant la position relative des nœuds indépendamment de l'existence ou non d'une connexion. Ainsi, un seuil `Seuil_Planification` est choisi et la boucle (3) de l'algorithme précédent (cf. figure 5.16) sera remplacée par : `Pour tous les nœuds` N_k `à une distance inférieure à Seuil_Planification de` N_j. Ainsi, une connexion manquante dans la carte ne conduira pas à des erreurs locales de planification (cf. figure 5.18).

Cependant, l'absence de connexions entre deux nœuds peut aussi indiquer l'impossibilité

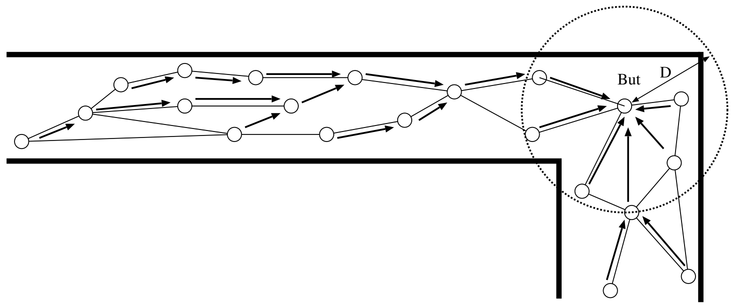


FIG. 5.18: *Exemple de planification utilisant la position relative des nœuds, plutôt qu'un parcours du graphe. Une direction est associée à tous les nœuds dont la distance à un nœud déjà planifié est inférieure à* `Seuil_Planification`*. L'utilisation de cet algorithme dans une carte incomplète permet d'avoir une planification correcte, et le rend donc utilisable à toutes les étapes de la cartographie (Comparer avec la figure 5.17).*

de passer d'un nœud à l'autre, par exemple à cause d'un obstacle, au lieu de résulter d'une cartographie incomplète. Ignorer les connexions et se baser uniquement sur la proximité spatiale des nœuds peut donc conduire à une planification incorrecte, qui pourrait tenter de faire traverser un obstacle au robot. Pour limiter ce risque, nous avons donc utilisé un algorithme qui utilise les deux approches simultanément. Ainsi, nous propageons la planification aux nœuds proches pour lesquels le chemin le plus direct suivant les connexions contient moins d'un certain nombre de connexions (cf. figure 5.19). Cette limitation permet donc de gérer un manque local de connexions dû à l'incomplétude de la carte tout en n'autorisant pas la propagation à des nœuds vers lesquels le manque de connexion indique la structure réelle de l'environnement (cf. figure 5.20).

Le choix du paramètre `Seuil_Planification` doit permettre la propagation entre les nœuds. Nous avons donc utilisé une valeur légèrement supérieure à la longueur moyenne des connexions `Seuil_Localisation` (cf. paragraphe 5.3). Pour le choix du paramètre P, nous avons choisi la valeur de deux fois le paramètre `Seuil_Exploration` réglant la distance que peut parcourir le robot en zone inconnue (cf. paragraphe 5.5).

5.6.2 Choix de la direction

Pour sélectionner la direction, à partir de la politique et de la distribution d'activité, nous avons utilisé une stratégie de vote [Simmons and Koenig, 1995, Cassandra et al., 1996] (cf. paragraphe 4.3.2). Pour cela, nous avons discrétisé l'ensemble des directions possibles pour le robot suivant un pas de 10 degrés. Pour chacune des directions discrétisées, nous calculons ensuite un poids correspondant à la somme des activités des nœuds auxquels cette direction est associée :

$$Poids(i) = \sum_{i<Dir(N_i)<i+10} Act(N_i)$$

où les i sont les directions discrétisées multiples de 10 degrés et $Dir(N_i)$ est la direction associée par la politique au nœud N_i.

- Pour chaque nœud N_i, $index_{N_i} = 0$
- $index_{But} = 1$
- $index = 1$
- (1) Tant que tous les nœuds ne sont pas parcourus
  - Mettre les nœuds N_i tels que $index_{N_i} = index$ dans L
  - (2) Pour tous les nœuds N_j de L
    - (3) Pour tous les nœuds N_k à une distance inférieure à Seuil_Planification de N_j et connectés par moins de P connexions.
      - Si $index_{N_k} = 0$, associer à N_k la direction de N_j
      - $index_{N_k} = index + 1$
    - fin (3)
  - Vider L
  - $index = index + 1$
  - fin (2)
- fin (1)

FIG. *5.19: Algorithme de planification utilisant la position des nœuds et la connectivité de la carte. Cet algorithme propage la direction de proche en proche depuis le but.*

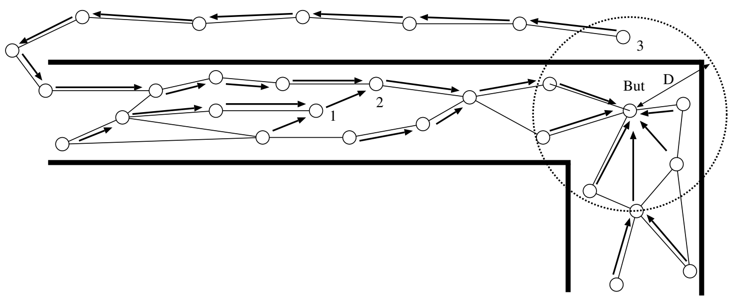


FIG. *5.20: Exemple de planification utilisant la position relative des nœuds, avec une limitation du nombre de connexions entre les nœuds. Cet algorithme propage la direction du but aux nœuds proches, à condition que le nombre de connexions joignant ces nœuds soit assez faible. Dans cet exemple, le nombre de connexions est limité à 10. Cela permet d'associer la direction du nœud 2 au nœud 1, car ces nœuds sont reliés par seulement 6 connexions. Mais cela empêche l'association directe de la direction du but au nœud 3, car le chemin reliant ces deux nœuds contient 14 connexions.*

La direction choisie est alors celle pour laquelle le poids est le plus élevé.

5.6.3 Réaction aux obstacles imprévus

L'algorithme de planification que nous avons décrit suppose que la carte représente de manière correcte l'environnement du robot. Par conséquent, en cas d'obstacle imprévu, l'exécution du plan peut se révéler problématique. Si les différences entre l'environnement réel et la carte sont faibles, les procédures locales d'évitement d'obstacles utilisées par le robot (cf. paragraphe 6.1.4) vont permettre de contourner l'obstacle imprévu et d'atteindre le but. Cependant, dans le cas où les différences entre la carte et l'environnement sont importantes[1], le plan risque de ne pas permettre d'atteindre le but. Pour éviter de telles situations, nous utilisons, lors d'un déplacement planifié, une procédure qui permet de vérifier que le plan est correctement exécuté. En cas de problème, un chemin alternatif rejoignant le but pourra alors être planifié.

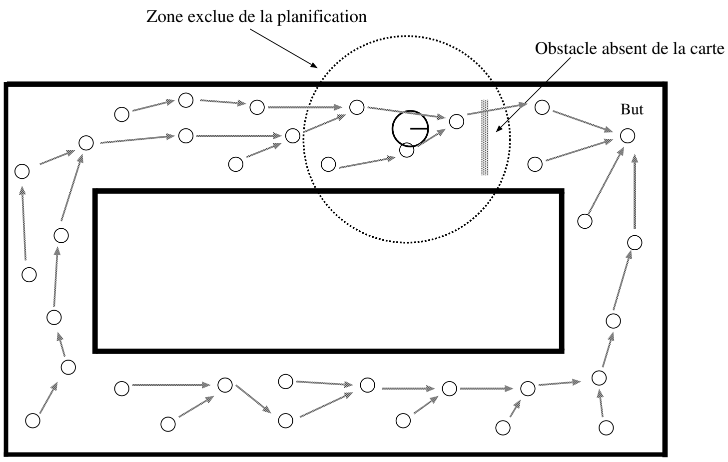
Afin de vérifier que le robot se dirige bien vers son but et ne reste pas bloqué par un obstacle imprévu, nous mesurons en permanence la distance parcourue au cours des derniers pas de temps. En cas de blocage par un obstacle du chemin planifié, les procédures d'évitement d'obstacle du robot le conduiront en effet à rester sur place et à ne plus s'approcher du but. Ainsi, si la distance parcourue au cours des derniers pas de temps est trop faible, le chemin actuel est considéré comme bloqué.

Dans un tel cas, afin de planifier un chemin alternatif menant au but, tous les nœuds de la carte proches de la position courante sont exclus du processus de planification. Le calcul de la politique est alors repris sur les nœuds restants (cf. figure 5.21). La taille de la zone dans laquelle la planification n'est pas faite correspond à la taille D de la zone utilisée pour la propagation de la planification (cf. paragraphe 5.19). Le résultat est une politique qui n'entraînera plus la traversée de la zone autour de la position courante du robot. Les nœuds qui ont été exclus de cette nouvelle planification n'ayant pas de direction associée, la politique n'est ensuite utilisée que si le nœud courant est bien associé à une direction planifiée. Tant que le robot se trouve dans la zone de l'environnement qui a été exclue du calcul de la politique, les procédures standard d'exploration sont utilisées. Ces procédures permettent de sortir rapidement de la zone non planifiée pour reprendre les déplacements en direction du but.

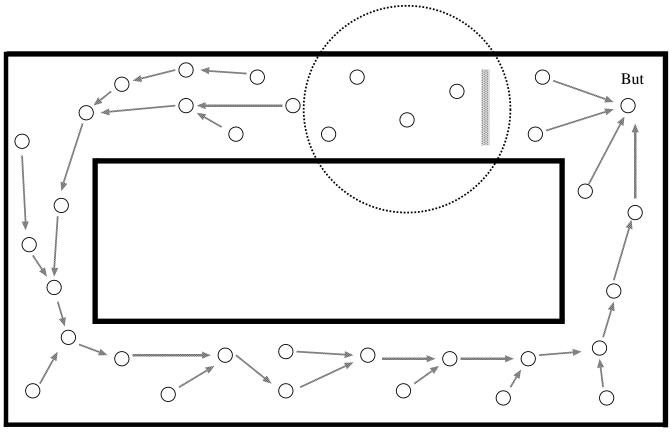
5.7 Choix de la direction des capteurs directionnels

Notre modèle intègre des procédures de perception active qui permettent de sélectionner une direction utile pour les capteurs directionnels. Ces procédures sont utilisées lorsque la précision de la localisation est jugée insuffisante, afin de fournir le plus d'informations possibles pour la localisation. Nous décrivons ici, par ordre de complexité croissante, les différentes procédures que nous avons testées. Leurs performances seront évaluées dans le chapitre suivant.

[1] Par exemple si un couloir de l'environnement est bloqué



Plan initial



Plan initial modifié

FIG. *5.21: Exemple de reprise de la planification en cas d'obstacle imprévu. Ici l'analyse des déplacements du robot a permis de se rendre compte que la progression vers le but était stoppée. Les nœuds de la zone entourant la position courante du robot sont alors marqués et exclus de la planification. Une nouvelle planification permet alors de créer un chemin évitant l'obstacle imprévu.*

La première stratégie, appelée `MaxDonnees` permet de sélectionner la direction pour laquelle les données sont les plus complètes au sein de la carte. Cette stratégie se base sur le fait que, lors de la construction de la carte, l'incomplétude de l'information apportée par les capteurs directionnels est un handicap à leur utilisation pour la comparaison des différentes positions. Ainsi, choisir une direction pour laquelle les données sont les plus complètes possibles permet potentiellement une meilleure localisation. Cette stratégie sélectionne une direction qui ne dépend pas de l'estimation courante de la position, ni du contenu des données allothétiques. Ainsi, elle n'est potentiellement intéressante que pour la construction de la carte, car elle ne prend en compte que l'incomplétude de l'information. La direction choisie est donc :

$$\alpha = argmax \sum_{N_i \in Carte} I_{Dir}(D_i, D_\alpha) \tag{5.14}$$

où D_α est un ensemble de données allothétiques couvrant le champ de vision de la caméra si elle est dirigée dans la direction α.

La seconde stratégie, appelée `MaxNbrPercep` sélectionne la direction dans laquelle le plus de données différentes sont disponibles. Cette stratégie utilise le fait qu'une direction pour laquelle les données sont uniformes n'apporte pas d'information, tandis qu'une plus grande variété de perceptions possibles permettra une meilleure estimation des probabilités relatives des positions. Cette stratégie n'utilise pas l'estimation courante de la position, mais se base sur les données allothétiques. Elle est donc potentiellement plus efficace que la stratégie précédente. Toutefois, elle ne prend pas en compte l'incomplétude de l'information, ce qui peut être un handicap en cours de construction de la carte.

Pour estimer le nombre de données différentes $NB_{Percep}(\alpha)$ dans la direction α, nous catégorisons les données allothétiques des différents nœuds de la carte dans cette direction. Cette catégorisation regroupe simplement les données allothétiques dont la similitude est supérieure à un certain seuil (0,9 dans notre implémentation). Le nombre de catégories résultantes est utilisé comme nombre de perceptions différentes. La direction sélectionnée est donc :

$$\alpha = argmaxNB_{Percep}(\alpha) \tag{5.15}$$

La troisième stratégie, appelée `MaxGainEntropie` permet de sélectionner la direction pour laquelle la diminution de l'entropie de la distribution de probabilités est maximale. Ainsi la direction choisie permet de réduire l'incertitude de l'estimation de la position. Cette méthode est potentiellement la plus efficace car elle prend en compte à la fois l'estimation courante de la position et les données allothétiques. Comme la méthode précédente, elle ne prend pas explicitement en compte l'incomplétude des données. Dans ce cas, toutefois, le fait que l'information soit partielle est intégré à la distribution de probabilités par la méthode de mise à jour que nous utilisons (cf. paragraphe 5.3.4). Ainsi, si cette méthode ne cherche pas les directions pour lesquelles plus de données sont disponibles, celles-ci seront pénalisées par le fait qu'elles conduisent à des distributions de probabilités plus uniformes et permettront donc une moins bonne estimation de la position. Cette méthode est le pendant de la méthode décrite dans la section 2.4 dans le cadre des systèmes utilisant des PDMPO.

L'entropie de l'activité de la carte est simplement définie par :

$$H(A) = -\sum_{N_i \in Carte} Act(N_i)\log(Act(N_i)) \tag{5.16}$$

Le gain d'entropie considéré pour une perception dans une direction donnée est le gain d'entropie moyen pour l'ensemble des perceptions possibles dans la direction en question :

$$G(\alpha) = \sum_{P_i \in \mathcal{P}_\alpha} P(P_i)H(A_{P_i}) - H(A) \tag{5.17}$$

où $\mathcal{P}_\alpha$ est l'ensemble des perceptions possibles dans la direction α, $P(P_i)$ est la probabilité de la perception P_i et $H(A_{P_i})$ l'entropie de la distribution d'activité après intégration de la perception P_i. Dans notre modèle, la probabilité $P(P_i)$ est simplement le rapport du nombre d'occurrences de P_i parmi les nœuds de la carte sur le nombre de nœuds de la carte.

Chapitre 6

Expérimentations

6.1 Plateforme expérimentale

Nous avons utilisé, pour nos expérimentations, un robot Pioneer 2, construit par la société ActivMedia [ActivMedia, 1999].

Fig. *6.1: Le robot Pioneer 2 utilisé pour nos expériences.*

Ce robot, d'un diamètre d'environ cinquante centimètres, utilise deux types de capteurs allothétiques :

- Une ceinture de sonars qui donne la distance des obstacles proches du robot. Les sonars sont au nombre de 16 et sont disposés autour du robot.
- Une caméra orientable donnant une image qui couvre un champ de vision d'environ 50 degrés.

Ces deux capteurs, qui sont disponibles sur un grand nombre de robots mobiles, relèvent respectivement de la catégorie des capteurs omnidirectionnels et de la catégorie des capteurs directionnels.

Les valeurs de tous les paramètres de notre modèle sont regroupées dans l'annexe A. Ces valeurs ont été gardées identiques pour toutes les expériences reportées dans ce mémoire.

6.1.1 Capteurs omnidirectionnels

Les données des 16 sonars sont utilisées pour estimer les valeurs de 8 capteurs virtuels qui donnent la distance des obstacles dans huit directions absolues régulièrement réparties autour du robot (cf. figure 6.2). Ces valeurs sont calculées en utilisant la direction fournie par l'odométrie du robot.

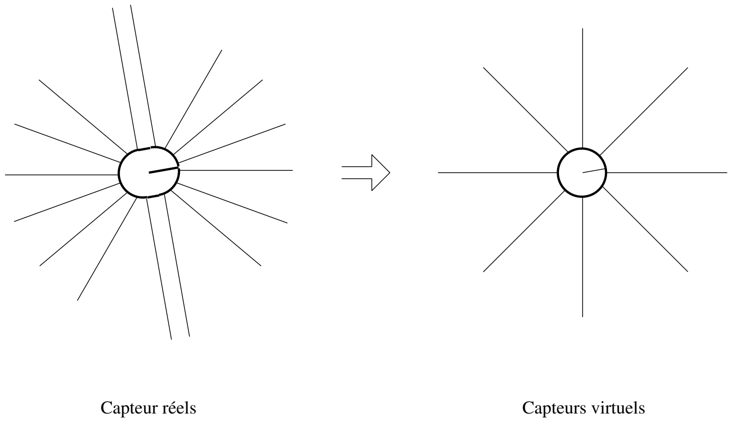


FIG. *6.2: Les valeurs des 16 sonars répartis irrégulièrement autour du robot sont utilisées pour calculer les valeurs de 8 capteurs virtuels de direction absolue fixe.*

Les valeurs S_k de ces huit capteurs sont mémorisées dans chaque nœud. Lorsqu'un nœud est reconnu comme position courante du robot, les valeurs mémorisées sont mises à jour en utilisant les nouvelles valeurs perçues par le robot. Chaque nouvelle valeur est la moyenne pondérée de la valeur mémorisée et de la nouvelle valeur :

$$S_k^M = \frac{K \times S_k^M + S_k^P}{1 + K} \tag{6.1}$$

où S_k^M est la valeur mémorisée pour le capteur k et S_k^P la valeur perçue à la position courante pour ce capteur. Le coefficient K est simplement le nombre de fois où le nœud a déjà été reconnu, ce qui permet aux valeurs mémorisées de converger vers une estimation des valeurs réelles des capteurs pour cette position, sous l'hypothèse que les données fournies par ces capteurs sont entachées d'un bruit blanc.

La mesure de similarité entre un lieu perçu et un lieu mémorisé est simplement la moyenne géométrique des écarts entre les valeurs perçues et mémorisées pour les différents capteurs :

$$P_{Omn}(S^M, S^P) = \sqrt[8]{\prod_{k=1}^{8} F(abs(S_k^M - S_k^P))} \tag{6.2}$$

où S_k^M et S_k^P sont les valeurs mémorisées et perçues par les capteurs virtuels dans la direction k et F une gaussienne de moyenne 0 et de variance σ_S donnée par $F(x) = \exp(x^2/\sigma_S^2)$. Ainsi, la valeur de $P_{Omn}(S^M, S^P)$ est maximale si $S^M = S^P$ et diminue quand la différence entre S^M et S^P augmente. La moyenne géométrique a été choisie plutôt que la moyenne arithmétique car elle permet de pénaliser plus fortement les cas où peu de capteurs ont des valeurs différentes. Ainsi, une forte similarité n'est obtenue que pour des données similaires dans toutes les directions.

6.1.2 Capteurs directionnels

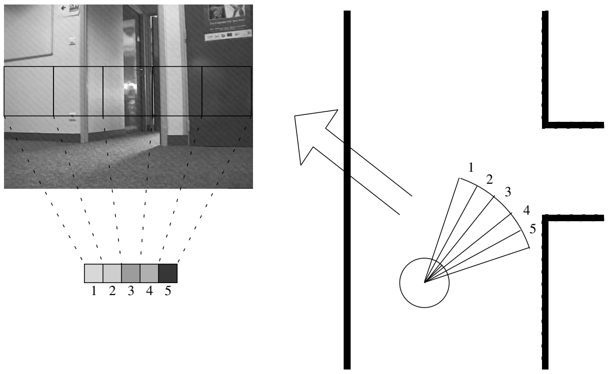


FIG. *6.3: En utilisant les images obtenues par la caméra du robot, le niveau de gris moyen de l'environnement est calculé pour des directions espacées de dix degrés. Cette valeur est la moyenne du niveau de gris des pixels inclus dans un carré dont le côté correspond à un angle de vision de dix degrés.*

Une caméra orientable est utilisée comme capteur directionnel sur notre robot. Les images obtenues par la caméra sont utilisées pour calculer les valeurs de capteurs virtuels qui mesurent le niveau de gris moyen de l'environnement dans une direction absolue fixée, sur un secteur angulaire de dix degrés (cf. figure 6.3). La totalité de l'environnement autour du robot est donc couverte par trente-six capteurs virtuels. Les valeurs de ces capteurs sont calculées en utilisant la direction fournie par l'odométrie du robot, ainsi que l'angle de la caméra par rapport au robot. La valeur du capteur virtuel dans une direction donnée est simplement la valeur moyenne du niveau de gris des pixels de l'image dans une zone de dix degrés autour de la direction correspondante. L'angle d'ouverture de la caméra permet d'estimer la valeur de cinq capteurs virtuels pour chaque image.

Pour qu'une telle valeur soit significative, nous avons utilisé le mode d'exposition manuel de la caméra, afin d'empêcher un réglage automatique de la sensibilité en fonction des conditions de lumière. En effet, un tel réglage entraîne une modification du niveau de gris des objets en fonction du contexte de l'image. Ainsi un objet sombre paraît plus clair s'il est disposé sur un fond sombre que s'il est disposé sur un fond clair. L'utilisation d'un mode d'exposition manuelle fixe permet de garder une constance dans le niveau de gris des objets, au prix d'une sur ou sous-exposition dans certaines parties de l'environnement. De telles variations d'exposition ne posent cependant aucun problème au système de cartographie car elles sont caractéristiques des positions de l'environnement concernées.

Les valeurs C_k des capteurs virtuels sont mémorisées pour chacun des trente-six capteurs dans chaque nœud. Le nombre de mesures N_k qui ont été effectuées dans chacune des directions autour du robot pour le lieu représenté est également mémorisé. La mise à jour des valeurs d'un nœud se fait de manière similaire à celle utilisée pour les capteurs omnidirectionnels, dans le cas où les données sont communes entre les valeurs perçues et les valeurs mémorisées :

$$C_k^M = \frac{N_k^M \times C_k^M + C_k^P}{1 + N_k^M} \; si \; N_k^M \neq 0 \; et \; N_k^P \neq 0 \tag{6.3}$$

où C_k^M est la valeur mémorisée pour le capteur k et C_k^P la valeur perçue à la position courante pour ce capteur. Le coefficient N_k^M de la somme pondérée est simplement le nombre de fois où des informations ont été mémorisées pour ce capteur, ce qui permet aux valeurs mémorisées de converger vers une estimation des valeurs réelles des capteurs pour cette position, sous l'hypothèse de corruption des valeurs par un bruit blanc.

Dans le cas où il n'y a pas de données enregistrées correspondant à une donnée perçue, celle-ci est simplement enregistrée :

$$C_k^M = C_k^P \; si \; N_k^M = 0 \; et \; N_k^P \neq 0 \tag{6.4}$$

La mesure de similarité entre un lieu perçu et un lieu mémorisé est la moyenne géométrique des écarts entre les valeurs perçues et mémorisées pour les capteurs ayant des données en commun. Le nombre d'éléments dans cette moyenne dépend donc du nombre de capteurs pour lesquels des données sont disponibles, à la fois pour les valeurs mesurées et pour les valeurs enregistrées :

$$P_{Dir}(C^M, C^P) = \sqrt[N_c]{\prod_{N_k^M \neq 0, N_k^P \neq 0} F(abs(C_k^M - C_k^P))} \tag{6.5}$$

où C_k^M et C_k^P sont les valeurs mémorisées et perçues par les capteurs virtuels dans la direction k, N_k^M et N_k^P sont les nombre de perceptions mémorisées et perçues par les capteurs virtuels dans la direction k et F une gaussienne de moyenne 0 et de variance σ_C donnée par $F(x) = \exp(x^2/\sigma_C^2)$.

La similarité est donc maximale et vaut 1 lorsque toute les données communes sont identiques, et décroît lorsque les différences entre les données augmentent. Dans le cas où il n'y a aucune donnée commune, la valeur de la similarité est fixée à -1.

6.1.3 Capteurs odométriques

L'odométrie du robot permet d'estimer la position et la direction du robot par intégration du déplacement des roues. Cette valeur se révèle relativement fiable en ce qui concerne les distances mais dérive très rapidement pour l'estimation de la direction. Cette estimation de la direction n'est pas suffisante pour notre modèle qui utilise la direction absolue des capteurs pour mémoriser leurs informations et estimer la similitude de deux perceptions. La direction du déplacement est également utilisée pour estimer la position du robot et se doit donc de présenter une erreur bornée pour assurer le bon fonctionnement du modèle.

L'estimation de la direction fournie par l'odométrie du robot a donc dû être corrigée. En simulation, nous avons simplement utilisé une boussole virtuelle qui fournit une mesure de la direction du robot entachée d'un bruit blanc de 5 degrés. Cette direction est donc utilisée, conjointement à la distance mesurée grâce à la rotation des roues dans le processus d'intégration, pour estimer le déplacement du robot entre deux mises à jour de la carte.

Lors des expériences sur le robot réel, toutefois, les valeurs mesurées par le compas magnétique du robot se sont révélées beaucoup trop bruitées et inutilisables pour corriger l'estimation de la direction. La mauvaise qualité de ces mesures est due aux champs magnétiques parasites provenant des appareils électriques qui sont nombreux dans un environnement de bureaux et à la structure métallique du batiment. Nous avons donc dû utiliser une méthode différente pour corriger l'estimation de la direction et avons choisi d'utiliser l'estimation de la direction fournie par l'odométrie tout en effectuant un recalage périodique de cette estimation. Lorsque ce recalage doit être effectué, le robot est arrêté et aligné manuellement avec sa direction de référence. L'estimation de la direction fournie par l'odométrie est alors remise à zéro et l'exploration de l'environnement reprend. L'effet de cette procédure est similaire à l'utilisation périodique de la mesure de la direction qui serait fournie par un capteur séparé tel que le compas magnétique. Dans toutes nos expériences, ce recalage a été effectué tous les cinquante pas de temps.

Les déplacements et la direction estimée en utilisant ce dispositif permettent d'avoir une estimation suffisamment fiable pour assurer le fonctionnement correct de notre modèle. Pour estimer la probabilité d'un déplacement de longueur r_{AB}, dans la direction θ_{AB}, sachant que l'odométrie du robot a mesuré un déplacement de longueur r_{od} dans la direction θ_{od}, nous utilisons le produit de gaussiennes suivant :

$$P_{Depl} = exp(-\frac{(\theta_{od} - \theta_{AB})^2}{K^2}) * exp(-\frac{(r_{od} - r_{AB})^2}{L^2}) \tag{6.6}$$

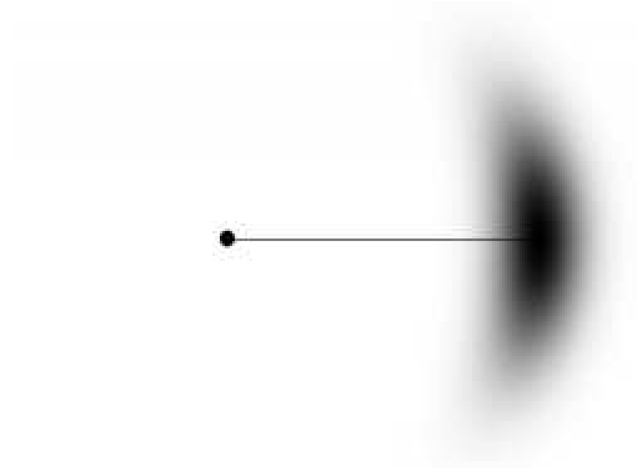


FIG. *6.4: La probabilité d'un déplacement en fonction de la valeur mesurée par l'odométrie est la valeur du produit de deux gaussiennes, fonctions de l'écart de direction et de distance entre les deux valeurs. Cette figure illustre la probabilité des déplacements si l'odométrie mesure un déplacement rectiligne. Le niveau de gris d'un point indique la probabilité d'un déplacement réel en ce point, depuis 0 pour le blanc, jusqu'à 1 pour le noir.*

Cette fonction vaut 1 lorsque od et AB sont égaux et tend vers 0 lorsque ces deux vecteurs divergent (cf. figure 6.4). Dans toutes nos expériences, nous avons utilisé $K = 30$ et $L = 200$.

6.1.4 Architecture logicielle

Nous avons utilisé pour notre robot une architecture mixte réactive/délibérative classique. Cette architecture comporte donc deux niveaux, l'un fonctionnant sur le robot et assurant les tâches de bas niveau, l'autre fonctionnant sur un second ordinateur réalisant les tâches de cartographie, de localisation et de planification (cf. figure 6.5).

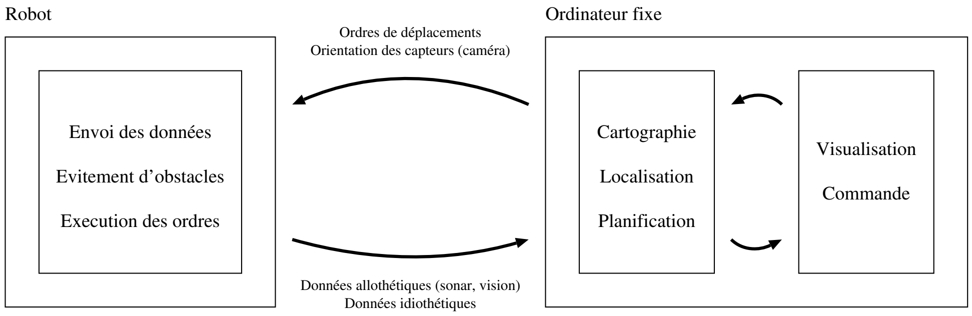


FIG. *6.5: L'architecture que nous avons utilisée pour notre robot est une architecture hybride classique. Un ensemble de comportements réactifs de bas niveau fonctionnant sur le robot permet d'assurer l'exécution des ordres provenant du système de navigation, tout en évitant les obstacles imprévus. Un second niveau fonctionnant sur un autre ordinateur effectue les opérations de cartographie, localisation et planification à partir des données reçues du bas niveau. Un second module fonctionnant sur ce même ordinateur permet de visualiser la carte et de choisir des buts pour le robot.*

Trois comportements de bas niveau, fonctionnant simultanément sur l'ordinateur à bord du robot, sont chargés d'éviter que le robot ne percute des obstacles ou ne s'y coince :

- **Avoid** est chargé d'éviter les obstacles détectés grâce aux sonars. Pour cela, il influe sur la direction et la vitesse du robot.
- **Center** est chargé de guider le robot préférentiellement sur l'axe des couloir afin de limiter les oscillations qui peuvent se produire lorsque le robot évite successivement chacun des côtés du couloir.
- **UnJam** est chargé de débloquer le robot si celui-ci détecte qu'il a percuté un obstacle grâce à ses capteurs de contact. Ce comportement fait alors reculer le robot avant de le diriger dans une direction sans obstacles.

Un comportement supplémentaire est chargé d'exécuter les ordres envoyés par le système de navigation. Ce comportement a trois fonctions :

- arrêter et démarrer le robot pour permettre l'acquisition de données par la caméra ;
- diriger le robot dans une direction spécifiée ;
- diriger la caméra dans une direction spécifiée.

Enfin un dernier comportement est chargé d'effectuer les traitements sur les données provenant des sonars, de la caméra et de l'odométrie, et d'envoyer ces données au système de navigation.

Le robot est relié par radio à un ordinateur qui se charge des tâches de cartographie, localisation et planification. Le programme qui réalise ces tâches reçoit les données provenant du robot, les organise au sein de la carte et envoie des commandes au robot, soit pour explorer l'environnement, soit pour rejoindre un but qui lui a été spécifié. Un second programme fournit une interface entre le système de cartographie et un utilisateur. Il permet de visualiser la carte et de désigner un but à atteindre.

6.2 Expériences en simulation

Nous avons testé notre modèle en utilisant un simulateur fourni avec notre robot [ActivMedia, 1999] qui, en ce qui concerne son aptitude à la navigation du moins, se comporte de manière identique au robot. Ce simulateur simule de façon réaliste l'odométrie, ainsi que les distances perçues grâce aux sonars. En particulier, il prend en compte les phénomènes de réflexion qui provoquent une surestimation de la distance des obstacles lorsque l'angle entre leur surface et l'axe des sonars est trop élevé.

De plus, nous avons développé un simulateur simplifié du système de vision qui permet de calculer le niveau de gris perçu par le robot en fonction de sa position et de la direction de la caméra. Dans toutes nos expériences en simulation, les murs de l'environnement sont uniformément noirs, tandis que les portes sont blanches.

L'utilisation d'un simulateur nous a permis d'avoir accès à la position réelle du robot simulé, et donc de pouvoir estimer la qualité de notre système de navigation. Nous avons premièrement utilisé notre système avec une caméra panoramique simulée, afin de tester la validité de l'approche et d'avoir une référence pour la comparaison des différentes stratégies de sélection de la direction de la caméra. Nous avons ensuite comparé les différentes méthodes de choix de direction des capteurs directionnels, afin de sélectionner la plus efficace dans notre implémentation sur le robot réel.

6.2.1 Utilisation d'une caméra panoramique

Cartographie

Nous avons utilisé pour ces expériences un environnement représentant une partie de nos locaux (cf. figure 6.6). Cette zone de l'environnement contient à la fois une zone ouverte et des portions de couloirs. Cette distinction est importante car, comme nous le verrons, les zones contenant des obstacles permettent une estimation efficace de la position grâce aux données des sonars.

L'utilisation d'une caméra panoramique dans un environnement de ce type permet une cartographie robuste et efficace. La figure 6.7 montre un exemple de carte créée dans cet envi-

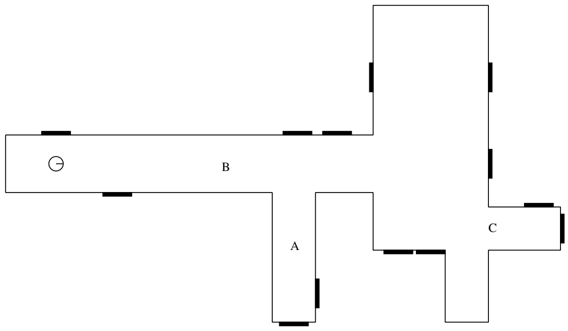


FIG. *6.6: L'environnement utilisé pour tester notre système de cartographie. Cet environnement correspond à une partie de nos locaux. Les rectangle noirs montrent les portes.*

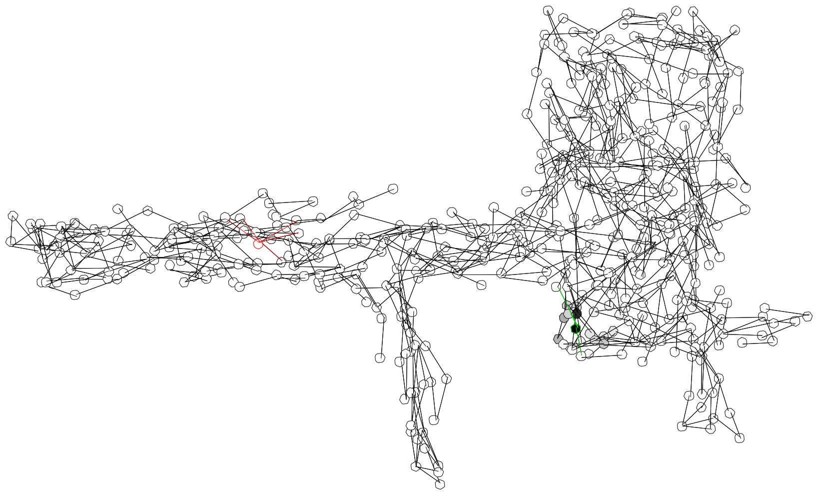


FIG. *6.7: Un exemple de carte créée dans l'environnement de la figure 6.6 en utilisant un système de vision panoramique.*

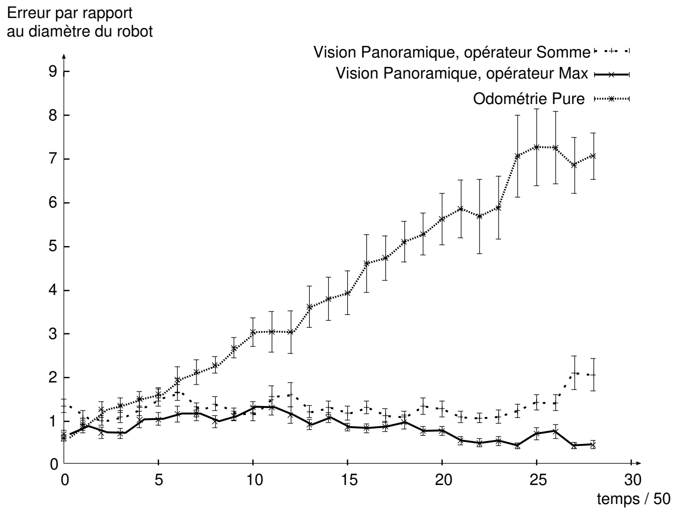


FIG. *6.8: Évolution de l'erreur de l'estimation de la position au cours de la cartographie de l'environnement de la figure 6.6 en utilisant l'opérateur* somme *et l'opérateur* max. *L'évolution de l'erreur de l'odométrie est données pour comparaison. Ces valeurs sont les moyennes sur dix expériences. Chaque unité de l'axe des abscisses correspond à cinquante pas de temps.*

ronnement. L'erreur dans l'estimation de la position du robot reste bornée aux alentours de la distance moyenne entre les nœuds, ce qui correspond à la taille de notre robot. Une telle précision est correcte pour une carte topologique et se révèle suffisante pour les tâches de navigation, comme nous le verrons dans la suite de ce chapitre. De plus, cette estimation est meilleure que l'estimation obtenue en utilisant l'odométrie de manière directe (cf. figure 6.8). Cette dernière estimation, en effet, montre une dégradation linéaire en fonction du temps qui provient du processus d'intégration d'une mesure bruitée du déplacement instantané du robot. Ce résultat montre que notre système permet effectivement de prendre en compte les données allothétiques de manière correcte pour recaler les données idiothétiques dans l'environnement et aboutir ainsi à une estimation entachée d'une erreur qui n'est plus cumulative.

Nous avons également comparé dans cette expérience les deux opérateurs que nous avons envisagés pour la mise à jour des activités en utilisant les données idiothétiques (voir paragraphe 5.3.2). Bien que les deux opérateurs permettent une estimation de la position suffisante pour une cartographie correcte, l'opérateur *somme* conduit de manière systématique à une plus faible précision de l'estimation de la position. Ce résultat semble pouvoir être attribué aux interactions entre cet opérateur et les particularités locales de la carte (voir paragraphe 5.3.2).

La figure 6.9 montre l'évolution du nombre de nœuds de la carte au cours du temps. Ce nombre cesse d'augmenter après environ deux mille pas de temps, lorsque l'environnement a été complètement exploré. Ce nombre se stabilise alors autour de quatre cents nœuds, même s'il subit encore de légères variations sous l'action des fonctions assurant le nettoyage de la carte. Cette stabilisation montre que la carte est alors complète et ne se dégrade pas, même à long terme, sous l'effet des procédures de nettoyage. Nous avons cependant pu constater que le nombre de nœuds

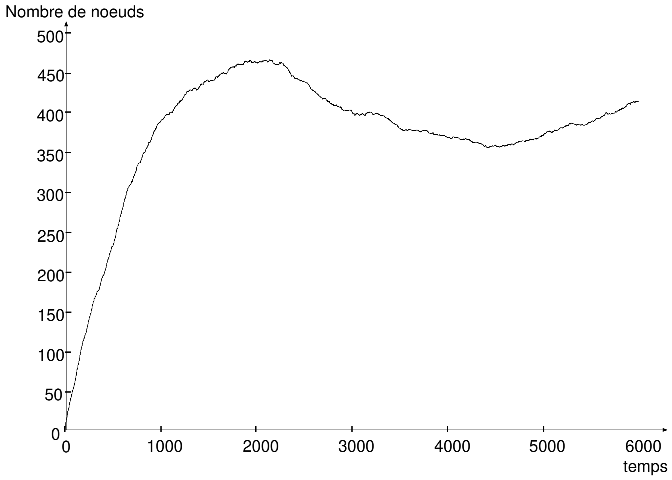


FIG. *6.9: Évolution du nombre de nœuds de la carte au cours du temps lors de la cartographie de l'environnement de la figure 6.6. Ces valeurs sont les moyennes sur dix expériences.*

total de la carte dépend des paramètres choisis pour ces fonctions de nettoyage. Ces paramètres n'ont toutefois pas d'influence sur la qualité de l'estimation de la position et de la cartographie. Nous avons donc gardé ces paramètres constants pour toutes les expériences reportées.

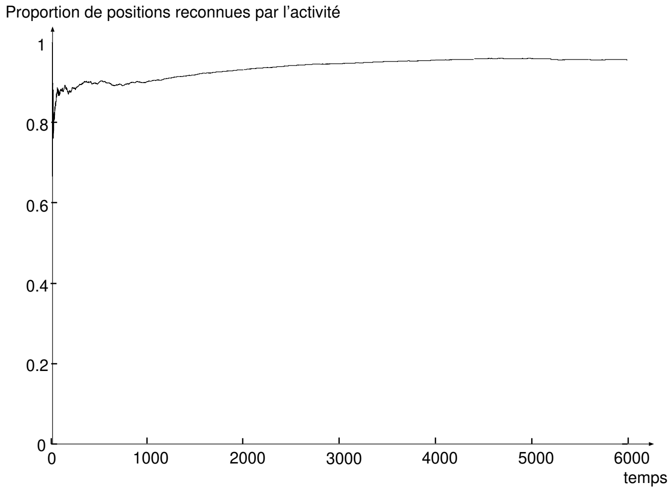


FIG. *6.10: Évolution de la proportion de nœuds reconnus grâce à l'activité de la carte par rapport au nombre total de nœuds reconnus au cours du temps lors de la cartographie de l'environnement de la figure 6.6. Ces valeurs sont les moyennes sur dix expériences.*

Notre modèle utilise une heuristique pour déterminer si la distribution des activités représente de manière correcte ou non la position du robot (cf. paragraphe 5.3.6). Dans le cas où cette représentation est correcte, le nœud courant est reconnu grâce à l'activité. Dans le cas contraire, il est reconnu ou créé à la position déterminée grâce à l'odométrie depuis le dernier pas de temps. La figure 6.10 montre que la proportion du nombre de nœuds reconnus en utilisant l'activité de la carte par rapport au nombre total de nœuds reconnus augmente rapidement et se stabilise autour de quatre-vingt quinze pour cent. Cette proportion est donc nettement en faveur de la reconnaissance par l'activité, ce qui montre que l'estimation de l'activité qui réalise la localisation globale est pertinente et que la carte construite permet rapidement une estimation correcte de la position par cette activité. La reconnaissance des nœuds par simple suivi de position, en utilisant leur position et l'odométrie, devient donc rapidement marginale.

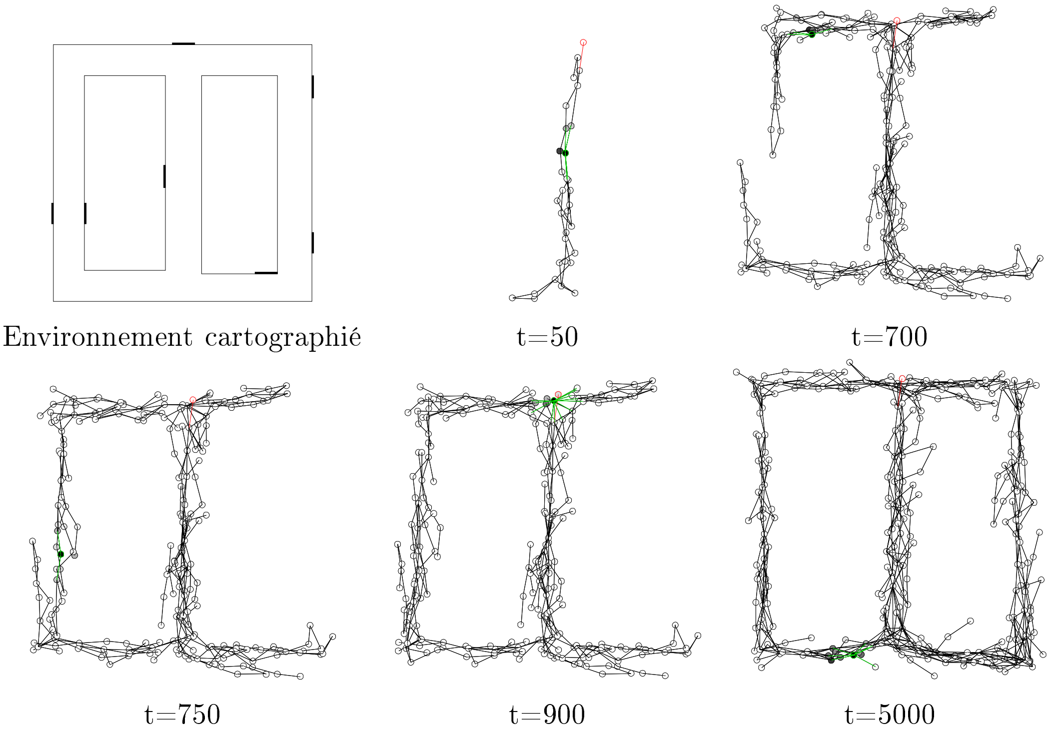


FIG. *6.11: Évolution de la carte lors de la cartographie d'un environnement cyclique (voir le texte pour les détails).*

La figure 6.11 montre l'utilité de la procédure permettant d'assurer la cohérence de la carte dans un environnement cyclique. Sous l'effet des erreurs de l'odométrie, l'environnement au temps $t = 700$ contient deux branches correspondant au couloir de gauche qui ne sont pas alignées et donc ne correspondent pas correctement à l'environnement. Au temps $t = 750$, le couloir correspondant a été complètement exploré et les deux branches ont été rejointes, sans toutefois fusionner complètement, ce qui conduit à une représentation légèrement fausse de l'environnement. Au temps $t = 900$, après d'autres passages dans ce couloir, les erreurs de cartographie ont été réduites grâce aux nouvelles données idiothétiques ajoutées à la carte. La direction du couloir de gauche est cependant légèrement biaisée. Le processus permet finalement d'obtenir une carte précise de l'environnement, avec des directions des couloirs correctement estimées (temps $t = 5000$).

La figure 6.12 montre les informations mémorisées dans un des nœuds de la carte correspondant à la zone qui entoure le point A (cf. figure 6.6). Nous pouvons constater que l'information visuelle mémorisée correspond correctement à l'environnement, avec des murs noirs dans la plu-

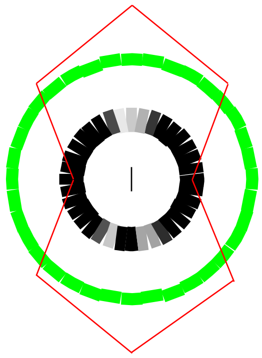


FIG. *6.12: Représentation des données mémorisées dans un nœud de la zone entourant le point A (cf. figure 6.6). La ligne brisée joint les points détectés au moyen des sonars. Nous pouvons constater l'effet des réflexions des sonars dont l'angle d'incidence sur les murs est de 45 degrés. Ces réflexions conduisent à une surestimation de la distance des obstacles. Le cercle intérieur montre le niveau de gris de l'environnement pour toutes les directions réparties autour du robot. L'épaisseur du cercle extérieur indique le nombre de perceptions effectuées dans chaque direction. Ici, cette épaisseur est constante car une caméra panoramique est utilisée.*

part des directions autour du robot, et les portes claires de part et d'autre du robot. Nous constatons, de plus, l'effet des réflexions des sonars sur les valeurs des distances mémorisées qui ne correspondent pas exactement à la configuration locale des obstacles dans l'environnement. Les valeurs des sonars ayant un angle d'incidence de quarante-cinq degrés sont en effet surestimées. Cette mauvaise estimation n'est toutefois pas problématique pour notre système de navigation, car elle est caractéristique de la position. Elle est donc intégrée dans la carte par le processus de cartographie.

Notre système de cartographie n'a pas pour but de construire une carte métrique précise de l'environnement. Toutefois, comme les positions relatives de chacun des nœuds et les valeurs mesurées par les sonars du robot à ces positions sont mémorisées, il est possible d'utiliser ces données pour avoir une idée de la position des obstacles. La figure 6.13 montre une telle reconstruction. Elle est obtenue en traçant un point à partir de la position de chacun des nœuds à la distance et dans la direction correspondant aux valeurs mémorisées des sonars. Le nuage de points obtenu reproduit de manière correcte la structure de l'environnement, ce qui prouve que les valeurs mémorisées dans la carte correspondent de manière relativement précise à l'environnement cartographié. La précision de la reconstruction obtenue est cependant faible et ne reproduit que les obstacles de grande taille comme les murs.

Enfin, la figure 6.14 montre des exemples de distribution d'activité des nœuds de la carte. Nous pouvons constater que l'information de position fournie par les données allothétiques peut être très ambiguë et correspondre de manière correcte à une proportion importante de nœuds de l'environnement. Toutefois, le processus d'intégration des données idiothétiques et allothétiques permet de calculer une distribution d'activité beaucoup plus précise qui correspond bien à la position réelle du robot.

FIG. *6.13: Exemple de reconstruction de la structure de l'environnement à partir des informations mémorisées dans la carte. Les valeurs des sonars mémorisées dans chaque nœud sont utilisées pour tracer un point à la distance correspondante du nœud. L'environnement possédant essentiellement des murs selon deux directions, seuls les sonars perpendiculaires à ces deux directions sont utilisés. Les sonars formant des angles de 45 degrés avec ces direction, et qui ne renvoient pas des valeurs correspondant à des obstacles réels (cf. figure 6.12), ne sont pas utilisés.*

Relocalisation

Pour démontrer les capacités de localisation globale de notre système de navigation, nous avons utilisé notre simulateur pour déplacer passivement le robot du point B au point C, ce qui implique que ce déplacement ne se reflète pas dans l'odométrie. Le système de localisation et de cartographie continue de fonctionner sans modifications. La figure 6.15 montre l'écart entre la position réelle du robot et la position estimée par notre système de localisation. Seuls quelques pas de temps sont nécessaires au système pour se recaler sur la position réelle du robot. Cela démontre la capacité de notre modèle à effectuer une localisation globale lorsque l'estimation initiale de la position est fausse. De plus, cette localisation globale est effectuée sans mettre le système de navigation dans un mode particulier, mais en continuant la cartographie. Cela démontre la capacité de notre système à effectuer de manière intégrée la cartographie et la localisation globale.

Déplacement vers un but

Enfin, nous avons utilisé notre système de planification pour diriger le robot vers le point C (cf. figure 6.6), depuis différentes positions de l'environnement. La figure 6.16 montre la direction associée à chacun des nœuds de la carte. La figure 6.17 montre les différentes trajectoires obtenues. Le but est atteint rapidement depuis tout point de l'environnement, sauf dans quelques cas où

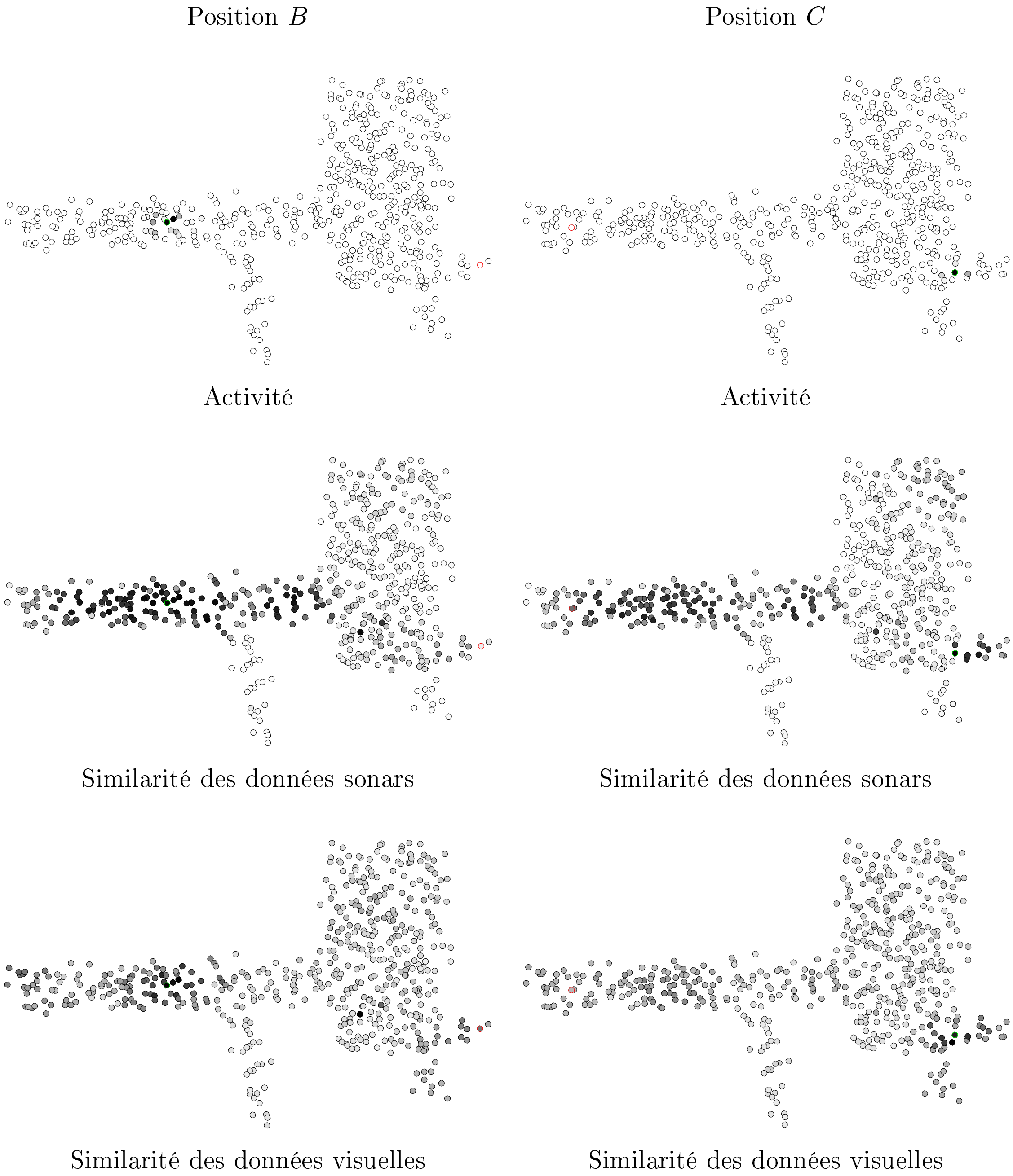


FIG. *6.14: Exemples d'activité au sein de la carte de la figure 6.7 pour les positions B et C (cf. figure 6.6). Les données provenant des sonars et de la caméra fournissent une estimation très imprécise de la position du robot. Toutefois, leur intégration permet d'avoir une estimation de bonne qualité, l'activité étant concentrée sur quelques nœuds seulement.*

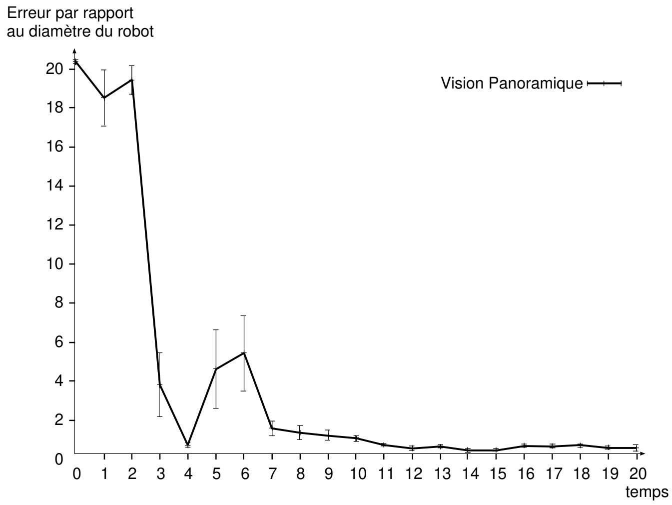


FIG. *6.15: Erreur d'estimation de la position après un déplacement passif du robot du point B au point C (cf. figure 6.6). Le système de localisation retrouve la position correcte du robot en environ sept pas de temps. Cette courbe représente la moyenne sur dix expériences.*

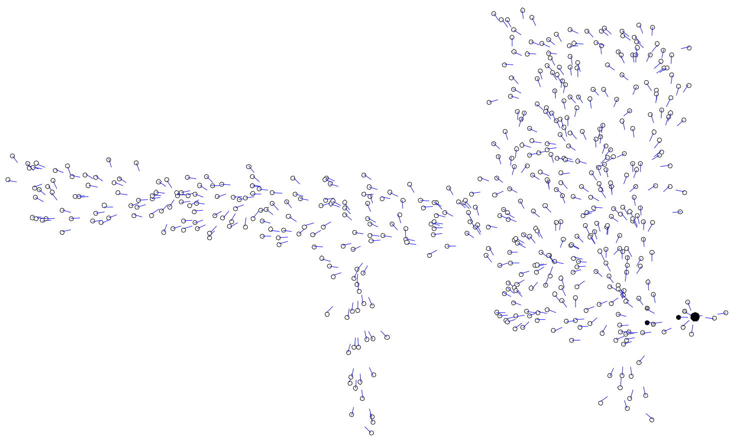


FIG. *6.16: Direction associée à chacun des nœuds de la carte pour atteindre le point C (cf. figure 6.6). Chaque point indique la position d'un nœud. Chaque ligne partant d'un point indique la direction à prendre au point correspondant pour rejoindre le but.*

une localisation imprécise entraîne de légers détours. Ces détours, toutefois, n'empêchent pas le robot d'atteindre le but car le système de localisation fonctionne en permanence. Durant la période de temps où ces erreurs surviennent, les informations acquises permettent de retrouver une estimation correcte de la position et d'atteindre finalement le but.

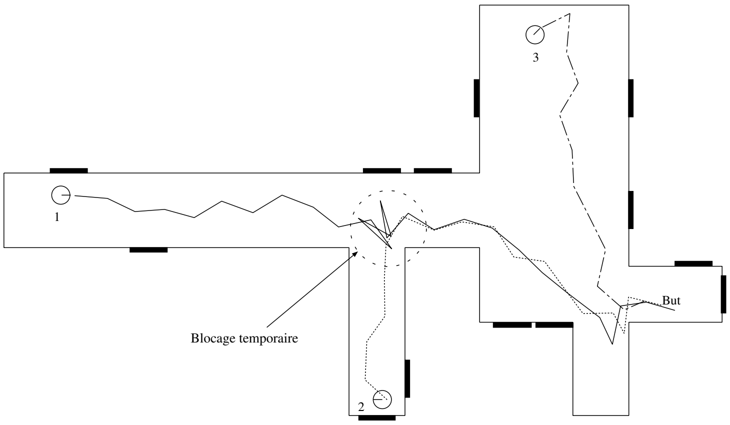


FIG. *6.17: Trois exemples de trajectoires effectuées par le robot en utilisant les directions représentées figure 6.16, pour trois points de départ différents. Les déplacements du robot sont quelquefois temporairement bloqués, mais finissent par reprendre pour atteindre le but.*

Nous avons également réalisé des tests de la procédure permettant de reprendre la planification en cas de problèmes d'exécution de la politique calculée. Pour cela, nous avons repris l'environnement de la figure 6.11 dans lequel nous avons ajouté un obstacle bloquant le passage du robot. La figure 6.18 montre un exemple de trajectoire obtenue dans de telles circonstances. La politique initiale guidait le robot sur un chemin passant par l'obstacle inconnu. Après avoir été bloqué pendant trois pas de temps face à cet obstacle, la procédure vérifiant l'exécution correcte du plan a été activée pour supprimer les nœuds de cette zone du processus de planification. Après une nouvelle planification, la politique obtenue entraîne le robot sur le chemin passant par le bas de l'environnement pour finalement atteindre le but. En d'autres termes, le robot a été capable d'effectuer un détour - une capacité qui est à la base de l'invention par Tolman [Tolman, 1948] de la métaphore de la *carte cognitive* appliquée au rat et à l'homme.

6.2.2 Utilisation d'une caméra directionnelle

Performance des méthodes de choix de la direction pour la localisation

Afin de comparer les différentes stratégies de sélection de la direction des capteurs directionnels décrites dans le paragraphe 5.7, nous avons effectué des expériences de relocalisation avec chacune de ces stratégies. Pour cela, nous avons utilisé une carte complète de l'environnement obtenue avec une caméra panoramique, afin de supprimer les effets de l'incomplétude de la carte sur l'estimation de la position. L'utilisation de l'environnement précédent pour effectuer ces comparaisons montre cependant que les différentes stratégies de sélection de la direction de la caméra conduisent à des résultats très similaires. Cela est dû au fait que, dans ce cas, l'information la

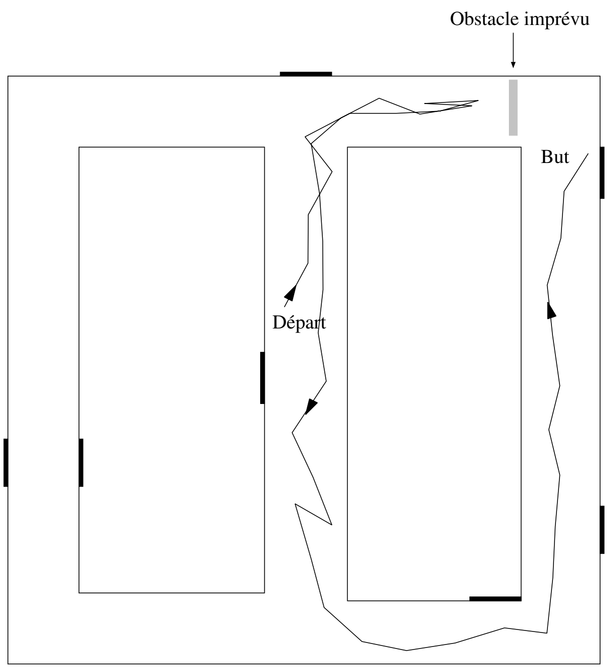


FIG. *6.18: Un exemple de reprise de la planification suite à un blocage du robot par un obstacle imprévu. La trajectoire initiale entre le point de départ et le but devait passer par la zone contenant l'obstacle inconnu. Après avoir détecté un problème dans l'exécution du plan, le système de planification permet de planifier une trajectoire évitant cette zone. Le robot utilise alors le chemin passant par le bas de l'environnement pour atteindre son but.*

plus importante pour l'estimation de la position semble être la conjonction des informations recueillies par les sonars et les données idiothétiques, plutôt que les informations recueillies par la vision. Pour limiter de tels effets, nous avons utilisé un environnement en forme d'arène dans lequel les informations des sonars sont inutilisables la plupart du temps (cf. figure 6.19).

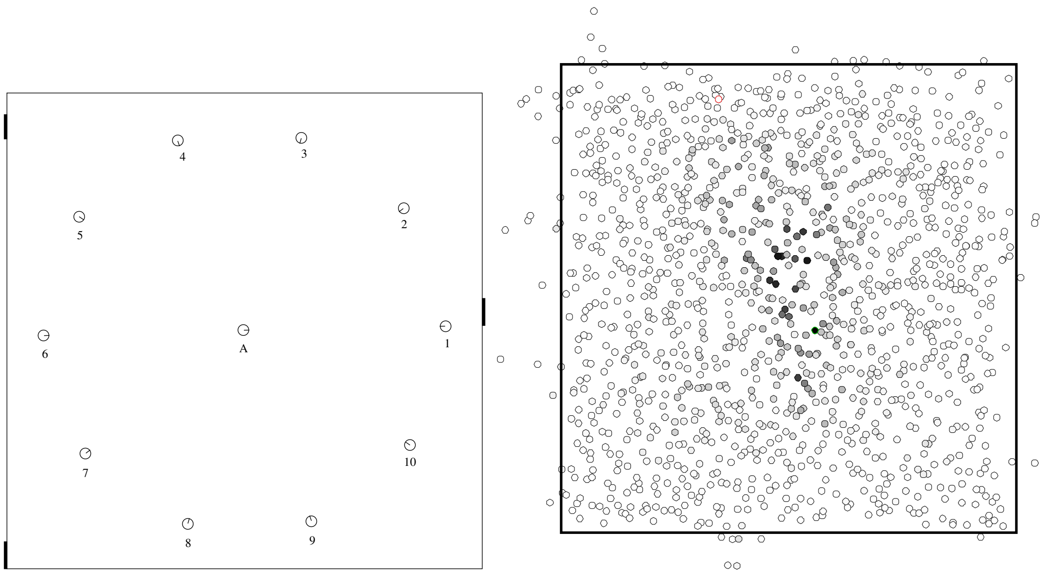


FIG. *6.19: Environnement utilisé pour comparer les différentes stratégies de sélection de la direction. Cet environnement permet de réduire la dépendance du système de cartographie aux données des sonars et de privilégier les données visuelles. La partie droite de la figure montre la carte utilisée ainsi que la distribution initiale de probabilité associée à la position A.*

Au sein de cet environnement, nous avons déplacé passivement le robot de la position A à chacune des position 1 à 10, en le dirigeant à chaque fois vers le centre de l'environnement. Chacune de ces positions se trouve à la même distance du point A, correspondant à 8 fois le diamètre du robot. Après ce déplacement, le robot est dirigé en ligne droite vers le centre afin qu'il ne rencontre pas les bords de l'arène qui lui permettraient de se localiser facilement. Cette méthode de déplacement permet également de supprimer l'effet de l'algorithme d'exploration afin de se concentrer sur l'effet de l'information apportée par la vision. Pour chacun de ces déplacements, nous avons mesuré la distance entre la position réelle et la position estimée du robot, en utilisant les différentes procédures de relocalisation.

La figure 6.20 montre des exemples d'activités et de similarités des nœuds de la carte avec les données allothétiques courantes obtenues avec la vision panoramique et la vision directionnelle. Ces similarités montrent bien les effets de l'incomplétude de l'information découlant de l'utilisation de la vision directionnelle. La similarité dans ce cas est en effet moins concentrée autour d'un point et conduit à une surestimation de la crédibilité d'une grande partie de l'environnement.

La figure 6.21 montre les écarts obtenus en utilisant une sélection aléatoire de la direction, par comparaison avec une vision panoramique, ainsi que les écarts obtenus en n'utilisant aucune information visuelle. Cette figure montre que, conformément à l'intuition, la précision atteinte par

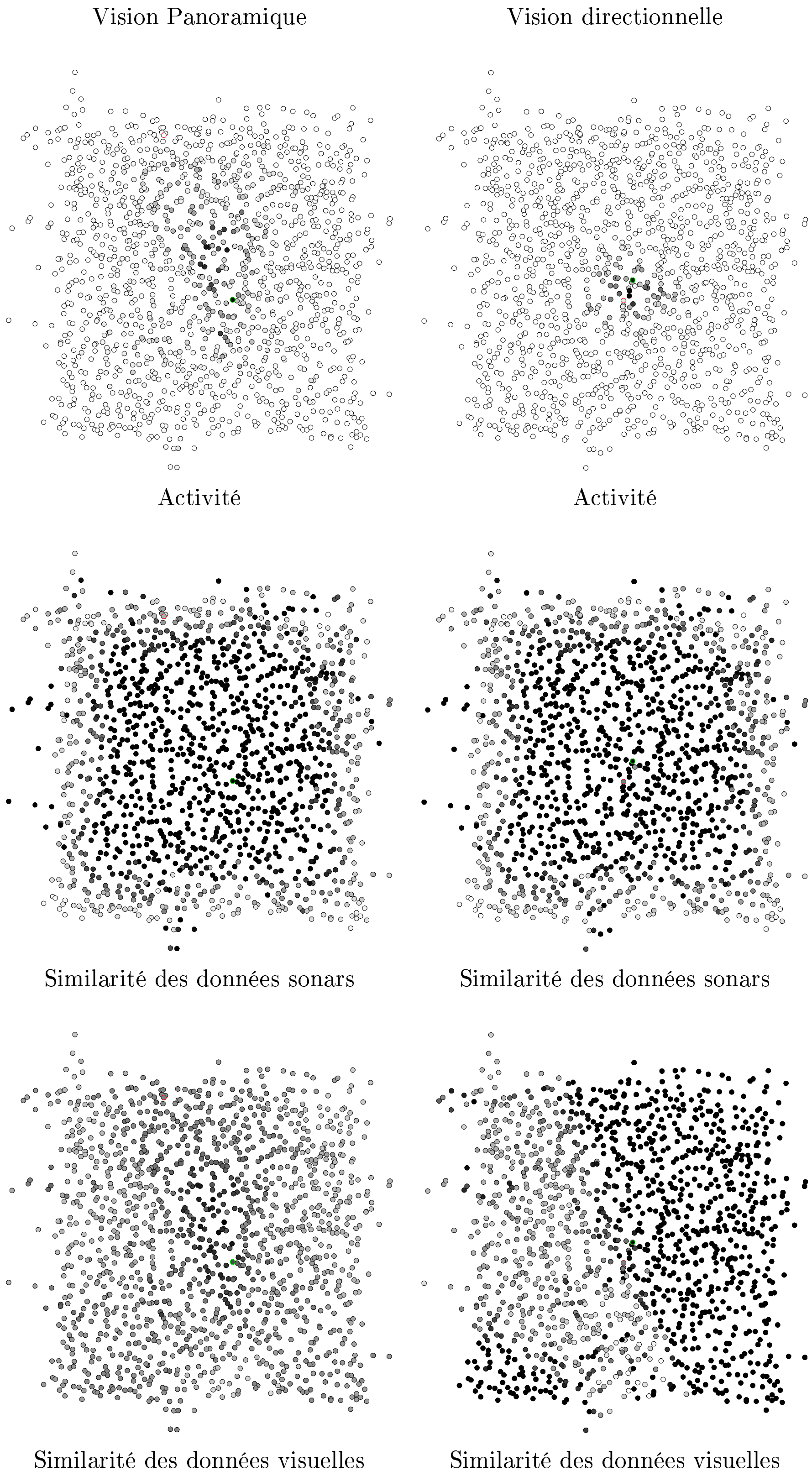


FIG. *6.20: Exemples d'activités au sein de la carte de la figure 6.19 en utilisant la vision panoramique et directionnelle. Les données provenant des sonars ne fournissent aucune information de position loin des murs de l'environnement. Les données visuelles fournissent une information relativement précise dans le cas panoramique, mais très partielle dans le cas directionnel.*

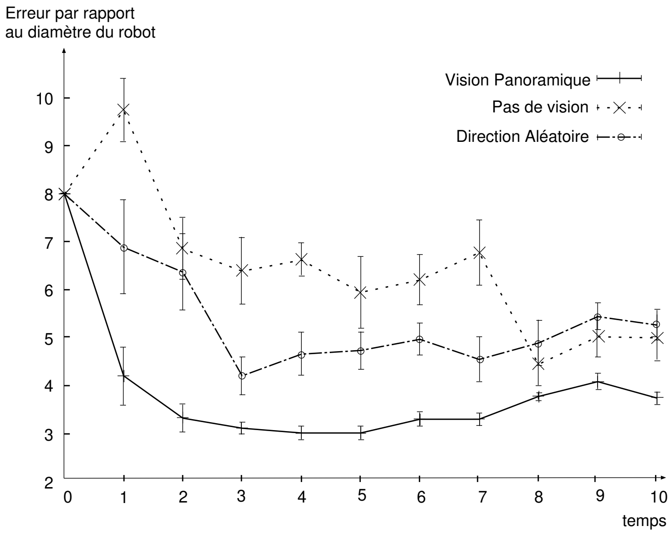


FIG. *6.21: Erreur de l'estimation de position pour la tâche de relocalisation avec différentes utilisations de la vision. Ces erreurs, données en rapport à la taille du robot, sont les moyennes sur dix expériences.*

la localisation avec une sélection aléatoire de la direction est moins bonne que lors de l'utilisation d'une vision panoramique, mais meilleure que la stratégie n'utilisant aucune information visuelle. La précision obtenue, même en utilisant une caméra panoramique, n'est toutefois pas très élevée. Cela démontre que les informations des sonars sont cruciales pour une localisation précise.

La relocalisation est effectuée très rapidement, en deux pas de temps, en utilisant la vision panoramique. L'utilisation d'une direction aléatoire permet une relocalisation moins précise en environ trois pas de temps. L'utilisation du système sans aucune vision permet également d'obtenir une estimation relativement correcte de la position après huit pas de temps. Cette performance peut paraître contre-intuitive puisque le système n'utilise ni information visuelle, ni information provenant de la perception d'obstacles pour estimer la position. Les informations allothétiques sont donc constantes pour toutes les positions visitées par le robot. Cela s'explique par le fait qu'il existe peu de positions dans la carte qui soient compatibles avec un déplacement en ligne droite durant huit pas de temps sans percevoir d'obstacles. L'utilisation des données idiothétiques et la constance des données sonars permet donc de retrouver ces positions.

La figure 6.22 compare les performances des stratégies `MaxNbrPerception` et `MaxGainEntropie` de sélection de la direction (voir paragraphe 5.7). La stratégie `MaxNbrPerception`, qui sélectionne la direction dans laquelle les perceptions sont les plus différentes, se comporte mieux que la stratégie aléatoire en obtenant une meilleure précision plus rapidement. Elle est moins performante que l'utilisation d'une caméra panoramique pendant les trois premiers pas de temps, mais est presque aussi efficace par la suite. La stratégie `MaxGainEntropie` est, pour sa part, aussi efficace que la caméra panoramique, et ce dès le premier pas de temps. L'incomplétude de l'information visuelle est donc complètement compensée par cette stratégie.

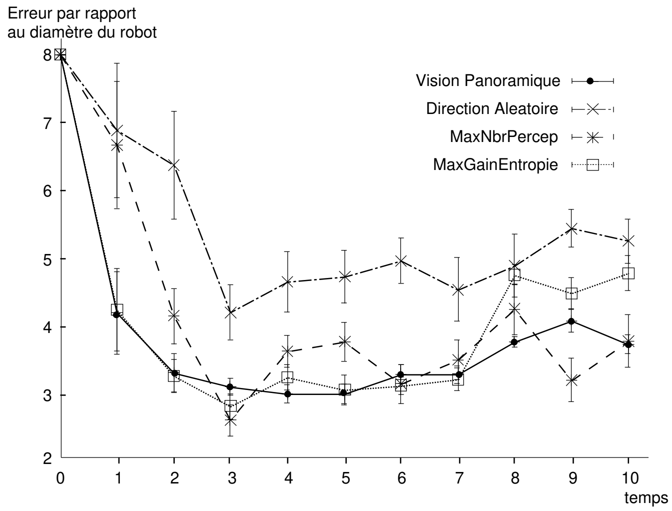


FIG. *6.22: Erreur de l'estimation de position pour la tâche de relocalisation avec différentes utilisations de la vision. Ces erreurs, données en rapport à la taille du robot, sont les moyennes sur dix expériences.*

Pour la seule tâche de localisation, les stratégies de sélection de la direction se comportent donc de manière intuitive. Ainsi, la recherche de la direction apportant le plus grand gain d'entropie est la plus efficace, suivie de la stratégie cherchant une direction dans laquelle les perceptions sont le plus variables, suivi enfin de la stratégie de sélection aléatoire. Toutefois, ces résultats se basent sur une carte complète de l'environnement et risquent de se comporter de manière moins satisfaisante pour la cartographie avec une caméra directionnelle.

Cartographie

Nous avons donc testé l'aptitude de notre modèle à cartographier un environnement avec une caméra directionnelle en utilisant ces différentes stratégies. Dans ce but, nous avons de nouveau utilisé l'environnement de la figure 6.6.

L'utilisation d'une stratégie sans vision ne permet pas la construction fiable de cartes de l'environnement. En effet, les erreurs de localisation qui surviennent entraînent des erreurs de topologie et conduisent à des cartes localement incorrectes (cf. figure 6.23). Nous avons donc utilisé pour les comparaisons une stratégie de choix aléatoire de la direction de la caméra. Cette stratégie permet de construire une carte correcte de l'environnement considéré, mais avec de nombreuses erreurs de localisation temporaires. Ces erreurs se traduisent par une moins bonne précision globale de l'estimation de la position et une plus grande variance lorsque nous effectuons la moyenne sur plusieurs essais (cf. figure 6.24).

La carte obtenue avec une caméra directionnelle est évidemment partielle. La figure 6.25 montre un exemple de données allothétiques mémorisées dans un nœud.

La figure 6.26 montre les erreurs de positionnement obtenues avec la stratégie `MaxGainEntropie`. Cette stratégie, qui s'était révélée la meilleure sur la tâche de relocalisation

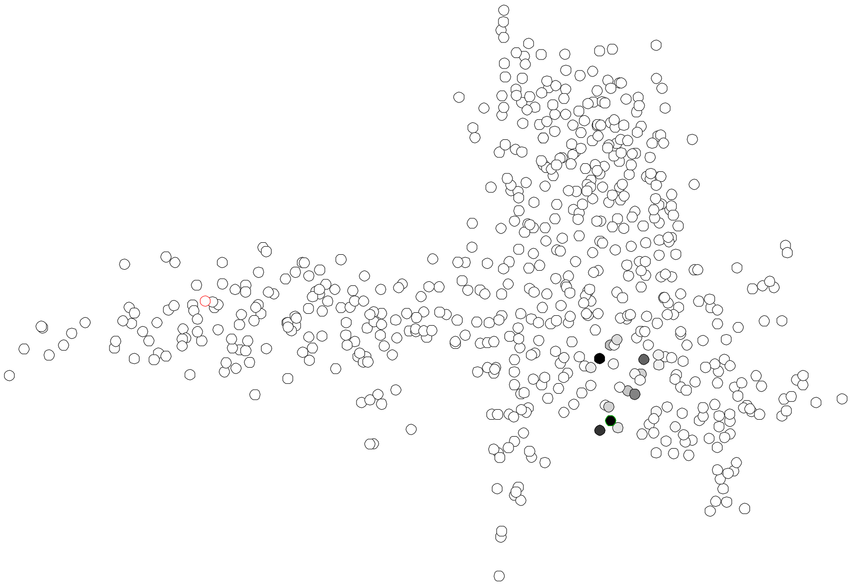


FIG. *6.23: Exemple de carte de l'environnement de la figure 6.6 créée sans utilisation de la vision. Certaines parties de cette carte, par exemple l'espace ouvert au centre de la carte, sont correctes, mais des erreurs de localisation ont conduit à la création de portions de carte ne correspondant pas à l'environnement.*

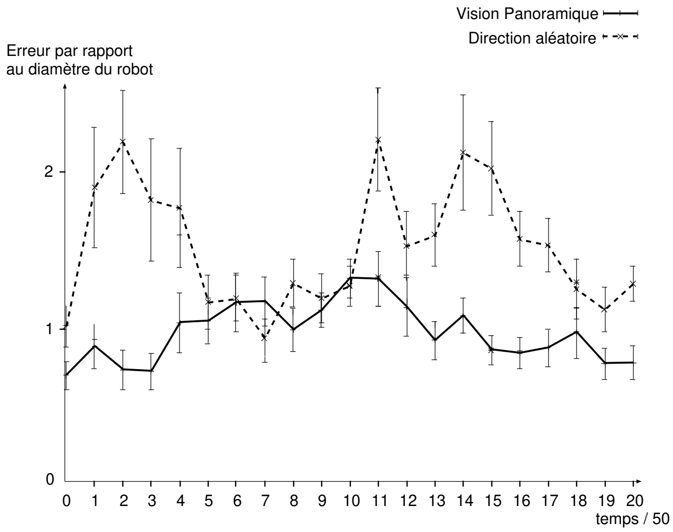


FIG. *6.24: Erreur de l'estimation de position pour la tâche de cartographie avec une vision panoramique et une sélection aléatoire de la direction. Ces erreurs sont les moyennes sur dix expériences. Chaque unité de l'axe des abscisses correspond à cinquante pas de temps.*

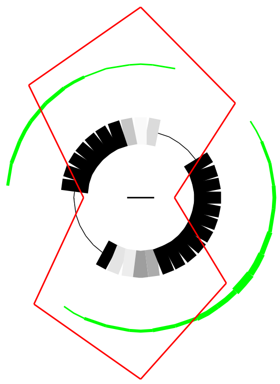


FIG. *6.25: Représentation des données mémorisées dans un nœud proche du point A (cf. figure 6.6) en utilisant une caméra directionnelle. La ligne brisée joint les points détectés au moyen des sonars. Le cercle intérieur montre le niveau de gris de l'environnement pour toutes les directions réparties autour du robot. Une ligne fine indique une absence de données. L'épaisseur du cercle extérieur indique le nombre de perceptions effectuées dans chaque direction.*

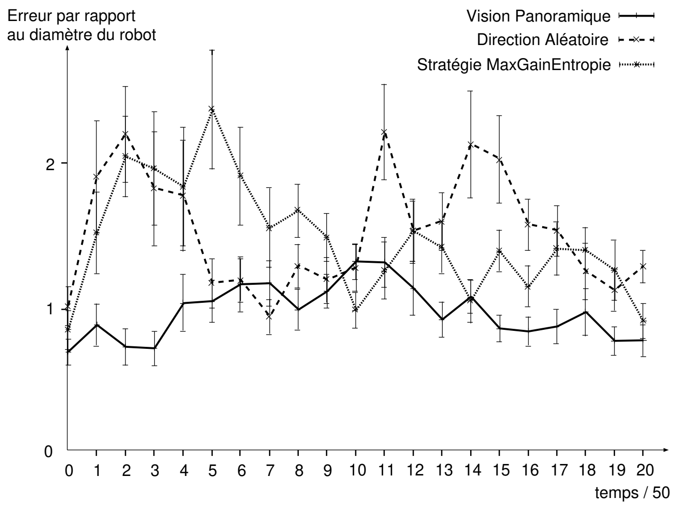


FIG. *6.26: Erreur de l'estimation de position pour la tâche de cartographie avec la stratégie de sélection de la direction* `MaxGainEntropie`*. Cette stratégie se comporte de manière similaire à la sélection aléatoire de la direction. Ces erreurs sont les moyennes sur dix expériences. Chaque unité de l'axe des abscisses correspond à cinquante pas de temps.*

avec une carte complète, produit ici des résultats similaires à la sélection aléatoire de la direction. Cette perte d'efficacité semble due essentiellement à l'incomplétude de la carte. En effet, cette stratégie sélectionne la direction dans laquelle la perception fournira une estimation de la position la moins ambiguë. Dans le cas d'une carte complète, les crédibilités relatives des différents nœuds de la carte seront correctement estimées pour toutes les directions. Cela permet une estimation correcte de l'apport des différentes directions à l'estimation de la position et conduit à une discrimination correcte des différents nœuds. Il en résulte donc une estimation correcte de la position. Dans le cas d'une carte incomplète, les crédibilités relatives des différents nœuds seront très mal estimées dans les directions pour lesquelles pas ou peu d'informations sont disponibles. Mais cette mauvaise estimation pourra conduire à une distribution d'activités moins ambiguë, et la direction correspondante pourra donc être sélectionnée par la stratégie. Cela conduit alors à une estimation moins ambiguë, mais fausse, de la position. Ce phénomène semble relativement courant et conduit donc à une estimation de la position de mauvaise qualité.

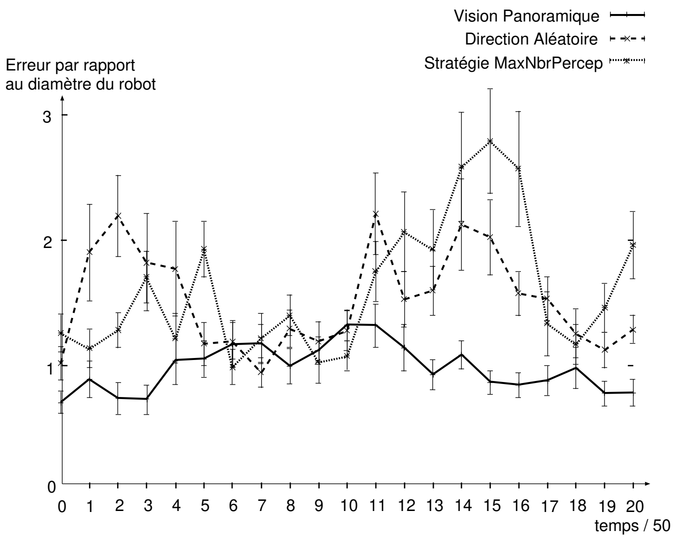


FIG. 6.27: Erreur de l'estimation de position pour la tâche de cartographie avec la stratégie de sélection de la direction `MaxNbrPerception`*. Cette stratégie se comporte, elle aussi, de manière similaire à la sélection aléatoire de la direction. Ces erreurs sont les moyennes sur dix expériences. Chaque unité de l'axe des abscisses correspond à cinquante pas de temps.*

La stratégie `MaxNbrPerception` donne, elle aussi, des résultats similaires à la sélection aléatoire de la direction sur la tâche de cartographie (cf. figure 6.27). Cette stratégie, qui était relativement efficace pour la seule tâche de relocalisation est donc, comme la stratégie précédente, fortement perturbée par une carte partielle de l'environnement. Cela s'explique encore par le fait que cette stratégie ne gère pas l'incomplétude des informations visuelles et peut donc sélectionner une direction qui fournit a priori beaucoup d'information pour la localisation, mais qui ne donne pas une estimation correcte des crédibilités des différents nœuds.

Pour pallier ces problèmes d'incomplétude de l'information, nous avons donc utilisé la stratégie `MaxDonnees` qui sélectionne simplement la direction dans laquelle le plus d'information est

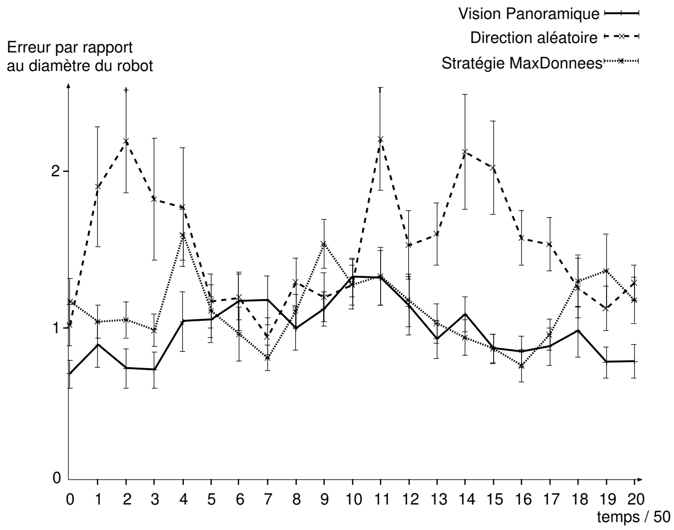


FIG. *6.28: Erreur de l'estimation de position pour la tâche de cartographie avec la stratégie de sélection de la direction* `MaxDonnees`. *Cette stratégie simple fournit de meilleurs résultats que les stratégies précédentes. Ces erreurs sont les moyennes sur dix expériences. Chaque unité de l'axe des abscisses correspond à cinquante pas de temps.*

disponible (cf. figure 6.28). Cette stratégie s'est révélée plus performante que les deux stratégies précédentes, avec des performances proches de l'utilisation de la vision panoramique. Cela montre que, dans notre modèle, la complétude de l'information, et donc la fiabilité de l'estimation de la crédibilité de chaque nœud, prime sur toute autre considération. Ce résultat indique que les valeurs que nous utilisons pour remplacer les informations manquantes (la moyenne des informations présentes, cf. paragraphe 5.3.4) introduisent un biais dans l'estimation de la position. La stratégie `MaxDonnees`, qui minimise le nombre d'informations manquantes, permet donc de réduire ce biais et permet ainsi une meilleure estimation de la position.

Pour la cartographie, il est donc préférable de privilégier cette complétude, plutôt que la discrimination que pourrait apporter une information visuelle. En pratique, comme nous souhaitons permettre à notre robot de cartographier son environnement et d'estimer globalement sa position simultanément et de façon la plus précise possible, nous utiliserons donc la stratégie `MaxDonnees`. Notons que cette stratégie est la plus simple de toutes celles mises en œuvre car elle ne tient pas compte des données particulières disponibles dans l'environnement, mais seulement de la complétude de ces données.

6.3 Expériences sur un robot réel

Nous avons également testé notre système de navigation sur un robot réel. Ne disposant pas d'un système de perception panoramique, nous avons utilisé une caméra directionnelle contrôlée par la stratégie qui s'est révélée la meilleure en simulation pour la cartographie. Nous avons donc utilisé la stratégie `MaxDonnees` .

6.3.1 Cartographie

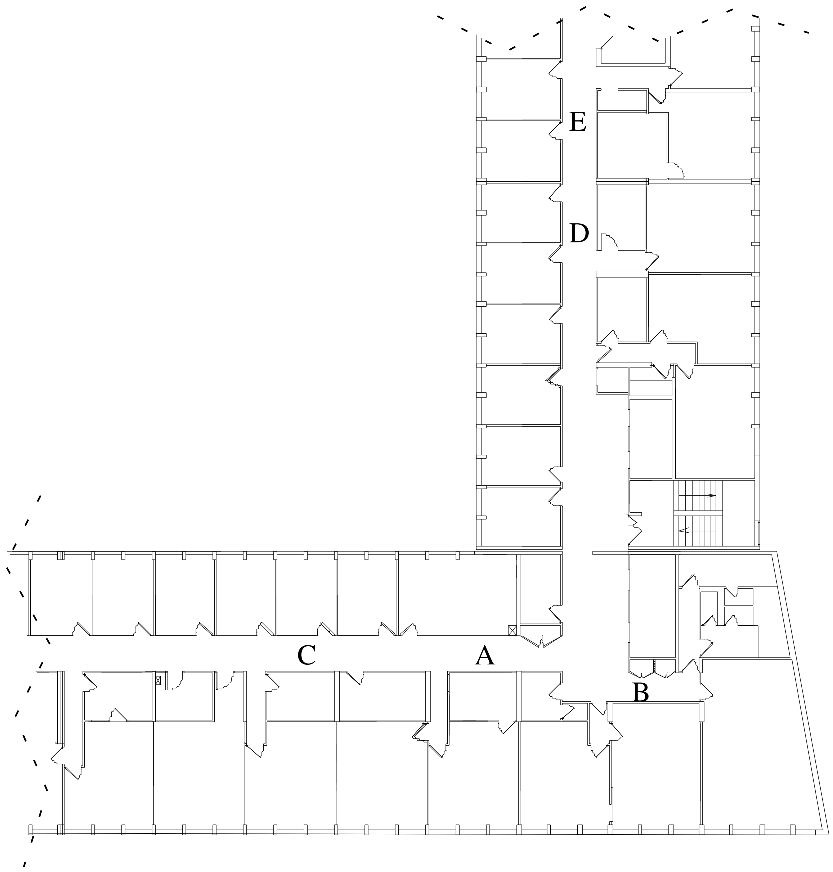


Fig. *6.29: Plan du bâtiment dans lequel notre robot a été testé. La zone cartographiée par le robot a été volontairement restreinte pour des problèmes de temps.*

Nous avons testé notre robot dans les couloirs de nos locaux, dans la zone représentée par le plan d'architecte de la figure 6.29. Pour des problèmes de temps, nous avons volontairement restreint la zone explorée par le robot en lui faisant faire un demi-tour chaque fois qu'il atteignait la limite de la zone représentée sur ce plan. En effet, le comportement du robot est relativement lent car il doit s'arrêter à chaque phase d'estimation de la position afin de permettre à la caméra d'atteindre la direction qui a été choisie par le système de navigation. Ainsi arrêter le robot, tourner la caméra puis redémarrer le robot prend environ sept à huit secondes à chaque étape. Les calculs requis par le système de navigation, pour leur part, ne demandent que quelques dixièmes de seconde à chaque étape.

La figure 6.30 montre l'une des cartes obtenue, en surimpression avec le plan d'architecte de l'environnement. Cette figure permet de constater que la structure globale de la carte est correcte et correspond bien à l'environnement. Certains nœuds ont cependant une position estimée qui se trouve dans des pièces que le robot n'a pas explorées. Ces nœuds sont le résultat d'erreurs de localisation temporaires qui ont conduit à des ajouts de nœuds erronés dans la carte. Ces erreurs n'ont toutefois pas de conséquences à long terme sur la cartographie car la position correcte finit par être retrouvée par le système de localisation globale.

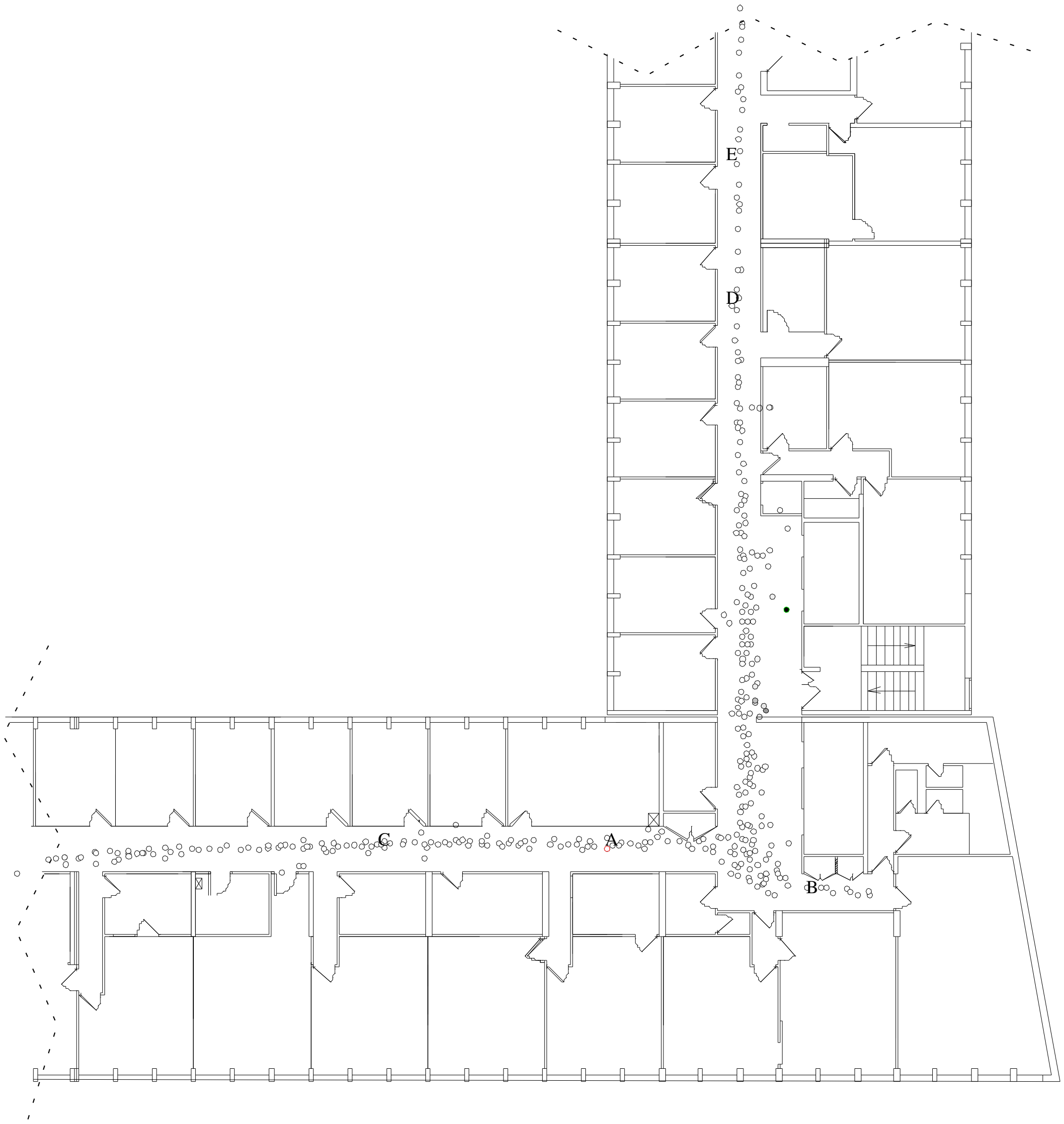


FIG. 6.30: *Un exemple de carte créée par notre robot dans l'environnement de la figure 6.29.*

Nous ne disposons pas sur le robot réel de référence absolue de position qui nous permette d'estimer la qualité de notre modèle. Nous pouvons toutefois comparer les positions estimées par notre modèle aux positions estimées par l'odométrie brute du robot (cf. figure 6.31). Le trajet du robot estimé par notre modèle semble plus proche du trajet réel que le trajet mesuré par l'odométrie. En effet ce dernier relève une trajectoire qui traverse les murs de l'environnement, alors que l'estimation de notre modèle reste dans les espaces ouverts et reproduit correctement la struc-

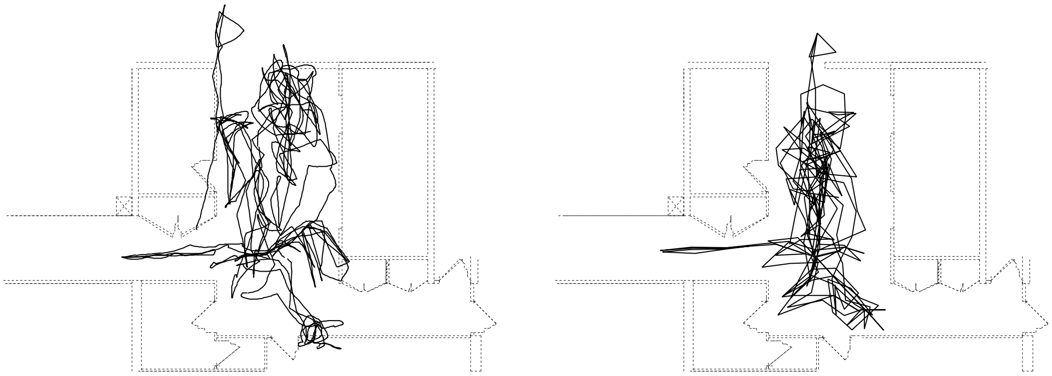


FIG. *6.31: Un exemple de trajectoire relevée par l'odométrie du robot et par notre système de navigation durant la cartographie. L'estimation fournie par notre système semble plus proche de la trajectoire réelle du robot.*

ture de l'environnement. Cela permet de vérifier que les données allothétiques sont correctement utilisées pour recaler l'estimation idiothétique de la position.

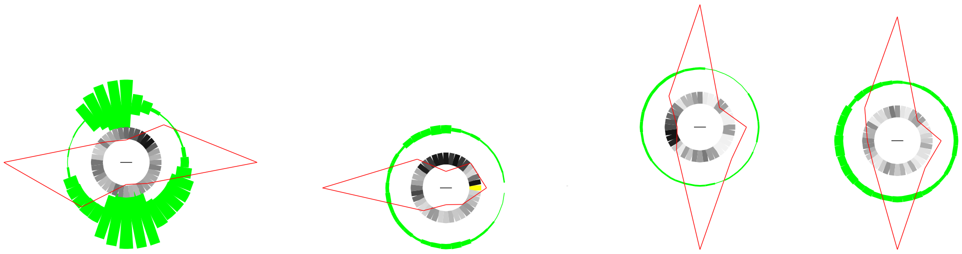


FIG. *6.32: Trois exemples de données allothétiques mémorisées au sein de la carte de la figure 6.30 pour les positions A, B, D et E de la figure 6.29.*

La figure 6.32 montre des exemples de données allothétiques enregistrées en différentes positions de l'environnement. Nous pouvons constater que les données provenant des sonars correspondent de manière relativement correcte à la structure locale de l'environnement, en reproduisant approximativement la forme et la direction des couloirs. La mauvaise estimation des distances pour les directions non perpendiculaires aux obstacles est confirmée, mais de manière moins prononcée qu'en simulation. Les données provenant de la caméra sont, pour leur part, beaucoup plus bruitées qu'en simulation. Il existe différentes raisons pour cela. La première est la plus grande variabilité des couleurs de l'environnement, d'une part au niveau des murs eux-mêmes - qui étaient simplement noir ou blanc en simulation - et, d'autre part, à cause des personnes se déplaçant dans le laboratoire - alors que l'environnement de simulation était statique. Il est toutefois possible, pour la plupart des nœuds de la carte, de reconnaître les portes sombres qui se détachent sur les murs clairs (cf. position A, B, et D de la figure 6.32).

Comme on pouvait s'y attendre a priori, la localisation est parfois moins précise sur le robot réel qu'en simulation, ce qui entraîne l'ajout de données allothétiques à des positions erronées et conduit à une uniformisation des données mémorisées. Ainsi, dans certaines zones qui ont été peu explorées ou pour lesquelles l'estimation de la position précise s'avère délicate, les nœuds peuvent contenir des données relativement uniformes qui ne correspondent pas à l'environnement (cf. position E de la figure 6.32).

La figure 6.33 montre des exemples de similarité entre des données perçues par le robot et les données enregistrées dans la carte. Les données provenant des sonars permettent de distinguer assez nettement les deux couloirs, ainsi que l'espace ouvert au centre de la carte, mais ne fournissent pas d'information de position très précise. L'information provenant de la vision est également relativement discriminante pour certaines positions, mais est en général très imprécise et apporte souvent peu d'information. C'est très clairement l'intégration de ces données et des données idiothétiques qui permet une estimation correcte de la position.

Enfin, la figure 6.34 montre une reconstruction de la forme des obstacles à partir des données mémorisées dans la carte. Cette reconstruction est relativement précise pour la plupart des régions de l'environnement. Certaines zones dans lesquelles la localisation et la cartographie ont été moins efficaces peuvent cependant être identifiées. Par exemple, les zones 1 et 2 correspondent à des zones dans lesquelles le bruit des valeurs mémorisées pour les sonars est particulièrement élevé. Dans la zone 1, cela est dû au début d'un couloir, au bord de la zone ouverte (entre les zones 1 et 2) dans laquelle l'estimation de la position est difficile. Ainsi, en arrivant dans la zone 1 l'estimation de la position est peu précise et conduit à un ajout de données erronées. La zone 2 correspond à une zone de l'environnement dans laquelle un mur isolé au centre de l'espace ouvert produit un fort bruit sur les valeurs des sonars. Ce bruit conduit à une mauvaise représentation de la zone entourant ce mur.

6.3.2 Relocalisation

Nous avons également testé la capacité de notre modèle à retrouver la position du robot après un déplacement passif d'un point de l'environnement à un autre.

Nous avons effectué une première expérience en déplaçant le robot du point A au point B (cf. figure 6.29). Nous avons effectué dix essais en utilisant le système de navigation durant dix-huit pas de temps pour estimer la position du robot. Sur ces dix essais, neuf ont permis de retrouver la position correcte du robot après environ dix pas de temps. L'évolution moyenne de l'erreur lors de ces neuf essais est représentée sur la figure 6.35. On constate sur cette figure que cette erreur augmente systématiquement entre les pas de temps 2 et 7. Cela est dû à l'existence d'une position très similaire au point B près du point C. La position estimée du robot passe ainsi de A aux alentours de C entre les pas de temps 2 et 7, avant d'être correctement estimée près de B après dix pas de temps. Lors du dixième essai qui n'a pas été pris en compte, la position estimée du robot est restée près de la position C pendant les dix-huit pas de temps.

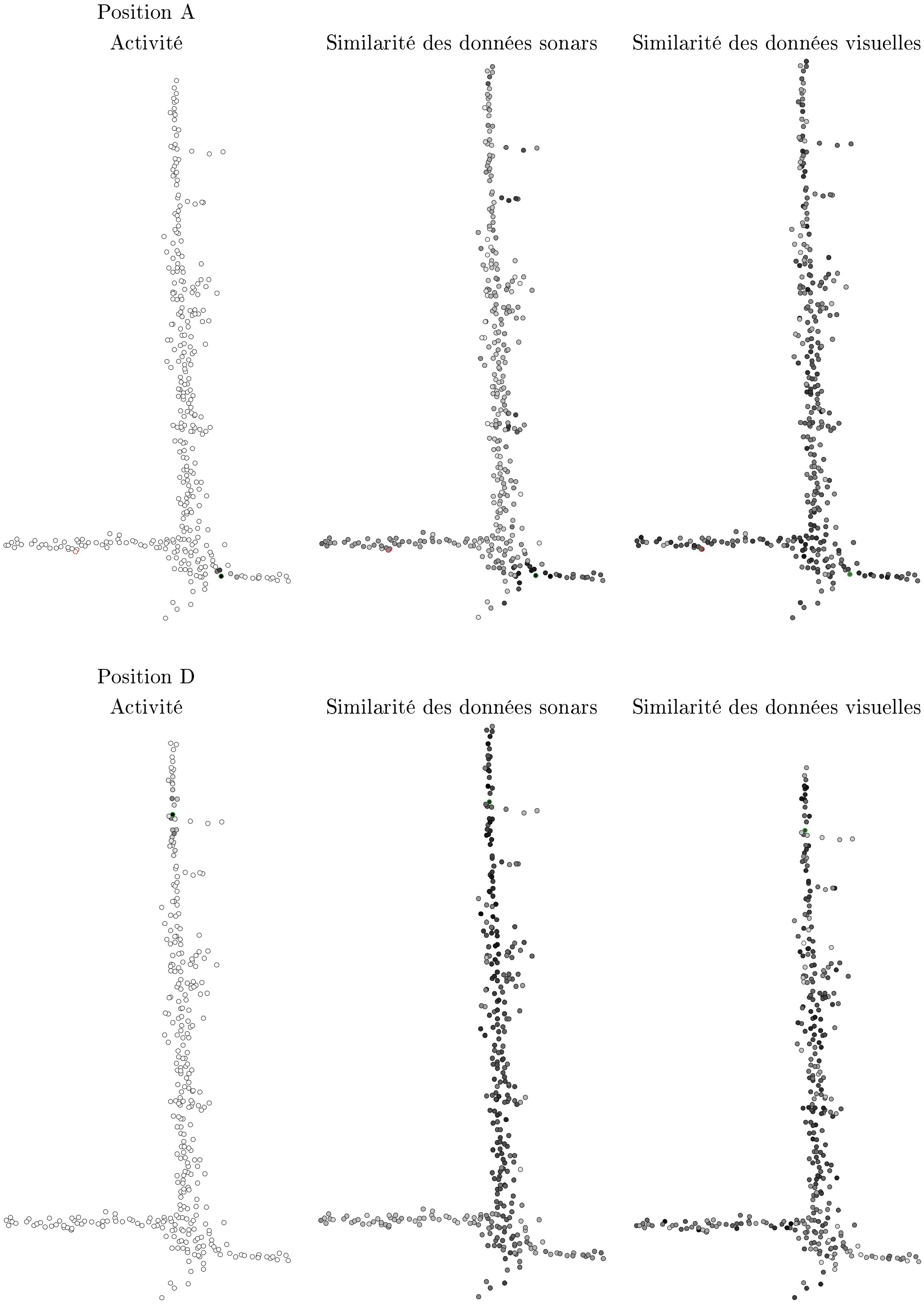


FIG. *6.33: Exemples d'activités au sein d'une partie de la carte de la figure 6.30. Les données provenant des sonars permettent de distinguer les deux couloirs de l'environnement, tandis que les données provenant de la vision sont beaucoup plus bruitées. Cependant, l'intégration des deux types de données fournit une estimation relativement précise de la position.*

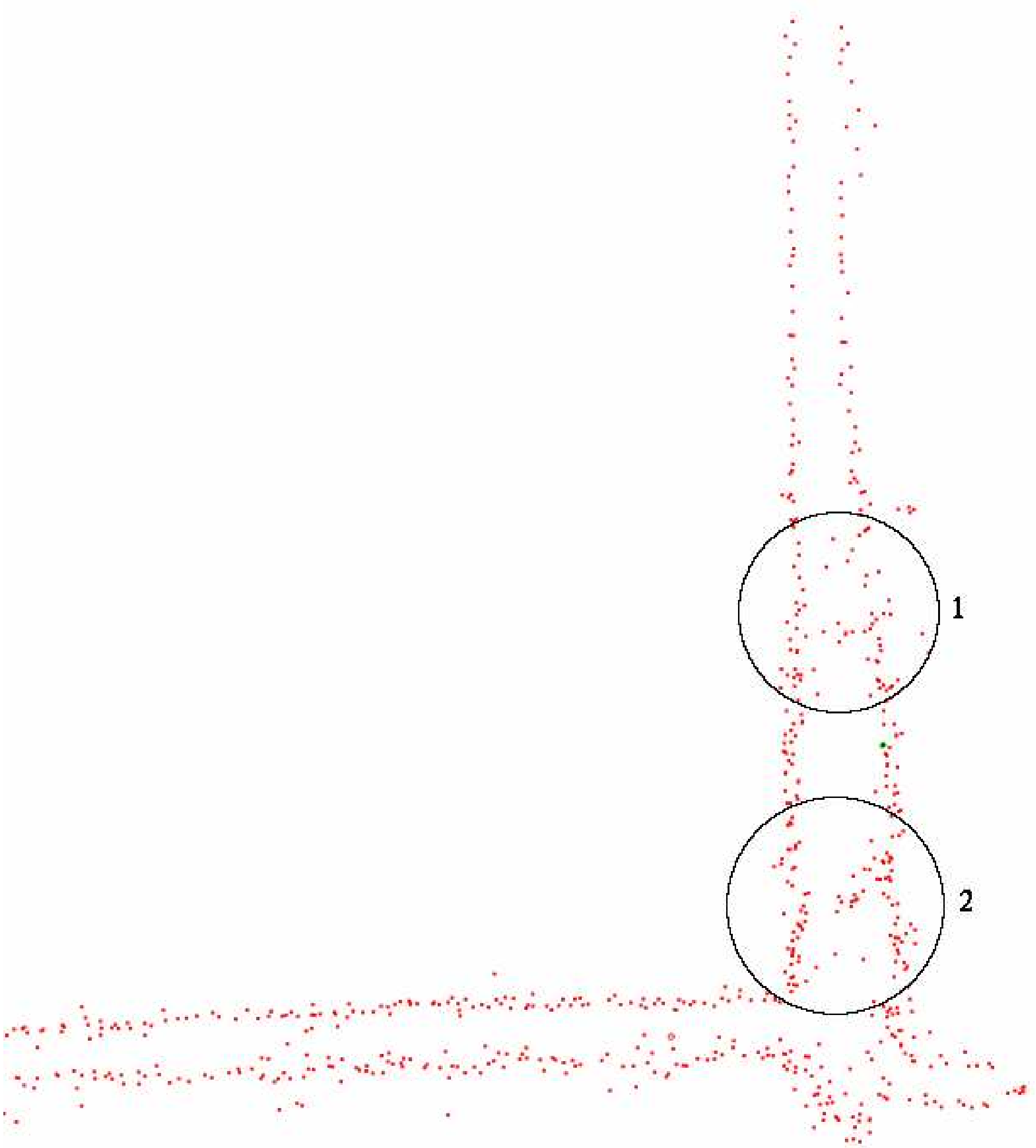


FIG. 6.34: *Reconstruction des obstacles de l'environnement à partir des données mémorisées dans la carte de la figure 6.30.*

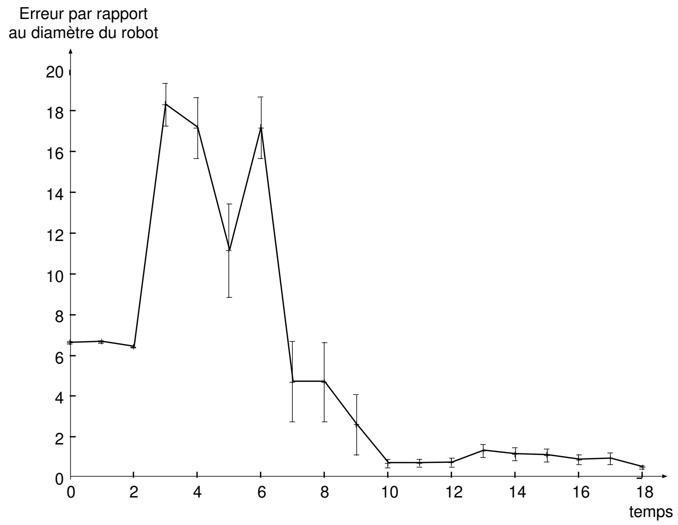


FIG. *6.35: Évolution de l'erreur de positionnement après que le robot ait été déplacé passivement du point A au point B (cf. figure 6.29). La position de référence utilisée est mesurée grâce à l'odométrie à partir de la position dans laquelle le robot a été passivement déplacé. Ces résultats sont les moyennes sur 9 expériences.*

Toutefois, l'efficacité du processus de relocalisation dépend des positions concernées. Il est par exemple peu efficace si la position initiale et la position finale du déplacement se trouvent dans le même couloir. Dans ce cas, en effet, les informations apportées par les sonars ne varient pas et seule la vision permet de retrouver la position correcte. La raison principale de la difficulté de relocalisation dans ce cas est la faiblesse de l'information apportée par la vision, qui n'est pas suffisamment discriminante pour retrouver la position du robot. La relocalisation est rapidement effectuée lorsque le robot sort du couloir et lorsque les données des sonars lui apportent une information nouvelle.

La procédure d'exploration utilisée par notre modèle montre des limitations dans de telles situations. En effet, notre procédure d'exploration est une procédure locale qui a tendance à limiter les déplacements du robot afin d'éviter qu'il ne s'éloigne trop de la zone cartographiée. Dans le cas de relocalisation évoqué, où il est nécessaire de sortir du couloir pour pouvoir estimer correctement sa position, cette procédure présente l'inconvénient d'empêcher le robot de sortir rapidement du couloir et interdit donc une relocalisation rapide. La solution temporairement adoptée pour résoudre ce problème est d'assigner un but au robot en dehors du couloir concerné, afin de faciliter une relocalisation rapide. Toutefois, il serait intéressant d'implémenter des procédures de *navigation active* qui permettraient de choisir le déplacement du robot de façon à assurer une estimation efficace de la position. Nous reviendrons sur cette suggestion dans le prochain chapitre.

6.3.3 Déplacement vers un but

Enfin, nous avons utilisé notre système de navigation pour planifier des déplacements dans l'environnement. Nous avons pour cela effectué dix essais, depuis différentes positions de l'environnement afin d'atteindre un but situé devant une porte, au milieu d'un couloir. La figure 6.36 montre trois exemples de trajectoires obtenues.

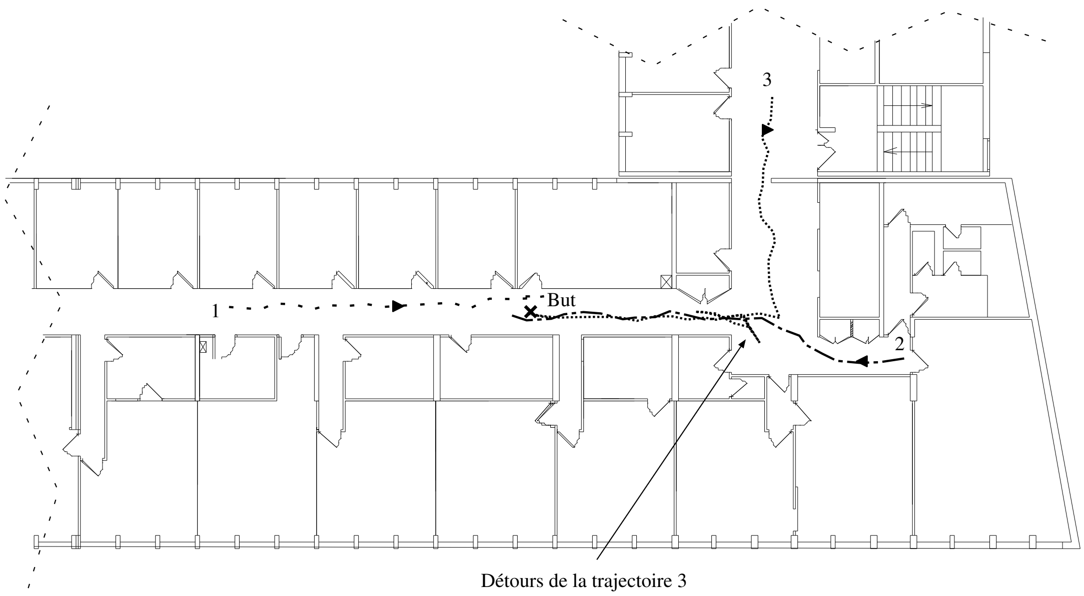


Fig. *6.36: Trois exemples de trajectoires mesurées par l'odométrie sur le robot réel en utilisant la carte de la figure 6.30. Les trajectoires 1 et 2 permettent d'atteindre le but directement, avec un précision d'environ 50 cm. La trajectoire 3 permet d'atteindre le but avec la même précision, mais après des détours dûs à une mauvaise estimation de la position au bout du couloir.*

Sur ces dix essais, un n'a pas permis de rejoindre directement le but à cause d'une mauvaise estimation de la position. Les procédures permettant la reprise de la planification n'ont pas permis, dans ce cas, de sortir le robot d'un coin dans lequel il était bloqué. Afin d'atteindre néanmoins le but dans ce cas, nous avons associé un autre but au robot le guidant vers une zone où la localisation est relativement efficace. Nous lui avons ensuite de nouveau donné le but initial qu'il a alors atteint sans problèmes. Une telle procédure pourrait être implémentée sous la forme d'une procédure navigation active qui guiderait le robot pour une localisation efficace. Nous reviendrons sur ces méthodes dans le prochain chapitre.

Les neuf autres essais ont, par contre, permis de rejoindre le but, soit directement dans 5 cas, soit après déclenchement de la procédure de détour et une ou plusieurs replanifications dans les 4 autres cas. La précision finale obtenue en moyenne sur ces neuf essais correspond à la taille du robot, soit 50 cm. Aucun de ces essais ne s'est achevé à moins de 80 cm du but.

Ces résultats montrent donc que notre système de navigation permet la construction correcte d'une carte de l'environnement en utilisant un robot réel. Il permet également d'estimer la position au sein de cette carte et de rejoindre de manière fiable des buts qui lui sont assignés dans l'environnement.

Chapitre 7

Discussion

7.1 Comparaison avec des modèles existants

Afin de montrer les apports de notre système de navigation, nous allons le comparer à deux systèmes réalisant simultanément la cartographie et la localisation dont nous nous sommes inspirés. Le premier modèle est celui d'Arléo et Gerstner [Arleo and Gerstner, 2000], un modèle biomimétique qui s'inspire des structures présentes dans l'hippocampe du rat. Le second modèle est celui de Thrun et al. [Thrun et al., 2000] qui utilise un modèle markovien pour la localisation et effectue simultanément une cartographie «en-ligne».

7.1.1 Modèle d'Arléo et Gerstner

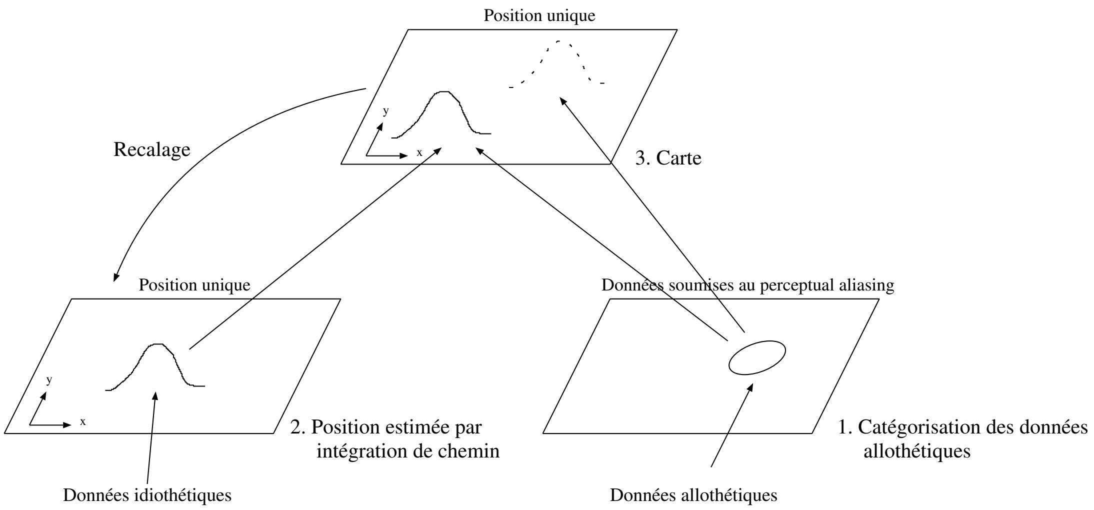


FIG. *7.1: Structure fonctionnelle du modèle d'Arléo et Gerstner (voir le texte pour les explications).*

Le modèle d'Arléo et Gerstner est composé de trois modules distincts, chacun contenant un groupe de neurones différent (cf. figure 7.1). Le premier module reçoit en entrée les données allothétiques et permet de les catégoriser. Comme dans notre modèle, aucun modèle métrique n'est associé à ces données. Chaque neurone est activé lorsque les données allothétiques caractéristiques d'un lieu donné sont présentes en entrée. Comme l'environnement est soumis au perceptual aliasing, l'activation de certains neurones peut correspondre à plusieurs positions dans l'environnement et cette partie ne suffit donc pas pour estimer la position de l'animat de manière non ambiguë.

Le second module permet de représenter la position de l'animat en utilisant les données idiothétiques. Pour cela, chacun des neurones de ce module est associé à une position dans un espace métrique à deux dimensions. La position de l'animat est définie par un point dans cet espace métrique, dont la position est mise à jour par intégration des données idiothétiques. L'activité de chacun des neurones est une fonction décroissante de la distance de sa position à la position de l'animat et le résultat d'ensemble est une «bulle d'activité», centrée autour de la position de l'animat. Il ne faut pas confondre l'activité de ces neurones, qui est fonction de la distance entre la position associée au neurone et la position de l'animat, avec l'activité des nœuds de notre modèle qui représente la probabilité que l'animat se trouve à cette position. Dans le modèle d'Arléo et Gerstner, cette bulle d'activité représente la position de manière unique au moyen d'un codage par population de vecteurs, et ne représente pas plusieurs hypothèses de position. L'estimation de position fournie par ce module n'est pas liée à l'environnement par les données allothétiques.

Ce lien est réalisé par le troisième module, contenant la carte proprement dite de l'environnement. Chaque neurone de ce module est connecté à des neurones des deux modules précédents et son activité dépend de l'activité des neurones auxquels il est connecté. Ainsi les neurones activés dans ce module seront ceux qui correspondent à la fois aux données allothétiques et à la position courante estimée par les données idiothétiques. Chacun des neurones de ce module permet donc de relier une position donnée dans l'environnement aux données allothétiques disponibles à cette position. L'activité globale obtenue permet de définir la position de l'animat de manière non ambiguë par une bulle d'activité centrée autour de cette position.

De plus, le modèle d'Arléo et Gerstner permet de recaler l'estimation idiothétique de la position en fonction des données allothétiques, lors de l'introduction de l'animat dans son environnement, ou lorsque la qualité de cette estimation devient trop faible. Il repose pour cela sur des positions de l'environnement qui sont uniques du point de vue allothétique et qui permettent donc d'estimer la position sans l'apport des données idiothétiques. Pour détecter ces positions, l'activité des neurones de la carte est calculée en fonction des seules données allothétiques. Lorsqu'une telle position est atteinte, c'est-à-dire lorsque l'activation des neurones du premier module ne correspond qu'à une seule position de l'environnement, l'activité des neurones de la carte prend la forme d'une «bulle d'activité» centrée autour d'un point. Cette activité est alors transférée dans le deuxième module du modèle et sert de nouvelle estimation de la position[2].

[2]La méthode de calibration implémentée dans le modèle utilise en fait l'activation des neurones du premier module afin de recaler cette position. Or ces neurones ne sont pas associés à une position de l'environnement. La position est donc calculée par un dispositif externe au modèle qui permet d'estimer la position correspondant

La différence essentielle entre le modèle d'Arléo et Gerstner et le nôtre est l'utilisation, dans leur cas, d'une hypothèse unique de position. La position du robot est ainsi estimée par les données idiothétiques dans une partie du modèle séparée de la carte en elle-même. Cette position est représentée sous forme d'un point unique. Le modèle d'Arléo et Gerstner réalise donc un suivi de position. L'estimation initiale de cette position est fournie par le processus de recalage que nous avons décrit et utilise des positions de l'environnement uniques du point de vue allothétique. Cette procédure correspond à une localisation globale n'utilisant que les données allothétiques (cf. chapitre 2).

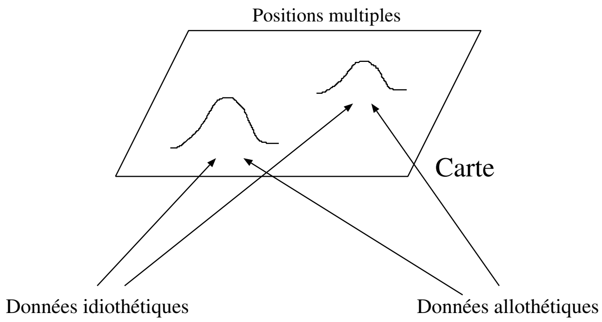


FIG. *7.2: Schéma fonctionnel de notre modèle (voir le texte pour les explications).*

Notre modèle, en revanche, réalise une localisation globale utilisant à la fois les données allothétiques et idiothétiques. Il ne dépend donc pas d'une position unique du point de vue allothétique pour estimer globalement sa position, mais capitalise à la place sur des séquences uniques de perceptions et de déplacements. Ainsi notre modèle pourra fonctionner dans un environnement dans lequel toutes les situations allothétiques correspondent à au moins deux positions différentes de l'environnement, du moment que des suites de perceptions et de déplacements uniques existent.

Pour réaliser cette localisation globale, notre modèle utilise plusieurs hypothèses de position qui ne sont pas représentées dans une partie séparée de la carte, comme dans le modèle d'Arléo et Gerstner, mais intégrées au sein de cette carte. Les données idiothétiques sont donc utilisées directement dans la carte pour mettre à jour ces différentes hypothèses en parallèle (cf. figure 7.2). La partie réalisant la catégorisation des données allothétiques n'est pas non plus présente explicitement dans notre modèle. Les données allothétiques sont en effet utilisées directement au sein de la carte pour être comparées avec les données mémorisées dans les différents nœuds. Dans le modèle d'Arléo et Gerstner, cette comparaison est réalisée dans une partie séparée de la carte mais cette différence ne semble pas avoir de conséquences importantes sur les capacités respectives des modèles.

En conséquence de ces différences de fonctionnement et de structure, la procédure de recalage

aux neurones de ce premier module. L'information requise est toutefois présente dans le modèle au niveau du troisième module représentant la carte, qui permet d'associer des données allothétiques et une position. Comme les auteurs le mentionnent, cette carte peut donc être utilisée pour la calibration comme nous le décrivons ici, sans utilisation d'une méthode externe au modèle.

de la position est inutile dans notre modèle. Elle est remplacée par le recalage continu qui est réalisé par l'estimation des crédibilités relatives des différentes hypothèses de position. Notre modèle gagne ainsi la possibilité de fonctionner dans des environnements pour lesquels aucune position n'est unique du point de vue allothétique.

Notre modèle est donc potentiellement plus performant pour l'estimation et le recalage de la position. Toutefois, le modèle d'Arléo et Gerstner est biologiquement plausible, ce qui n'est pas le cas du nôtre car il ne cherche pas à reproduire les structures et les flux de données existant dans le cerveau des rats. Par exemple, notre modèle mémorise dans chaque nœud les données allothétiques disponibles au lieu correspondant dans l'environnement alors que le modèle d'Arléo et Gerstner utilise les poids des synapses qui déterminent l'activité de chaque neurone pour mémoriser cette information. La fonction réalisée est donc la même dans les deux cas, mais nous nous sommes affranchis des contraintes de plausibilité biologique. La capacité de localisation théoriquement plus limitée de leur modèle est donc plus en accord avec les connaissances biologiques actuelles sur les mécanismes de navigation du rat. De plus, cette limitation de la capacité de localisation semble compensée chez le rat par la puissance de perception. En effet, les positions différentes strictement identiques pour le rat du point de vue allothétique semblent assez rares. Notre modèle gagne en intérêt lors de l'utilisation sur des robot mobiles aux capacités de perception plus limitées et donc beaucoup plus soumis au *perceptual aliasing*.

7.1.2 Modèle de Thrun et al.

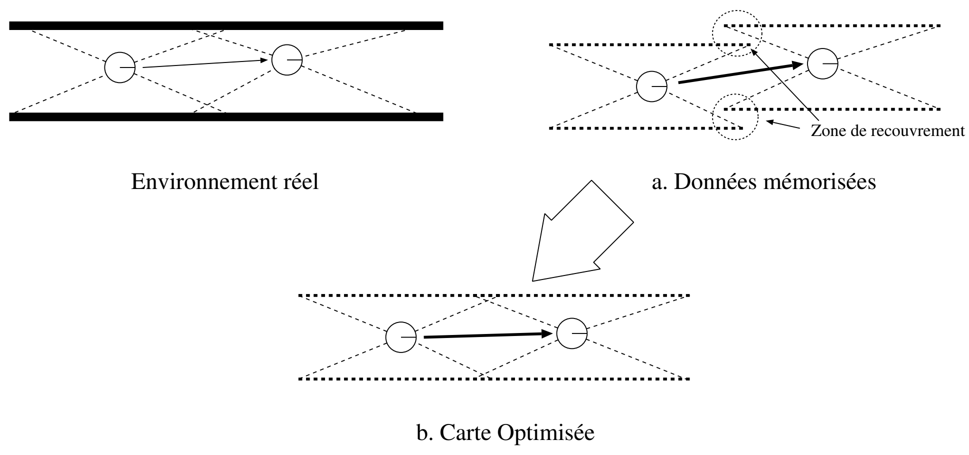


Fig. *7.3: Type de carte utilisée par le modèle de Thrun et al. Des données recueillies par un télémètre laser sont mémorisées, ainsi que les déplacements du robot entre les perceptions (Partie a). L'utilisation de données allothétiques utilisant un modèle métrique et se recouvrant permet par la suite d'optimiser la carte pour assurer la cohérence des données (Partie b, voir le texte pour les détails).*

Le modèle de Thrun et al. est un modèle purement robotique qui permet de construire «en-ligne» une carte métrique de l'environnement tout en utilisant une méthode de localisation globale. La méthode de cartographie utilisée permet le retour en arrière lorsque la fermeture de cycles dans l'environnement apporte de l'information sur les positions passées du robot. Pour cela, la carte utilisée mémorise les données allothétiques et idiothétiques de manière dissociées (cf. paragraphe 3.3.1). Dans l'implémentation de ces auteurs, des ensembles de points obtenus grâce à un télémètre laser ayant un modèle métrique associé, ainsi que la position depuis laquelle ces points ont été obtenus, sont enregistrés. Les différentes positions de perception sont suffisamment rapprochées pour que deux ensembles de points consécutifs se recouvrent partiellement, ce qui permet d'utiliser les données allothétiques et le modèle métrique associé, en plus de l'odométrie, pour estimer le déplacement du robot (cf. paragraphe 1.3.2 et figure 7.3).

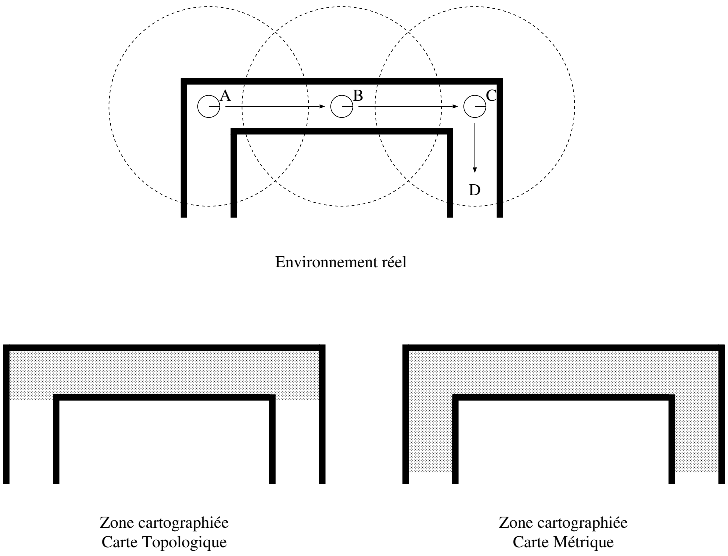


FIG. *7.4: L'utilisation d'une carte métrique et d'un modèle métrique associé aux capteurs permet de cartographier une zone de l'environnement plus grande que la zone physiquement explorée par le robot. Dans cet exemple, si les informations recueillies aux positions A, B et C sont utilisées, une carte métrique contiendra la position D, tandis qu'une carte topologique ne la contiendra pas. La zone cartographiée correspond de plus à la zone au sein de laquelle il est possible d'utiliser une méthode de localisation globale.*

La position du robot est estimée en utilisant une méthode de localisation markovienne représentant la probabilité de présence du robot de manière continue sur l'espace couvert par la carte. La crédibilité relative des différentes positions possibles en fonction des données allothétiques peut être facilement estimée dans ce modèle grâce à l'utilisation du modèle métrique associé aux données allothétiques. Ce modèle métrique permet en effet d'avoir à tout moment une carte couvrant une partie de l'environnement plus grande que celle dans laquelle le robot s'est physiquement déplacé (cf. figure 7.4). Cette garantie que le robot se trouve dans une zone cartographiée, et donc dans laquelle la localisation globale fonctionne, permet d'utiliser la méthode de localisation markovienne dans tous les cas.

Au contraire, notre modèle n'utilise pas de modèle métrique des capteurs et par conséquent la localisation globale ne fonctionne que dans la zone de l'environnement qui a déjà été physiquement explorée. Il faut donc, au cours de la cartographie, estimer si le robot se trouve ou non dans une zone de l'environnement couverte par la carte. Dans notre modèle, une heuristique est utilisée pour réaliser cette estimation et utiliser la localisation globale si le robot se trouve dans la zone couverte par la carte, ou le suivi de position dans le cas contraire.

Le modèle de Thrun et al. cartographie son environnement en ajoutant chaque nouvel ensemble de points perçu associé à la position la plus probable du robot au moment de la perception. Cette opération correspond à une phase de simple cartographie incrémentale, avec l'apport d'une estimation robuste de la position grâce à la méthode markovienne. Une méthode permettant un retour sur les modifications passées est de plus utilisée lorsqu'un cycle est détecté dans l'environnement. Pour cela, l'erreur de positionnement détectée lors de la fermeture du cycle est propagée avec une amplitude décroissante tout le long du cycle (cf. figure 3.9). Les positions des différentes parties de la carte sont ensuite optimisées afin de permettre une cohérence maximale de la carte autour du cycle. Dans ce modèle, la cohérence de la carte est définie de manière différente de la cohérence dans notre modèle. En effet, elle est mesurée en fonction du recouvrement des données allothétiques entre deux positions successives, ce qui n'est possible que grâce à l'utilisation d'un modèle métrique pour le capteur. Ainsi une carte sera cohérente lorsque tous les ensembles de points ayant une partie commune seront identiques sur cette partie commune.

Cette méthode de répartition de l'erreur le long du cycle et de maintien de la cohérence fonctionne de manière similaire à la méthode utilisée dans notre modèle. Toutefois, le modèle de Thrun et al. bénéficie de l'apport du modèle métrique des données allothétiques qui permet de définir la cohérence de la carte en fonction du recouvrement des données allothétiques, et non en fonction des seules données idiothétiques comme c'est le cas dans notre modèle. L'utilisation de capteurs très précis que sont les télémètres laser permet ainsi d'obtenir une carte plus précise que dans notre modèle.

Notre modèle présente donc des fonctionnalités très similaires à celui de Thrun et al. En effet, il permet une estimation de la position utilisant une méthode markovienne et ajoute les données perçues à la position courante la plus probable du robot. Lors de la fermeture d'un cycle non cohérent, les positions des différents nœuds appartenant au cycle sont optimisées pour tenir compte de cette contrainte. Ces méthodes permettent ainsi d'obtenir une cartographie «en-ligne». Les différences principales entre nos modèles proviennent de l'utilisation, dans notre cas, d'une carte topologique et de capteurs beaucoup moins précis sans modèle métrique. La réalisation de telles fonctionnalités au sein de cartes topologiques n'a, à notre connaissance, été réalisée dans aucun modèle utilisant des PDMPO associés à des cartes topologiques, tous ces modèles utilisant des méthodes de cartographie «hors-ligne» [Cassandra et al., 1996, Hertzberg and Kirchner, 1996, Shatkay and Kaelbling, 1997, Simmons and Koenig, 1995, Theocharous et al., 2001]. La raison de l'inexistence de tels modèles semble être la difficulté de déterminer si le modèle actuel de l'environnement contient ou non la position courante du robot. Cette détermination est relativement simple lorsqu'on utilise une carte métrique, grâce à la couverture très complète qui est faite de l'environnement, mais s'avère plus complexe pour une carte topologique.

Cette utilisation de capteurs sans modèles métriques oblige notre modèle à explorer de manière beaucoup plus exhaustive l'environnement et conduit à une estimation de la position moins précise tant que la carte de l'environnement n'est pas complète. En revanche, notre modèle est capable d'utiliser les informations de tous les types de capteurs, incluant ceux ne pouvant pas être associés à un modèle métrique. Il est ainsi possible de répertorier dans notre carte des informations non métriques, telles que les couleurs ou la température de l'environnement, en plus de la simple position des obstacles. L'utilisation des capteurs est également beaucoup plus simple que dans le modèle de Thrun et al. car le modèle probabiliste qui leur est associé prend simplement en compte la similarité de deux perceptions. Le modèle probabiliste utilisé par Thrun et al. est relativement complexe car il utilise le modèle métrique associé au capteur et est spécifiquement adapté au type de capteur utilisé.

7.1.3 Résumé

Les apports de notre modèle peuvent donc être vus sous deux aspects différents. Il peut en effet être présenté comme ajoutant une capacité de localisation globale à des systèmes classiquement utilisés pour modéliser le fonctionnement du cerveau des rats pour la navigation. De tels systèmes utilisent en effet des cartes topologiques et des capteurs allothétiques sans modèles métrique, comme le fait notre modèle. Dans le même temps, notre modèle permet une localisation globale parce qu'il s'inspire de méthodes existant par ailleurs en robotique mobile. Notre modèle présente ainsi une structure plus simple en utilisant une seule méthode de localisation, sans avoir recours à une méthode de suivi de position qui nécessite d'être régulièrement recalée pour garder sa cohérence avec l'environnement. Notre modèle peut également fonctionner dans des environnements plus complexes, c'est-à-dire soumis à un plus fort perceptual aliasing.

D'un autre point de vue, notre modèle peut être présenté comme une adaptation à l'utilisation d'une carte topologique et de capteurs sans modèles métriques de systèmes de robotique classique utilisant des modèles markoviens. Comme certaines de ces réalisations, notre modèle permet, en effet, une estimation globale de la position et une cartographie simultanée. Le gain obtenu est l'utilisation possible de capteurs plus variés, ainsi qu'une mise en œuvre plus simple de chaque capteur qui ne nécessite plus d'être intégré dans un cadre de référence unique via un modèle métrique. Il faut noter que, à notre connaissance, aucun système de navigation markovien utilisant des cartes topologiques ne permet de réaliser une cartographie «en-ligne». Seuls certains modèles basés sur des carte métriques, comme celui de Thrun et al., le permettent et reposent pour cela fortement sur le modèle métrique associé aux capteurs.

7.2 Bilan

D'un point de vue général, notre système assure donc une grande autonomie au robot utilisé. En effet, les phases de localisation, cartographie et relocalisation en cas d'erreur d'estimation de la position sont totalement intégrées et ne demandent aucune intervention extérieure. La localisation obtenue est de plus une localisation globale, qui ne requiert aucune estimation initiale de la position. Notre modèle peut également fonctionner dans des environnements soumis à un très fort *perceptual aliasing*, allant jusqu'à des environnements dans lesquels aucune position n'est unique du point de vue des capacités perceptives du robot.

Cette autonomie est obtenue en utilisant des capteurs relativement simples, qui ne demandent pas de mise en œuvre complexe. En effet, les valeurs mesurées par les sonars sont directement utilisées, sans modèle métrique et sans modélisation précise du bruit auquel ces capteurs sont soumis. De même, les images obtenues par la caméra sont simplement sous-échantillonnées, sans aucun algorithme complexe de traitement d'image. Cette simplicité conduit à un fort *perceptual aliasing* qui est compensé par la méthode de localisation utilisée.

La précision de la localisation obtenue permet de piloter le robot de manière suffisamment précise pour un grand nombre de tâches. L'erreur dans l'estimation de la position est, en effet, de l'ordre du diamètre du robot, soit environ cinquante centimètres. Cette précision permet au robot, par exemple, de rejoindre une pièce dans un environnement de bureau dont les portes et les couloirs sont en moyenne plus larges que cette erreur. Cette précision ne permet évidemment pas de réaliser des tâches telles que la prise d'un objet, mais permet de guider le robot à proximité d'un objet où une procédure locale permettra le positionnement précis par rapport à l'objet recherché.

Nous avons démontré l'efficacité des méthodes de perception actives que nous avons utilisées dans le cas où une carte complète de l'environnement est disponible. La méthode la plus efficace dans ce cadre se révèle être celle qui maximise le gain d'entropie sur la distribution de probabilité représentant la position. Cette méthode sélectionne la direction conduisant potentiellement à la plus faible ambiguïté de l'estimation de la position. Ce résultat est conforme aux attentes théoriques car cette méthode est celle qui prend le mieux en compte la situation courante pour sélectionner la direction de perception.

Cependant, nous avons également montré que pour le problème de cartographie et de localisation simultanée, les procédures de perception active les plus efficaces sont celles qui privilégient la complétude de l'information, au lieu de sa capacité de discrimination. Dans ce cadre, la méthode maximisant le gain d'entropie conduit à des performances similaires à une sélection aléatoire de la direction. La méthode privilégiant la complétude de l'information permet par contre d'obtenir une précision proche de celle que permettrait l'utilisation d'un capteur panoramique.

La limitation principale de notre système de navigation est la nécessité de disposer d'une estimation relativement correcte de la direction absolue. Cette estimation doit donc être fournie par un capteur fiable pour que le robot soit complètement autonome, indépendamment de l'environnement. Toutefois, ces capteurs sont soit très onéreux[3], soit fonctionnent très mal dans des environnements intérieurs[4]. La solution actuelle, qui utilise un recalage périodique de l'estimation de la direction fournie par l'odométrie, présente l'inconvénient de restreindre l'autonomie

[3] C'est, par exemple, le cas des gyroscopes de bonne qualité

[4] C'est le cas des compas magnétiques

du robot car sa direction, mesurée par un dispositif externe doit lui être communiquée régulièrement. Ce recalage pourrait toutefois être modifié pour fonctionner sans intervention extérieure, ni aménagements de l'environnement.

Un autre inconvénient de notre système est la nécessité d'une exploration exhaustive de l'environnement pour obtenir une carte complète. Cette exploration est beaucoup plus longue que l'exploration typiquement nécessaire pour obtenir une carte métrique complète du même environnement avec les modèles de cartographie métrique classiques. Ce problème est encore accentué dans le cas de l'utilisation de capteurs directionnels. Dans le cas de l'utilisation de capteurs panoramiques, en effet, l'exploration de l'environnement ne requiert qu'un passage en chacun des lieux à représenter. Dans le cas des capteurs directionnels, il faut plusieurs passages pour obtenir une représentation complète d'un lieu.

Notre système de localisation montre de plus certaines limites sur un robot réel. En effet, la relocalisation lorsque le robot est déplacé le long d'un même couloir est très délicate. Notre modèle ne permet d'estimer correctement la position que lorsque le robot sort du couloir, c'est-à-dire lorsque les données omnidirectionnelles (les données des sonars dans notre cas) varient. Ces difficultés sont dues principalement à la conjonction de l'incomplétude des données directionnelles et au bruit important entachant ces données.

Enfin, la méthode de planification utilisée peut ne pas permettre d'atteindre correctement le but dans certains cas. Cette méthode pose notamment des problèmes lorsque des zones de l'environnement sont séparées par des obstacles de faible épaisseur. Dans ce cas, en effet, des trajectoires traversant ces obstacles peuvent être planifiées qui ne permettront évidemment pas d'atteindre directement le but. Cependant, les procédures vérifiant l'exécution correcte du plan permettent d'atteindre le but dans un tel cas, au prix d'une trajectoire fortement sous-optimale dont la longueur est très supérieure à la longueur du chemin le plus court.

7.3 Perspectives

Différentes prolongations intéressantes de notre travail sont apparues durant le développement du modèle présenté dans ce mémoire. En premier lieu, il serait souhaitable d'inclure des méthodes d'estimation de la direction et des méthodes de *navigation active* afin de renforcer la capacité de localisation de notre modèle dans des environnement très symétriques. Les cartes construites par notre modèle sont également un substrat intéressant pour des méthodes de planification plus efficaces que celles qui ont été utilisées dans ce travail. Enfin, des similarités de notre modèle avec certaines données biologiques peuvent conduire à s'interroger sur la signification des activités des cellules de lieux chez les rats.

7.3.1 Estimation de la direction

Il serait souhaitable d'ajouter à notre modèle la capacité d'estimer la direction du robot de manière autonome. Deux méthodes sont a priori utilisables pour cela.

La première méthode, utilisée par exemple dans le modèle de Simmons et Koenig [Simmons and Koenig, 1995] avec une carte topologique donnée a priori et associée à un PDMPO, serait d'ajouter aux nœuds de la carte une information de direction. Chaque nœud représenterait alors une position et une direction du robot. Cette méthode conduit cependant à un espace de recherche beaucoup plus grand et semble relativement complexe à intégrer dans une logique utilisant une carte topologique irrégulière telle que la nôtre. De plus, cette représentation augmente encore le perceptual aliasing, du fait que la direction du robot n'est plus donnée par un dispositif extérieur. Le niveau de perceptual aliasing atteint semble difficilement compatible avec une construction «en-ligne» de la carte.

La seconde méthode, qui nous semble plus simple à mettre en œuvre et plus efficace, consiste à améliorer le recalage périodique de la direction utilisé pour notre implémentation sur le robot réel. Ainsi, au lieu de fournir une direction de référence au système de navigation par un dispositif externe, il serait plus pertinent que le système cherche et mémorise la direction d'un amer distant depuis une position donnée. La direction de cet amer lui servirait par la suite de direction de référence lors de son retour à cette position. Le système de navigation serait ainsi complètement autonome et n'utiliserait plus de référence artificielle de direction.

7.3.2 Navigation active

Nous nous sommes intéressés dans notre thèse à la perception active, c'est-à-dire au choix des actions que l'on peut effectuer avec un capteur dans le but d'améliorer la capacité de localisation d'un robot. Les déplacements de ce robot sont, pour leur part, sélectionnés par une procédure d'exploration locale dont le but est d'assurer une couverture exhaustive de l'environnement tout en évitant qu'il ne s'éloigne trop de la zone cartographiée.

Comme nous l'avons vu dans la section 6.3.2, ces deux procédures peuvent se révéler limitées lorsque deux zones identiques de grandes tailles existent dans l'environnement. En effet, si le robot est déplacé au sein d'une de ces zones sans qu'il puisse utiliser le suivi de position, deux hypothèses de position correspondant à ces deux zones seront soulevées. La perception active ne permettra pas de sélectionner l'une de ces hypothèses puisque l'environnement est identique dans les deux cas. Le choix de l'hypothèse correcte ne pourra donc se faire qu'en sortant de ces zones identiques, ce qui peut être long car les procédures d'exploration ont tendance à favoriser une exploration locale exhaustive de l'environnement.

Pour résoudre ce type de problème, il serait pertinent de s'intéresser aussi à la navigation active [Fox et al., 1998a], c'est-à-dire au choix d'un déplacement qui conduirait à une localisation plus précise. Pour cela, il est possible d'utiliser la même mesure d'incertitude utilisant l'entropie que celle utilisée pour la perception active. Le problème est alors de trouver le déplacement qui conduira à la distribution de probabilité ayant la plus faible entropie.

Cette entreprise est toutefois plus complexe que la simple détermination de la direction d'un capteur car l'espace de recherche correspondant est plus grand. De plus, pour être efficace et résoudre les problèmes de localisation que nous avons mentionnés, les procédures correspondantes devront prendre en compte plusieurs déplacements successifs du robot afin de le guider hors de zones identiques de l'environnement. Or le modèle très imprécis que nous utilisons pour l'odométrie conduit rapidement à une très grande incertitude s'il n'est pas utilisé en conjonction avec des données allothétiques. Prévoir une distribution de probabilité après plusieurs déplacements successifs est donc délicat et risque de ne pas être d'une précision suffisante pour sélectionner un déplacement pertinent.

7.3.3 Planification

Notre modèle utilise une méthode de planification très simple. Cette méthode peut parfois se révéler inefficace si l'estimation de la position est incorrecte et générer des comportements cycliques qui ne conduisent pas au but [Cassandra et al., 1996]. Si de tels comportements n'ont jamais été observés dans nos expériences, diverses améliorations susceptibles de garantir un fonctionnement correct dans tous les cas peuvent être apportées à notre système de planification. En particulier, il serait utile de ne chercher à se diriger vers le but que lorsque l'estimation de la position du robot est suffisamment précise. Dans le cas où cette estimation est jugée peu fiable, il serait préférable d'utiliser une procédure de navigation active telle que mentionnée précédemment afin d'améliorer l'estimation de la position avant de chercher à se diriger vers le but. De telles procédures permettraient de prendre en compte l'imprécision de l'estimation de la position de manière plus efficace que nous ne le faisons à l'heure actuelle.

Un autre axe de recherche intéressant serait l'ajout d'informations supplémentaires concernant la navigation de l'animat au sein de la carte. En effet, la carte que nous utilisons actuellement dans notre modèle représente simplement la structure spatiale de l'environnement dans le but de permettre d'atteindre un point quelconque de l'environnement. Il serait intéressant d'ajouter à cette carte des informations telles que, par exemple, la difficulté de traverser chaque zone de l'environnement. Ainsi, le plan calculé pourrait entraîner des détours pour éviter des zones difficiles, s'il n'est pas urgent d'atteindre le but, ou pourrait conduire à prendre le risque de traverser une zone difficile si le temps presse. La prise en compte de tels facteurs peut se réaliser de manière très naturelle dans le cadre du calcul de politique au sein des PDMPO dans lesquels des récompenses et des punitions traduisant ces facteurs peuvent être associées aux actions et aux états du système. Rechercher la politique optimale du PDMPO maximisant la récompense globale permet alors de tenir compte de tous ces facteurs.

Dans le cas d'un robot ayant plusieurs tâches simultanées à résoudre, il serait également intéressant d'ajouter directement à la carte des informations sur l'intérêt des différents lieux de l'environnement vis-à-vis de chacune des tâches du robot. La planification pourrait ainsi chercher des chemins permettant de résoudre au mieux toutes les tâches du robot simultanément. Cet axe de recherche sera développé au sein du projet Psikharpax qui débute dans notre laboratoire. Ce projet vise à synthétiser dans un robot certaines des données biologiques connues sur la navigation et la sélection de l'action chez le rat. La sélection de l'action est le problème consistant à choisir à chaque instant quelle action exécuter en fonction de la situation courante et de motivations concernant différentes tâches contradictoires à résoudre. Au niveau de la navigation, une stratégie de sélection de l'action efficace devra donc planifier des déplacements rejoignant les positions permettant de résoudre les différentes tâches.

Avec notre modèle actuel, la position à atteindre doit être déterminée par un processus extérieur et la carte peut alors être utilisée pour rejoindre directement cette position. Cette stratégie permettrait de résoudre successivement les différentes tâches. Toutefois, en intégrant les différents buts possibles à la carte, il deviendrait possible de réaliser, en plus, des compromis et de générer des trajectoires permettant de résoudre les différentes tâches simultanément. Prenons l'exemple d'un rat artificiel ayant très faim et peu soif : notre modèle actuel permettrait de le diriger directement vers une source de nourriture en ignorant la soif, puis de le diriger vers une source de boisson. Mais, dans le cas où le chemin direct vers la nourriture passe près d'une source de boisson, il serait plus efficace d'effectuer un léger détour pour boire avant d'aller manger. Un tel compromis pourra être réalisé en intégrant les différents buts directement à la carte, puisqu'il deviendra alors possible de planifier des trajectoires sous la contrainte que les différentes tâches soient résolues simultanément.

7.3.4 Fonctionnement des systèmes biologiques

Notre modèle associe une activité à chaque nœud de la carte, comme le font les systèmes de navigation biologiquement plausibles, sans chercher pour autant à reproduire les structures fonctionnelles existant chez le rat. Toutefois, l'idée que l'activité des nœuds représente une probabilité de présence du robot à la position correspondante trouve certains échos dans la littérature biologique.

En premier lieu, l'activité d'un groupe de neurones semble pouvoir représenter une distribution de probabilité complète sur les valeurs d'une variable [Zemel et al., 1997]. En effet, de manière classique, les méthodes telles que le codage par population de vecteur supposent que l'activité d'un groupe de neurones représente une valeur unique d'une variable. Or, les modèles de navigation biologiquement plausibles utilisent en général une telle méthode de codage par population de vecteurs et supposent donc que l'activité des cellules de lieux représente une position unique. Cela revient à faire l'hypothèse que le rat réalise un suivi de position. Si on suppose que l'activité des cellules de lieux représente une distribution de probabilité sur les positions possibles, et non une position unique, il est possible de concevoir que le cerveau des rats réalise une fonction de localisation globale similaire à celle réalisée par notre modèle.

D'autre part, Zhang et al. [Zhang et al., 1998] montrent que la reconstruction de la position d'un rat à partir de l'activité enregistrée de ses cellules de lieux est beaucoup plus précise si on utilise une méthode probabiliste que si on utilise une méthode faisant appel à des *fonctions de base*. Parmi ces dernières méthodes se trouvent notamment le codage par population de vecteurs classiquement utilisé dans les modèles biomimétiques. Les méthodes probabilistes, pour leur part, associent à chaque cellule de lieu une probabilité de présence sur les différentes positions de l'environnement. La position du rat est estimée en utilisant la loi de Bayes qui permet de construire, à partir de l'activité des cellules de lieu, une distribution de probabilité sur l'ensemble des positions possibles du rat. La position la plus probable obtenue se révèle alors plus précise que la position obtenue par les méthodes utilisant des fonctions de base. L'article de Zhang et al. montre également que ces mécanismes de reconstruction probabilistes peuvent être implémentés sous forme de réseau de neurones et suggère donc qu'ils puissent être utilisés par le rat.

Ces différents indices en faveur de l'utilisation d'une méthode de localisation probabiliste chez le rat ne permettent cependant absolument pas de rejeter l'idée communément admise que le rat réalise un simple suivi de position. Néanmoins, l'étude du fonctionnement d'un modèle tel que le nôtre pourrait amener à imaginer des expériences sur le rat permettant de faire avancer cette question. Tester ces hypothèses reste toutefois problématique car les enregistrements de cellules de lieu demandent, à l'heure actuelle, d'effectuer une mesure moyenne sur un temps relativement long. Une telle contrainte empêche l'observation des phénomènes transitoires qui permettraient de distinguer une approche classique de suivi de position d'une approche probabiliste telle que la nôtre. En effet, le rat possédant de très bonnes capacités de navigation, les situations dans lesquelles il aurait une estimation ambiguë de la position semblent devoir être de très courte durée. Les méthodes de mesure en moyenne utilisées rendent difficile de mesurer si, durant cette courte durée, une seule ou chacune des différentes positions possibles sont représentées par les cellules de lieu.

7.3.5 Intégration dans un système de navigation plus général

Nous avons mentionné dans l'introduction que les stratégies de navigation par cartes ne représentent qu'une partie des stratégies de navigation existantes. De telles stratégies sont très utiles lorsque l'on cherche à rejoindre un but lointain dans l'environnement, mais peuvent être moins efficaces qu'une simple stratégie de guidage vers un but lorsque celui-ci est proche et visible. Ainsi, il serait intéressant d'intégrer notre modèle dans un système de navigation plus général, pouvant mettre en œuvre différentes stratégies de navigation suivant le contexte, et permettant de renforcer encore la robustesse de la tâche de navigation.

Notre modèle se prête très bien à cette intégration. Il peut en particulier être utilisé pour diriger le robot vers un but lointain et invisible, puis être remplacé par une stratégie de guidage permettant d'atteindre précisément le but. Lorsque cette seconde stratégie peut être mise en œuvre, notre modèle peut simplement recevoir passivement les informations servant à cartographier l'environnement et à estimer la position du robot. Cette estimation continue de la position, qui est à l'œuvre quelle que soit la tâche du robot ou sa stratégie de déplacement, permet ainsi à tout moment d'utiliser notre modèle pour atteindre un but dans l'environnement. Cette intégration avec d'autre stratégies de navigation pourrait également être réalisée dans le cadre du projet Psikharpax (cf. paragraphe 7.3.3).

Conclusion

En résumé, nous avons développé un système de navigation pour un robot autonome basé sur une carte de l'environnement permettant de résoudre simultanément les problèmes de cartographie et de localisation. Nous avons utilisé un mode de représentation de l'espace classiquement utilisée pour modéliser les capacités de navigation du rat. Pour exploiter cette représentation, nous avons utilisé une méthode de localisation globale inspirée de modèles probabilistes utilisés en robotique classique. Pour ce faire, nous avons résolu les problèmes d'incomplétude de l'information qui limitent traditionnellement ces approches pour la cartographie. De plus, notre modèle utilise des stratégies de perception actives qui permettent de choisir dans quelle direction orienter les capteurs du robot dans le but d'améliorer l'estimation de sa position.

Notre modèle permet donc de construire une carte de l'environnement, tout en estimant de manière robuste la position du robot. Il permet notamment de retrouver rapidement cette position lorsque ce robot est déplacé de manière passive dans son environnement. La carte obtenue peut être simultanément utilisée pour planifier des déplacements vers un but arbitraire. L'exécution du plan est contrôlé en permanence afin de vérifier son exécution correcte et le plan peut être repris en cas de problème pour planifier un détour vers le but.

Notre modèle présente certaines limitations lorsqu'il est utilisé sur un robot réel, essentiellement dues au bruit important auquel sont soumis les capteurs que nous utilisons. Le fort *perceptual aliasing* engendré par ce bruit entraîne, en effet, des difficultés d'estimation de la position dans certaines zones de l'environnement. Ces problèmes seraient certainement résolus par la mise en place de procédures de navigation actives qui permettraient de rejoindre des zones dans lesquelles l'estimation de la position fonctionnerait de manière correcte.

Enfin, notre modèle fournit une base solide pour la mise au point de méthodes de planification qui permettent, par exemple, la gestion de différents buts simultanés ou l'intégration de pénalités pour la traversée de certaines zones de l'environnement. Il suggère également quelques hypothèses nouvelles sur les capacités de localisation dont font preuve les rats.

Annexe A

Paramètres de notre modèle

Nous résumons dans cette annexe les valeurs des différents paramètres de notre modèle. Ces paramètres ont été utilisés sans modifications pour toutes les expériences décrites dans ce rapport.

Diamètre du robot : 500 mm

`Seuil_Localisation` : 500 mm

Nombre de pas de temps passé pris en compte : 5 *pas de temps*

`Seuil_Variance` : 10^6 mm^2

`Seuil_Reconnaissance` : 100 mm

`Seuil_Mise_A_Jour` : 1000 mm

`Seuil_Temps_Oubli` : 2000 *pas de temps*

`Seuil_Visites_Oubli` : 5 *visites*

`Nombre_Connexions_Max` : 5 *connexions*

`Seuil_Exploration` : 4 *noeuds*

`Seuil_Planification` : 800 mm

Bibliographie

[ActivMedia, 1999] ActivMedia (1999). *Pioneer 2 Mobile Robot. Saphira Manual.*

[Aloimonos, 1993] Aloimonos, Y. (1993). *Active Perception.* LEA.

[Arkin, 1989] Arkin, R. (1989). Towards the unification of navigational planning and reactive control. In *Proceedings of the AAAI Spring Symposium on Robot Navigation*, pages 1–5.

[Arléo, 2000] Arléo, A. (2000). *Spatial Learning and Navigation in Neuro-Mimetic Systems.* PhD thesis, Ecole Polytechnique Fédérale de Lausanne.

[Arleo et al., 1999] Arleo, A., del R. Millán, J., and Floreano, D. (1999). Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation. In *IEEE Transactions on Robotics and Automation*, volume 15, pages 990–1000.

[Arleo and Gerstner, 2000] Arleo, A. and Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation : A model of hippocampal place cell activity. *Biological Cybernetics, Special Issue on Navigation in Biological and Artificial Systems*, 83 :287–299.

[Arsenio and Ribeiro, 1998] Arsenio, A. and Ribeiro, M. I. (1998). Absolute localization of mobile robots using natural landmarks. In *Proceedings of the International Conference on Electronics, Circuits and Systems.*

[Ayache and Faugeras, 1989] Ayache, N. and Faugeras, O. (1989). Maintaning representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6) :804 – 819.

[Bachelder and Waxman, 1994] Bachelder, I. A. and Waxman, A. M. (1994). Mobile robot visual mapping and localization : A view-based neurocomputational architecture that emulates hippocampal place learning. *Neural Networks*, 7(6/7) :1083–1099.

[Bachelder and Waxman, 1995] Bachelder, I. A. and Waxman, A. M. (1995). A view-based neurocomputational system for relational map-making and navigation in visual environments. *Robotics and Autonomous Systems*, 16 :267–298.

[Balakrishnan et al., 1999] Balakrishnan, K., Bousquet, O., and Honavar, V. (1999). Spatial learning and localization in rodents : A computation model of the hippocampus and its implications for mobile robots. *Adaptive Behavior*, 7(2) :173–216.

[Betke and Gurvits, 1994] Betke, M. and Gurvits, K. (1994). Mobile robot localization using landmarks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-94)*, volume 2, pages 135–142.

[Boley et al., 1996] Boley, D., Steinmetz, E., and Sutherland, K. (1996). Robot localization from landmarks using recursive total least squares. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-96)*, volume 4, pages 1381–1386.

[Borghi and Brugali, 1995] Borghi, G. and Brugali, D. (1995). Autonomous map learning for a multi-sensor mobile robot using diktiometric representation and negotiation mechanism. In *Proceedings of the International Conference on Advanced Robotics (ICAR-95).*

[Braitenberg, 1984] Braitenberg, V. (1984). *Vehicles.* MIT Press, Cambridge MA.

[Brooks, 1991] Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, (47) :139–159.

[Buhmann et al., 1995] Buhmann, J., Burgard, W., Cremers, A. B., Fox, D., Hofmann, T., Schneider, F., Strikos, J., and Thrun, S. (1995). The mobile robot rhino. *AI Magazine*, 16(1).

[Burgard et al., 1998] Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S. (1998). The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98).* The MIT Press.

[Burgard et al., 1996] Burgard, W., Fox, D., Hennig, D., and Schmidt, T. (1996). Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 896–901.

[Burgard et al., 1999] Burgard, W., Fox, D., Jans, H., Matenar, C., and Thrun, S. (1999). Sonar-based mapping of large-scale mobile robot environments using EM. In *Proceeding ofthe International Conference on Machine Learning (ICML-99).*

[Burgess et al., 1994] Burgess, N., Recce, M., and O'Keefe, J. (1994). A model of hippocampal function. *Neural Networks*, 7 :1065–1081.

[Cartwright and Collett, 1987] Cartwright, B. A. and Collett, T. S. (1987). Landmark maps for honeybees. *Biol. Cybern.*, 57 :85–93.

[Cassandra et al., 1996] Cassandra, A. R., Kaelbling, L. P., and Kurien, J. A. (1996). Acting under uncertainty : Discrete bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[Castellanos et al., 1999] Castellanos, J. A., Montiel, J. M. M., Neira, J., , and Tardos, J. D. (1999). The SPmap : A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5) :948–953.

[Chatila and Laumond, 1985] Chatila, R. and Laumond, J. (1985). Position referencing and consistent world modelling for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-85)*, pages 138–170.

[Cox, 1991] Cox, I. J. (1991). Blanche - an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2) :193–204.

[Dedeoglu et al., 1999] Dedeoglu, G., Mataric, M., and Sukhatme, G. S. (1999). Incremental, online topological map building with a mobile robot. In *Proceedings of Mobile Robots XIV - SPIE*, pages 129–139.

[Donnart and Meyer, 1996] Donnart, J. Y. and Meyer, J. A. (1996). Spatial exploration, map learning, and self-positioning with monalysa. In Maes, P., Mataric, M., Meyer, J. A., Pollack, J., and Wilson, S. W., editors, *From animals to animats 4. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96).* The MIT Press.

[Duckett et al., 2000] Duckett, T., Marsland, S., and Shapiro, J. (2000). Learning globally consistent maps by relaxation. In *Proceedings of the International Conference on Robotics and Automation (ICRA'2000)*, pages 3841 – 3846.

[Duckett and Nehmzow, 1997] Duckett, T. and Nehmzow, U. (1997). Experiments in evidence based localisation for a mobile robot. In Corne, D. and Shapiro, J. L., editors, *Proceedings of the AISB 97 workshop on Spatial Reasoning in Animals and Robots.* Springer.

[Duckett and Nehmzow, 1998] Duckett, T. and Nehmzow, U. (1998). Mobile robot self-localization and measurement of performance in middle scale environments. *Robotics and Autonomous Systems*, 1-2(24).

[Dudek and Jenkin, 2000] Dudek, G. and Jenkin, M. (2000). *Computational Principles of Mobile Robotics.* Cambridge University Press.

[Dudek and MacKenzie, 1993] Dudek, G. and MacKenzie, P. (1993). Model-based map construction for robot localization. In *Proceedings of Vision Interface 1993.*

[Egerton and Callaghan, 2000] Egerton, S. and Callaghan, V. (2000). From mammals to machines : Towards a biologically inspired mapping model for autonomous mobile robots. In *Procceding of the 6th International Conference on Intelligent Autonomous Systems (IAS-6).*

[Einsele, 1997] Einsele, T. (1997). Real-time self-localization in unknown indoor environments using a panorama laser range finder. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-97)*, pages 697–703.

[Engelson, 1995] Engelson, S. P. (1995). Continuous map learning for mobile robots. Extended Abstract for the 3rd French-Israeli Symposium on Robotics.

[Engelson and McDermott, 1992] Engelson, S. P. and McDermott, D. V. (1992). Error correction in mobile robot map learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-92).*

[Feder et al., 1999] Feder, H., Leonard, J., and Smith, C. (1999). Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7) :650–668.

[Filliat and Meyer, 2001a] Filliat, D. and Meyer, J.-A. (2001a). Map-based navigation in mobile robots - i. a review of localisation strategies. *Journal of Cognitive Systems Research, submitted for publication.*

[Filliat and Meyer, 2001b] Filliat, D. and Meyer, J.-A. (2001b). Map-based navigation in mobile robots - ii. a review of map learning and path planing strategies. *Journal of Cognitive Systems Research, submitted for publication.*

[Fox et al., 1999] Fox, D., Burgard, W., Dellaert, F., and Thrun, S. (1999). Monte carlo localization : Efficient position estimation for mobile robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99).*

[Fox et al., 1998a] Fox, D., Burgard, W., and Thrun, S. (1998a). Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25 :195–207.

[Fox et al., 1998b] Fox, D., Burgard, W., Thrun, S., and Cremers, A. B. (1998b). Position estimation for mobile robots in dynamic environments. In *Proceedings of the Fifteenth National Conference on Articial Intelligence (AAAI-98)*, pages 983–988.

[Franz et al., 1998] Franz, M., Scholkopf, B., Georg, P., Mallot, H., and Bulthoff, H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, 5 :111–125.

[Gasós and Martín, 1997] Gasós, J. and Martín, A. (1997). Mobile robot localization using fuzzy maps. In Martin, T. and Ralescu, A., editors, *Fuzzy Logic in AI - Selected papers from the IJCAI '95 Workshop*, number 1188, pages 207–224. Springer-Verlag.

[Gaussier et al., 2000] Gaussier, P., Joulain, C., Banquet, J., Lepretre, S., and Revel, A. (2000). The visual homing problem : an example of robotics/biology cross-fertilisation. *Robotics and autonomous systems*, 30(1-2) :155–180.

[Gaussier et al., 1998] Gaussier, P., Leprêtre, S., Joulain, C., Revel, A., Quoy, M., and P., B. J. (1998). Animal and robot learning : experiments and models about visual navigation. In *Proceedings of the Seventh European Workshop on Learning Robots.*

[Georgopoulos et al., 1986] Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, (233) :1416–1419.

[Gomes-Mota and Ribeiro, 2000] Gomes-Mota, J. and Ribeiro, M. I. (2000). Mobile robot localisation on reconstructed 3d models. *Robotics and Autonomous Systems*, 31(1-2) :17–30.

[Gourichon and Meyer, 2001] Gourichon, S. and Meyer, J.-A. (2001). Using colored snapshots for short-range guidance in mobile robots. *International Journal of Robotics and Automation, submitted for publication.*

[Greiner and Isukapalli, 1996] Greiner, R. and Isukapalli, R. (1996). Learning to select useful landmarks. *IEEE Transactions on Systems, Man, and Cybernetics-Part B,Special Issue on Learning Autonomous Robots*, 26(3).

[Guibas et al., 1995] Guibas, L. J., Motwani, R., and Raghavan, P. (1995). The robot localization problem. *Algorithmic Foundations of Robotics*, pages 269–282.

[Gutmann and Konolige, 2000] Gutmann, J. and Konolige, K. (2000). Incremental mapping of large cyclic environments. In *Proceedingsof the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA-2000).*

[Gutmann and Schlegel, 1996] Gutmann, J. S. and Schlegel, C. (1996). Amos : Comparison of scan matching approaches for self-localization in indoor environments,. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press.

[Hafner, 2000] Hafner, V. V. (2000). Learning places in newly explored environments. In Meyer, J. A., Berthoz, A., Floreano, D., Roiblat, H. L., and Wilson, S. W., editors, *Sixth International Conference on simulation of adaptive behavior : From Animals to Animats (SAB-2000). Proceedings Supplement.*, pages 111–120. ISAB.

[Hébert et al., 1996] Hébert, P., Betgé-Brezetz, S., and Chatila, R. (1996). Decoupling odometry and exteroceptive perception in building a global world map of a mobile robot : The use of local maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-1996)*, pages 757–764.

[Hertzberg and Kirchner, 1996] Hertzberg, J. and Kirchner, F. (1996). Landmark-based autonomous navigation in sewerage pipes. In *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press.

[Hughes and Murphy, 1992] Hughes, K. and Murphy, R. R. (1992). Ultrasonic robot localization using dempster-shafer theory. In *SPIE Stochastic Methods in Signal Processing, Image Processing, and Computer Vision, invited session on Applications for Vision and Robotics.*

[Jensfelt and Kristensen, 1999] Jensfelt, P. and Kristensen, S. (1999). Active global localisation for a mobile robot using multiple hypothesis tracking. In *Proceedings of the IJCAI-99 Workshop on Reasoning with Uncertainty in Robot Navigation.*

[Kaelbling et al., 1998] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101 :99–134.

[Karch and Wahl, 1999] Karch, O. and Wahl, T. (1999). Relocalization – theory and practice. *Discrete Applied Mathematics : Special Issue on Computational Geometry*, 93.

[Kirsh, 1991] Kirsh, D. (1991). Today the earwig, tomorrow man ? *Artificial Intelligence*, 47 :161–184.

[Koren and Borenstein, 1991] Koren, Y. and Borenstein, J. (1991). Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transaction on Robotics and Automation*, 7(4) :535–539.

[Kortenkamp et al., 1994] Kortenkamp, D., Huber, M., Koss, F., Belding, W., Lee, J., Wu, A., Bidlack, C., and Rogers, S. (1994). Mobile robot exploration and navigation of indoor spaces using sonar and vision. In *Proceedings of the AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space (CIRFFSS 94)*, pages 509–519.

[Kortenkamp and Weymouth, 1994] Kortenkamp, D. and Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 979–984, Seattle, WA.

[Kuipers, 2000] Kuipers, B. J. (2000). The spatial semantic hierarchy. *Artificial Intelligence*, (119) :191–233.

[Kuipers and Byun, 1991] Kuipers, B. J. and Byun, Y. T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8 :47–63.

[Kunz et al., 1997] Kunz, C., Willeke, T., and Nourbakhsh, I. (1997). Automatic mapping of dynamic office environments. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-97)*, volume 2, pages 1681–1687.

[Kurz, 1995] Kurz, A. (1995). Alef : An autonomous vehicle which learns basic skills and construct maps for navigation. *Robotics and Autonomous Systems*, 14 :172–183.

[Lambrinos et al., 2000] Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., and Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems, special issue : Biomimetic Robots*, 30 :39–64.

[Latombe, 1991] Latombe, J.-C. (1991). *Robot Motion Planning.* Boston : Kluwer Academic Publishers.

[Laumond, 1998] Laumond, J.-P. (1998). *Robot Motion Planning and Control.* Lectures Notes in Control and Information Sciences 229. Springer.

[Leonard et al., 1992] Leonard, J. J., Durrant-Whyte, H. F., and Cox, I. J. (1992). Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4) :89–96.

[Levitt and Lawton, 1990] Levitt, T. S. and Lawton, D. T. (1990). Qualitative navigation for mobile robots. *Artificial Intelligence*, 44 :305–360.

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

[Luo and Kay, 1989] Luo, R. C. and Kay, M. G. (1989). Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5) :901–931.

[Madsen et al., 1997] Madsen, C., Andersen, C., and rensen, J. (1997). A robustness analysis of triangulation-based robot self-positioning. In *Proceedings of the 5th Symposium for Intelligent Robotics Systems.*

[Marr, 1971] Marr, D. (1971). Simple memory : A theory for archicortex. *Philosophical transactions of the Royal Society of London*, 176 :23–81.

[Mataric, 1992] Mataric, M. J. (1992). Integration of representation into goal-driven behaviour-based robots. *IEEE Transactions on Robotics and Automation*, 8(3) :304–312.

[Maybeck, 1979] Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control.* Academic Press.

[McLachlan and Krishnan, 1997] McLachlan, G. J. and Krishnan, T. (1997). *The EM Algorithm an Extensions.* Wiley.

[Meyer, 1995] Meyer, J. A. (1995). The animat approach to cognitive science. In Roitblat, H. and Meyer, J. A., editors, *Comparative Approaches to Cognitive Science*. MIT Press.

[Meyer, 1996] Meyer, J. A. (1996). Artificial life and the animat approach to artificial intelligence. In Boden, M., editor, *Artificial Intelligence*. Academic Press.

[Meyer, 1997] Meyer, J. A. (1997). From natural to artificial life : Biomimetic mechanisms in animat designs. *Robotics and Autonomous Systems*, 22 :3–21.

[Moravec and Elfes, 1985] Moravec, H. and Elfes, A. (1985). High resolution maps from wide angular sensors. In *Proceedings of the IEEE International Conference On Robotics and Automation (ICRA-85)*. IEEE Computer Society Press.

[Moutarlier and Chatila, 1990] Moutarlier, P. and Chatila, R. (1990). An experimental system for incremental environment modeling by an autonomous mobile robot. In *Experimental Robotics 1*, pages 327–346. Springer-Verlag.

[Muller and Kubie, 1987] Muller, R. U. and Kubie, J. (1987). The effects of changes in the environement on the spatial firing of hippocampal complex-spike cells. *Journal of Neurosciences*, 7(7) :1951–68.

[Murphy, 2000] Murphy, R. R. (2000). *Introduction to AI Robotics*. The MIT Press.

[Nehmzow and Owen, 2000] Nehmzow, U. and Owen, C. (2000). Robot navigation in the real world : Experiments with manchester's fortytwo in unmodified, large environments,. *Robotics and Autonomous Systems*, 33(4) :223–242.

[Nourbakhsh et al., 1995] Nourbakhsh, I., Powers, R., and Birchfield, S. (1995). Dervish, an office navigating robot. *AI Magazine*, 16(2) :53–60.

[O'Keefe and Conway, 1978] O'Keefe, J. and Conway, D. (1978). Hippocampal place units in the freely moving rat : Why they fire where they fire. *Experimental Brain Research*, 31 :573–590.

[O'Keefe and Dostrovsky, 1971] O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. preliminary evidence from unit activity in the freely moving rat. *Experimental Brain Research*, 34 :171–175.

[Olson, 2000] Olson, C. F. (2000). Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1).

[Oore et al., 1997] Oore, S., Hinton, G., and Dudek, G. (1997). A mobile robot that learns its place. *Neural Computation*, 9 :683–699.

[Piasecki, 1995] Piasecki, M. (1995). Global localization for mobile robots by multiple hypothesis tracking. *Robotics and Autonomous Systems*, 16 :93–104.

[Prescott, 1995] Prescott, T. J. (1995). Spatial representation for navigation in animats. *Adaptive Behavior*, 4(2).

[Quirk et al., 1990] Quirk, G. J., Muller, R. U., and Kubie, J. L. (1990). The firing of hippocampal place cells in the dark depends on the rat's recent experience. *Journal of Neuroscience*, 10(6) :2008–2017.

[Radhakrishnan and Nourbakhsh, 1999] Radhakrishnan, D. and Nourbakhsh, I. (1999). Topological localization by training a vision-based transition detector. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-99)*.

[Saffiotti and Wesley, 1995] Saffiotti, A. and Wesley, L. P. (1995). Perception-based self-localization using fuzzy locations. In *Reasoning with Uncertainty in Robotics*, volume 1093 of *Lecture Notes in Computer Science*. Springer-Verlag.

[Schiele and Crowley, 1994] Schiele, B. and Crowley, J. (1994). A comparison of position estimation techniques using occupancy grids. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-94)*, pages 1628–1634.

[Scholkopf and Mallot, 1995] Scholkopf, B. and Mallot, H. A. (1995). View-based cognitive mapping and path planning. *Adaptive Behavior*, 3(3) :311–348.

[Schoppers, 1987] Schoppers, M. J. (1987). Universal plans for reactive robots in unpredictable environments. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI 87)*, pages 1039–1046, Milan, Italy.

[Schultz and Adams, 1998] Schultz, A. C. and Adams, W. (1998). Continuous localization using evidence grids. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-98)*, pages 2833–2839.

[Sharp, 1991] Sharp, P. E. (1991). Computer simulation of hippocampal place cells. *Psychobiology*, 19(2) :103–115.

[Shatkay and Kaelbling, 1997] Shatkay, H. and Kaelbling, L. P. (1997). Learning topological maps with weak local odometric information. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*.

[Sim and Dudek, 1999] Sim, R. and Dudek, G. (1999). Learning visual landmarks for pose estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-1999)*.

[Simmons and Koenig, 1995] Simmons, R. and Koenig, S. (1995). Probabilistic navigation in partially observable environments. In Mellish, S., editor, *Proccedings of IJCAI'95*. Morgan Kaufman Publishing.

[Simmons and Koenig, 1996] Simmons, R. and Koenig, S. (1996). Unsupervised learning of probabilistic models for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-1996)*, pages 2301–2308.

[Smith et al., 1988] Smith, R., Self, M., and Cheeseman, P. (1988). Estimating uncertain spatial relationships in robotics. In Lemmer, J. F. and Kanal, L. N., editors, *Uncertainty in Artificial Intelligence*, pages 435–461. Elsevier.

[Theocharous et al., 2001] Theocharous, G., Rohanimanesh, K., and Mahadevan, S. (2001). Learning hierarchical partially observable markov decision processes for robot navigation. In *Proceedings of the IEEE Conference on Robotics and Automation*.

[Thrun, 1999] Thrun, S. (1999). Learning metric-topological maps maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1) :21–71.

[Thrun, 2000] Thrun, S. (2000). Probabilistic algorithms in robotics. *AI Magazine*, 21(4) :93–109.

[Thrun et al., 1999] Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Haehnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). Minerva : A second generation mobile tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-1999)*.

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

[Thrun et al., 1998] Thrun, S., Fox, D., and Burgard, W. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31 :29–53. joint issue with Autonomous Robots 5.

[Tolman, 1948] Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55 :189–208.

[Touretzky et al., 1994] Touretzky, D. S., Wan, H. S., and Redish, A. D. (1994). Neural representations of space in rats and robots. In Zurada, J. M., Marks, R. J., and Robinson, C. J., editors, *Computational Intelligence : Imitating Life*, pages 57–68. IEEE Press.

[Trullier and Meyer, 1997] Trullier, O. and Meyer, J. A. (1997). Biomimetic navigation models and strategies in animats. *AI Communications*, 10 :79–92.

[Trullier and Meyer, 2000] Trullier, O. and Meyer, J. A. (2000). Animat navigation using a cognitive graph. *Biological Cybernetics*, 83(3) :271–285.

[Trullier et al., 1997] Trullier, O., Wiener, S., Berthoz, A., and Meyer, J. A. (1997). Biologically-based artificial navigation systems : Review and prospects. *Progress in Neurobiology*, 51 :483–544.

[Ulrich and Nourbakhsh, 2000] Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2000)*.

[Von Wichert, 1998] Von Wichert, G. (1998). Mobile robot localization using a self-organised visual environment representation. *Robotics and Autonomous Systems*, 25 :185–194.

[Wijk and Christensen, 2000] Wijk, O. and Christensen, H. I. (2000). Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data. *Robotics and Autonomous Systems*, 31(1-2) :31–42.

[Yamauchi and Beer, 1996] Yamauchi, B. and Beer, R. (1996). Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part B,Special Issue on Learning Autonomous Robots*, 26(3) :496–505.

[Yamauchi and Langley, 1997] Yamauchi, B. and Langley, P. (1997). Place recognition in dynamic environments. *Journal of Robotic Systems, Special Issue on Mobile Robots*, 14(2) :107–120.

[Yamauchi et al., 1999] Yamauchi, B., Schultz, A., and Adams, W. (1999). Integrating exploration and localization for mobile robots. *Adaptive Behavior*, 7(2) :217–230.

[Zemel et al., 1997] Zemel, R. S., Dayan, P., and Pouget, A. (1997). Probabilistic interpretation of population codes. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9, page 676. The MIT Press.

[Zhang et al., 1998] Zhang, K., Ginzburg, I., McNaughton, B., and Sejnowski, T. (1998). Interpreting neuronal population activity by reconstruction : A unified framework with application to hippocampal place cells. *Journal of Neurophysiology*, 79 :1017–1044.