Contributions aux méthodes de détection visuelle de fermeture de boucle et de segmentation topologique de l'environnement.

Chapoulie Alexandre

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THESE

pour l'obtention du grade de

Docteur en Sciences
de l'Université de Nice-Sophia Antipolis

Spécialité Automatique et Traitement des Signaux et des Images

présentée et soutenue par

Alexandre CHAPOULIE

Contributions aux méthodes de détection visuelle de fermeture de boucle et de segmentation topologique de l'environnement.

Thèse dirigée par Patrick RIVES
soutenue le 10 Décembre 2012

Jury :

M.	**Philippe**	**Tarroux**	Rapporteur
M.	**Pascal**	**Vasseur**	Rapporteur
Mme.	**Eva**	**Crück**	Examinatrice
M.	**Jonathan**	**Courbon**	Examinateur
M.	**Rachid**	**Deriche**	Examinateur
M.	**David**	**Filliat**	Co-directeur de thèse
M.	**Patrick**	**Rives**	Directeur de thèse

À ma maman,
À ma sœur,
À mes frères.

Remerciements

Le doctorat dans le domaine du traitement du signal et des images dans le cadre d'applications robotiques a été pour moi l'opportunité de satisfaire mon désir d'apprendre toujours plus dans divers domaines scientifiques. Venant du domaine de l'électronique, j'ai débuté une thèse qui ne correspondait en rien à mes domaines de compétences. Le challenge était ardu mais l'expérience valait la peine d'être vécue. L'enrichissement scientifique et l'opportunité de participer activement à l'innovation confortent alors mon désir d'aller encore plus loin et de réaliser de nouveaux projets.

Je tiens tout d'abord à remercier Messieurs Philippe Tarroux et Pascal Vasseur pour leur travail de rapporteurs. Je souhaite aussi remercier Madame Eva Crück, Monsieur Jonathan Courbon et Monsieur Rachid Deriche pour avoir examiné mon manuscrit de thèse.

Je souhaite aussi remercier la DGA pour avoir financé mes travaux de recherche durant ma thèse.

Mes remerciements vont ensuite à mes encadrants de thèse : Monsieur Patrick Rives pour avoir permis la réalisation de cette thèse à l'INRIA Sophia Antipolis, pour ses conseils avisés, et critiques, quant à l'élaboration d'idées toujours nouvelles dans un souci de qualité et de vision d'ensemble, et pour sa constante bonne humeur créant un cadre de travail agréable ; Monsieur David Filliat pour ses conseils et idées pour l'amélioration des algorithmes élaborés mais aussi pour ses relectures minutieuses des articles et du manuscrit m'ayant permis d'améliorer continuellement la présentation de mes travaux.

Dans le cadre de mes activités de remise en service, de mise à jour et d'instrumentation du Cycab, je tiens à remercier Messieurs Nicolas Chleq, Erwan Demairy, Patrick Pollet et Fabien Spindler pour leur précieuse aide. Je tiens à remercier plus particulièrement Madame Soraya Arias pour son aide inestimable apportée dans ce projet.

J'aimerais aussi remercier tous mes collègues de travail au sein de l'équipe Arobas : Pascal, Claude, Ezio, Nathalie, Cyril, Thomas, Gabriella, Minh Duc, Stefan, Maxime, Mathieu, Claire, Glauco, Daniele, Luca, Adan, Wladyslaw et plus particulièrement Mélaine pour ses discussions intéressantes. Je voudrais aussi remercier nos deux nouveaux doctorants, Tawsif et Romain, pour leur amitié. Un remerciement particulier pour Tiago avec qui j'ai passé d'excellents moments, partagé des discussions enrichissantes et plus que diverses, et planifié des projets. J'ai également apprécié son soutien amical dans les périodes difficiles de la thèse.

Enfin, mes derniers remerciements, les plus chaleureux, vont à ma famille qui m'a toujours soutenu, voire parfois supporté. Elle a toujours été présente que ce soit dans les bons moments comme dans les mauvais. Plus spécialement, je souhaite remercier ma maman qui a toujours su m'encourager à m'investir dans mes projets. Je souhaite aussi la remercier puisqu'elle a même accepté de relire mon manuscrit de thèse.

I have not failed 700 times. I have not failed once. I have succeeded in proving that those 700 ways will not work. When I have eliminated the ways that will not work, I will find the way that will work.

Thomas Edison, about creating the light bulb.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

1 Introduction

Si les assistants mécaniques et golems apparaissent dans la mythologie, les premiers automates sont réalisés par Héron d'Alexandrie au I^{er} siècle. Viennent ensuite les inventions notamment de Léonard de Vinci durant le XVI^{ième} siècle. La robotique moderne commence réellement au début du XX^{ième} siècle avec la création de répliques plus ou moins réussies d'animaux existants. Par exemple, Hammond et Miessner réalisent un chien électrique en 1915. Parallèlement, la robotique se développe au sein des récits de science fiction. Le terme « robot » est employé pour la première fois par l'écrivain tchèque Karel Čapek pour un spectacle créé en 1921. Le mot « robot » est un mot issu des langues slaves et signifie esclave ou encore travailleur dévoué. L'écrivain américain d'origine russe Issac Asimov démocratise les robots dans ses nombreuses œuvres de science fiction. Il emploiera le terme « robotique » pour la première fois dans son œuvre « Menteur ! » publiée en mai 1941.

La robotique se concrétise réellement suite à la seconde guerre mondiale et notamment à partir des années 1960 dans le milieu industriel. Originellement conçus pour intervenir dans les milieux à risques tels que le nucléaire ou la forte corrosion, les robots ont pris de plus en plus d'importance dans les usines. Ils travaillent en permanence, rapidement, avec une précision élevée, dans les milieux dangereux et font très peu d'erreurs. D'abord dans l'industrie automobile puis dans l'ensemble de l'industrie, les robots remplacent alors les ouvriers. La robotique mobile, quant à elle, connait un développement plus long. Bien que non autonome, le Goliath est une mine téléguidée apparue pendant la seconde guerre mondiale. La robotique mobile autonome est bien plus complexe à réaliser que les robots industriels, elle constitue d'ailleurs toujours un domaine de recherche très actif. La différence provient du fait que la robotique industrielle fonctionne dans un environnement et des conditions définis et effectue un ensemble de tâches simples. La robotique mobile évolue dans des environnements et conditions très divers rendant l'exécution de tâches plus ardue. Dans certains cadres restreints, des applications

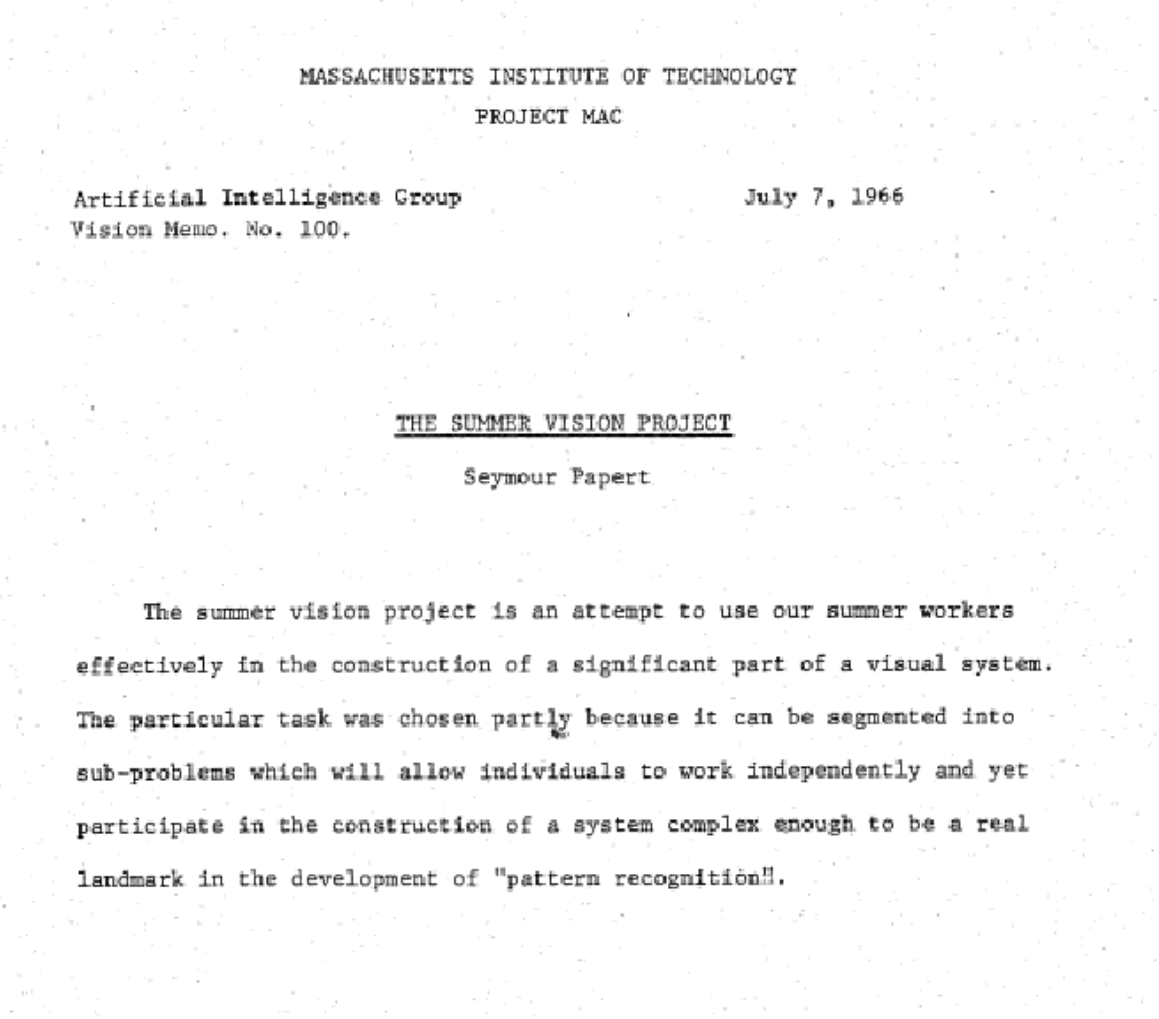
de robotique mobile présentent des résultats très convaincants. En industrie, des robots mobiles sont utilisés dans la gestion automatique d'entrepôt. En robotique domestique, les robots aspirateurs et les robots tondeuses sont d'excellents exemples d'applications efficaces. Récemment, la voiture automatique créée par Google étend ces concepts à des environnements beaucoup plus variés. Elle est capable de se déplacer de manière autonome dans un environnement urbain. Ce dernier exemple est, de nos jours, l'une des applications les plus abouties de la robotique mobile autonome.

Dans le cadre de la robotique mobile, étant donné que le robot évolue en permanence dans des milieux différents et sous des conditions différentes, il est nécessaire pour le robot d'analyser et d'interagir avec son environnement. De fait, le robot mobile doit accomplir deux tâches fondamentales : cartographier son environnement (ou posséder une connaissance a priori de celui-ci) et se localiser dedans. Les mécanismes sont formalisés dans les années 1980 par les auteurs [Smith et Cheeseman, 1986], [Smith *et al.*, 1987] et au début des années 1990 par [Leonard et Whyte, 1991] sous la problématique du SLAM, *i.e.* Simultaneous Localization And Mapping. Une fois le robot capable de gérer correctement ces deux tâches, il peut accomplir d'autres tâches plus ou moins complexes suivant ce pour quoi il est conçu.

Pour cartographier et se localiser, le robot doit être capable d'extraire de l'information de l'environnement. L'ensemble de capteurs utilisé présente donc une importance considérable. La capacité du robot à capter son environnement entre dans le domaine de la perception pour la robotique. Pendant longtemps, les capteurs de distance et d'angle (lasers, radars et sonars) ont été les seuls à être utilisés. Il s'agit toutefois de capteurs onéreux fournissant une information peu riche de l'environnement. La croissance de la puissance de calcul des processeurs, ainsi que l'apparition des processeurs graphiques, ont permis de supplanter les approches traditionnelles en permettant l'utilisation d'une simple caméra ou de systèmes de caméras, seules ou en conjonction avec les capteurs traditionnels. L'avantage des caméras est qu'elles sont des capteurs peu onéreux fournissant une information très riche de l'environnement. Le couplage de l'amélioration des processeurs à l'utilisation de caméras a mené à l'élaboration d'algorithmes performants en termes de robustesse et de temps de calcul.

L'utilisation de caméra pour percevoir l'environnement du robot entre dans le domaine de la vision par ordinateur. À l'origine, le problème de la vision par ordinateur a été abordé lors du Summer

Vision Project (*cf.* figure 1) en 1966. L'objectif était alors d'élaborer l'ensemble des mécanismes de vision permettant d'effectuer de la reconnaissance d'images. Cependant, la vision par ordinateur s'est révélée être un problème très ardu : il constitue aujourd'hui un domaine d'étude à part entière. La recherche dans ce domaine est très active et tente de résoudre notamment les problèmes suivants : analyse d'image (en termes d'information extraite et de contenu), reconnaissance d'image et classification.



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

FIG. 1 – The Summer Vision Project

Cette thèse, s'inscrivant dans le domaine de la perception visuelle pour la robotique, est donc connexe aux domaines de la robotique mobile et de la vision par ordinateur. Elle traite, dans le contexte du SLAM topologique, du problème de la détection visuelle de fermeture de boucle. Cet algorithme est crucial à la fois pour la localisation du robot et pour la cartographie. Toujours dans le cadre du

SLAM topologique, cette thèse traite aussi du problème de la segmentation de l'environnement en lieux topologiques. L'objectif est de créer automatiquement une représentation topologique significative de l'environnement dont les lieux sont définis par le processus de segmentation.

2 Objectifs

Le cadre général de cette thèse est le développement d'algorithmes purement visuels et exploitant au mieux la représentation sphérique de l'environnement. D'autre part, les méthodes développées ne reposent que sur de l'estimation d'information qualitative, aucune information métrique n'est donc calculée. Dans le contexte général du SLAM topologique les objectifs visés par cette thèse sont les suivants :

- Le premier algorithme développé concerne la détection visuelle robuste de fermeture de boucle. À partir de travaux présentant d'excellentes qualités en termes de temps de calcul et de robustesse à l'aliasing perceptuel, il s'agissait d'élaborer une méthode rendant la détection indépendante de l'orientation du robot. Cet ajout permet alors de fiabiliser la détection de fermeture de boucle en ne contraignant pas les conditions de prises de vue du robot lors de la revisite d'un même endroit.
- Le deuxième algorithme développé concerne la segmentation de l'environnement en lieux topologiques. Dans un premier temps, une définition générale du lieu topologique est donnée. Ensuite, en accord avec cette définition, deux algorithmes de détermination automatique des lieux topologiques ont été élaborés. Le premier est une preuve de concept permettant de démontrer la validité à la fois de la définition et de l'approche utilisée. Le deuxième algorithme est basé sur une méthode plus adéquate d'extraction d'information de l'environnement, liée à la définition du lieu topologique, et repose sur un algorithme de détection plus complet.

3 Contributions

Les travaux présentés dans ce manuscrit ont permis la publication de deux articles :

- Dans [Chapoulie *et al.*, 2011], nous présentons les travaux correspondant à la première partie du manuscrit. Cet article aborde le problème de la dépendance à l'orientation du robot dans le processus de détection visuelle de fermeture de boucle. À partir de travaux proposant une détection de fermeture de boucle calculable en temps-réel et robuste à l'aliasing perceptuel, nous ajoutons l'invariance à l'orientation du robot. Pour cela, nous utilisons une représentation sphérique ego-

centrée de l'environnement afin d'acquérir de l'information indépendamment du point de vue du robot. Nous élaborons aussi un descripteur sphérique global de répartition de l'information particulièrement adapté à la représentation utilisée. La représentation et le descripteur associé sont inclus dans un processus d'inférence bayésienne de détection de fermeture de boucle. La robustesse à l'aliasing perceptuel et surtout l'invariance à l'orientation du robot nous permettent de nous affranchir d'une phase de vérification de la consistance de la fermeture de boucle par géométrie épipolaire ; le mécanisme repose ainsi uniquement sur un processus décisionnel robuste. Des expérimentations sur une trajectoire de 1.5 km en environnement extérieur montrent l'efficacité de l'algorithme temps-réel proposé.

– Dans [Chapoulie *et al.*, 2012], nous présentons les travaux correspondant partiellement à la seconde partie du manuscrit (algorithme utilisant le GIST). Nous abordons dans cet article le problème de la segmentation de l'environnement en lieux topologiques. Nous commençons par donner une définition originale du lieu topologique utilisable pour les environnements d'intérieur et d'extérieur : un lieu topologique est un lieu dont les paramètres structurels sont suffisamment similaires à eux-mêmes. Notre recherche de segmentation de l'environnement repose sur la recherche de la variation de ces paramètres structurels. Pour cela, nous extrayons l'information structurelle de l'environnement grâce au descripteur global GIST que nous avons adapté à la représentation sphérique. Le descripteur obtenu, de très faible dimension, caractérise les fréquences et orientations de la scène courante. L'évolution de ce descripteur est suivie par un processus de détection de rupture de modèle. Ainsi, lorsqu'un changement trop important du descripteur GIST est détecté, un changement de lieu est défini ; les lieux topologiques étant alors définis entre deux ruptures. L'ensemble du processus est calculable en temps-réel. L'algorithme a été testé dans des environnements d'intérieur et d'extérieur. Les résultats sont très intéressants et convaincants quant à l'aspect structurel d'un lieu topologique.

4 Structure du manuscrit

Étant donné que deux algorithmes distincts ont été développés durant cette thèse, le manuscrit est divisé en deux parties :

La première partie de ce manuscrit traite de la détection visuelle de fermeture de boucle. Un

chapitre d'état de l'art met en évidence son importance au sein des algorithmes de SLAM et constitue un bilan des méthodes actuelles. Les points forts et les limitations sont alors mis en exergue pour déterminer les axes d'amélioration de ces algorithmes, et notamment en ce qui concerne l'indépendance à l'orientation du robot. Le chapitre suivant décrit le modèle de représentation sphérique utilisé. Les méthodes de description de l'information contenue dans l'image ainsi que le modèle du « sac de mots visuels » sont aussi abordés. Dans un troisième chapitre, une méthode de détection de fermeture de boucle efficace et particulièrement robuste à l'aliasing perceptuel est présentée. L'introduction du modèle de représentation sphérique dans cette méthode permet de rendre la détection de fermeture de boucle indépendante à l'orientation du robot. Les mécanismes modifiés et l'élaboration du descripteur sphérique global sont alors exposés. Les résultats fournis dans le dernier chapitre mettent en exergue les capacités de l'algorithme développé. Une analyse des capacités et performances de l'algorithme est alors effectuée.

La deuxième partie du manuscrit se concentre sur la segmentation de l'environnement pour la cartographie topologique. Le premier chapitre constitue un état de l'art des méthodes actuelles de segmentation de l'environnement. Une analyse des avantages et inconvénients de ces méthodes permet de mettre en avant les limitations notamment en termes de définition et d'expérimentation en environnement extérieur. Dans le chapitre suivant, une définition originale du lieu topologique est donnée. Cette définition reposant sur la description des paramètres structurels de l'environnement, deux descripteurs de structure sont présentés : le GIST et le spectre d'harmoniques sphériques. Le troisième chapitre détaille un algorithme de détection de rupture de modèle. L'objectif est alors de détecter les variations significatives dans la structure de l'environnement pour créer la segmentation topologique. Une première approche avec des hypothèses fortes permet de faire une preuve de concept de la définition élaborée et de la méthode développée en conséquence. Une seconde méthode corrigeant les défauts de la première est ensuite expliquée. Le dernier chapitre présente les résultats obtenus avec les deux méthodes développées. Les expérimentations sont effectuées dans un environnement d'intérieur et dans un environnement d'extérieur. Une analyse critique des résultats est présentée afin d'établir les forces et faiblesses des algorithmes développés.

Une conclusion générale présente un récapitulatif des résultats obtenus pour chacun des algorithmes. Les perspectives et axes de recherches que présentent nos travaux sont ensuite discutés.

Première partie

Détection visuelle de fermeture de boucle

Chapitre 1

État de l'art

Sommaire

1.1 Introduction

Origines historiques du problème

Le problème de la détection de fermeture de boucle apparaît dans le contexte de la localisation et cartographie simultanées (*Simultaneous Localisation And Mapping*, SLAM). Historiquement, il s'agissait alors de l'étude de la construction consistante de cartes dans des environnements dits « cycliques ». Les travaux de [Gutmann et Konolige, 1999] récapitulent les méthodes existantes de SLAM (jusqu'en 1999) et les problématiques liées. Notamment, ils étudient le problème de la création de cartes consistantes dans de grands environnements présentant des cycles. Ils introduisent alors pour la première fois le terme de « fermeture de boucle ». Ces travaux sont présentés ci-après. Lors du déplacement du robot, celui-ci estime la carte locale de l'environnement proche. Quand il rejoint un endroit déjà visité, il y a correspondance entre la carte locale courante et une carte précédemment établie. La carte globale de l'environnement présente alors un cycle. D'un point de vue plus abstrait, cela revient à étudier les cycles, ou boucles, possibles dans un environnement et ainsi déterminer les chemins permettant de revenir à une position précédente. De nombreuses approches ([Chatila et Laumond, 1985], [Durrant-Whyte, 1986], [Hébert *et al.*, 1995], [Smith *et al.*, 1987] et [Thrun *et al.*, 1998]) ont étudié le problème des environnements cycliques en utilisant des méthodes par filtrage de Kalman imposant une hypothèse markovienne de l'estimation. Cette dernière implique que l'estimation de l'état suivant, *i.e.* la carte locale et la pose du robot dans la carte, ne dépend que de l'état à l'instant précédent rendant le processus d'estimation incrémental. Toutefois, dans ce contexte d'estimation simultanée de la carte et de la pose, l'hypothèse markovienne n'est plus valable du fait des interdépendances entre les observations de la carte globale, la carte locale et les différentes poses du robot. Si la carte globale consiste en un large ensemble de caractéristiques M. À chaque instant, le scan du robot s_n ne se rapporte qu'à un petit sous-ensemble $a_n \subset M$. Le problème ne peut être alors réduit à l'estimation de la pose et au sous-ensemble a_n seuls car les scans précédents ont éventuellement lié a_n à d'autres sous-ensembles de M. Les approches précitées, si elles fournissent des solutions correctes pour l'estimation dans de petits environnements, n'offrent que des résultats médiocres quant à l'estimation dans de grands environnements et notamment lors de l'estimation dans des environnements cycliques. Les auteurs dans [Lu et Milios, 1997] et [Lu et Milios, 1994] présentent la première méthode de construction de cartes d'environnements cycliques ayant des résultats globalement fiables. Le principe de la méthode repose sur une optimisation globale de l'ensemble, c'est-à-dire à la fois la carte globale et l'ensemble des poses

du robot. Étant donné que les poses du robot sont considérées comme un ensemble, une estimation de pose consistante peut fonctionner dans les environnements cycliques. Les auteurs [Gutmann et Konolige, 1999] mettent en exergue les limitations de la méthode d'optimisation globale proposée par Lu et Milios. Le problème réside dans le fait que l'estimation d'erreur, le scan-matching dans ce cas, est une opération non-linéaire. La recherche de l'estimation de pose d'erreur minimale est alors une tâche difficile. De plus, Lu et Milios utilisent une approximation de « hill-climbing » dans le processus d'optimisation, approximation qui est très sensible à l'estimation initiale de la pose. Le processus converge donc souvent vers des minima locaux incorrects. Les auteurs [Gutmann et Konolige, 1999] indiquent que ce problème est critique dans les grands environnements cycliques puisque les erreurs d'estimation « en fermant la boucle » sont souvent suffisamment significatives pour que les scans proches ne se recouvrent pas. La méthode d'optimisation globale nécessite alors une identification correcte des scans qui doivent se recouvrir. Gutmann et Konolige introduisent ainsi le concept de fermeture de boucle dans leur article. Le terme est repris pour l'explication de leur méthode **LRGC** (Local Registration and Global Correlation) dont les résultats sont illustrés par la figure 1.1. La méthode repose sur celle de Lu et Milios mais en y ajoutant deux techniques que sont l'enregistrement local (Local Registration) et la corrélation globale (Global Correlation). La première consiste à ajouter efficacement de l'information à la carte courante. La deuxième est une détermination topologique correcte des relations entre les différentes poses, et ce notamment après de longs cycles. Les résultats sont illustrés par la figure 1.1.

Définition actuelle de la fermeture de boucle

Dans les travaux plus récents tel [Bosse *et al.*, 2004], le terme « fermeture de boucle » est plus couramment utilisé pour désigner le problème de consistance de construction de cartes dans le cadre d'environnements cycliques. Cette définition est celle communément utilisée dans les approches actuelles. De même, les travaux de détection de fermeture de boucle réalisés dans cette thèse reposent aussi sur cette définition.

Intérêts

Outre la construction de cartes consistantes abordée dans le premier paragraphe, le détection de fermeture de boucle présente d'autres avantages. Dans le contexte de la navigation, lors de l'estima-

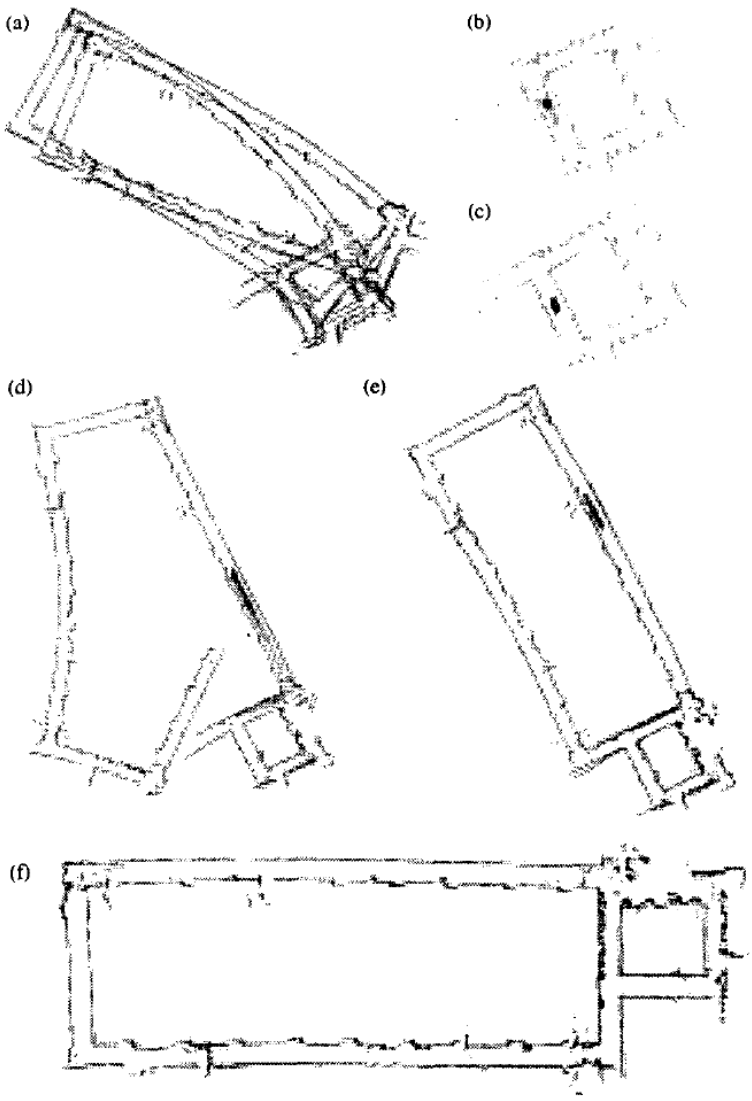


FIG. 1.1 – Construction de carte utilisant la méthode **LRGC** (Local Registration and Global Correlation). *(a)* Carte obtenue dans un environnement de 80mx25m à partir des données de l'odométrie. *(b)* Avant de fermer un petit cycle. *(c)* Après avoir fermé un petit cycle. *(d)* Avant de fermer un grand cycle. *(e)* Après avoir fermé un grand cycle. *(f)* Carte finale. Source [Gutmann et Konolige, 1999]

tion de trajectoire métrique suivie par le robot, il s'agit d'estimer l'ensemble des poses occupées par le robot. Le problème est alors le même que celui abordé dans le premier paragraphe mais sans construction explicite de la carte. L'hypothèse markovienne étant souvent prise en compte, l'estimation de la trajectoire est soumise au cumul de l'erreur lors du déplacement du robot menant à une dérive de la pose estimée. Des méthodes récentes ([Comport *et al.*, 2007], [Meilland *et al.*, 2011]) permettent d'obtenir une trajectoire contenant peu de dérive en utilisant des méthodes robustes d'estimation de

déplacements 3D entre deux poses successives du robot. Néanmoins, ces méthodes, bien qu'efficaces, sont sujettes au cumul d'erreur. La fermeture de boucle permet d'obtenir une estimation du cumul d'erreur lorsque le robot revient dans un endroit déjà visité. La trajectoire complète peut être re-estimée dans sa globalité afin de supprimer l'erreur. Le processus est une optimisation globale de minimisation de l'erreur comme dans les approches faites par [Lu et Milios, 1997] et [Gutmann et Konolige, 1999].

Quelque soit le modèle de représentation de l'environnement, la détection de fermeture de boucle est connexe au problème de la localisation du robot. En effet, la localisation globale, qui consiste à retrouver la position du robot dans une carte construite au préalable, est un problème qui requiert la reconnaissance de lieux passés sur la base des mesures actuelles du robot. De manière similaire, le problème du kidnapping de robot se rapporte à un problème de localisation globale. Le kidnapping de robot [Engelson et McDermott, 1992], [Choset *et al.*, 2005] consiste soit en un déplacement forcé du robot soit en une perte de la localisation du robot due à une occultation des capteurs ou une perte de l'information des capteurs. Quelque soit le cas, la localisation du robot dans la carte est perdue. Il s'agit alors pour le robot de retrouver sa localisation globale dans la carte connue de l'environnement.

1.2 Le problème de la détection visuelle de fermeture de boucle

Le problème de la fermeture de boucle a été introduit de manière assez générale dans la section précédente, indépendamment du type de représentation de l'environnement et du type de capteurs utilisés. Cette thèse s'inscrivant dans le contexte de la navigation basée vision, seules les méthodes de détection visuelle de fermeture de boucle sont présentées dans la suite. Les méthodes de localisation globale sont aussi abordées car les techniques sont souvent similaires. L'état de l'art s'organise en plusieurs sous-parties, chacune traitant une caractéristique importante permettant d'obtenir un algorithme de détection de fermeture de boucle fiable et robuste. Pour chaque caractéristique, une présentation du problème soulevé est faite, suivie d'une analyse des diverses solutions apportées. Les méthodes analysées sont regroupées suivant la caractéristique principale mise en avant par les auteurs, mais elles couvrent souvent plusieurs caractéristiques importantes. Lorsque c'est le cas, elles sont évoquées dans la présentation de chacune des méthodes proposées.

1.2.1 Algorithme hors-ligne/en-ligne

L'aspect hors-ligne/en-ligne de la détection de fermeture de boucle est une des caractéristiques les plus importantes et une des premières à avoir été abordées dans la littérature. L'objectif et l'utilisation

de l'algorithme déterminent s'il est nécessaire qu'il soit effectué en-ligne ou non. Si l'objectif est uniquement de construire une carte consistante de l'environnement à partir d'images de l'environnement, la détection de fermeture de boucle peut se faire hors-ligne. A contrario, une détection de fermeture de boucle en-ligne est indispensable dans deux cas de figure :

- La localisation dans une carte pré-existante. Afin de pouvoir naviguer convenablement dans l'environnement, il est nécessaire pour le robot de connaître sa position à chaque instant.
- Le cas plus général du SLAM. Le robot, en plus de devoir se localiser doit aussi créer une carte consistante de l'environnement. La position et la carte doivent alors être disponibles pour le robot à chaque instant afin de pouvoir planifier ses déplacements.

Les auteurs de [Se *et al.*, 2002] proposent une méthode de localisation globale par appariement entre les primitives locales et la carte. La position du robot est obtenue par reconstruction 3D à partir des positions des amers, primitives visuelles localisées précisément par une position absolue dans la carte, qui ressemblent le plus aux primitives de l'image courante. Afin d'obtenir une meilleure efficacité en terme de temps de calcul, une procédure RANSAC [Fischler et Bolles, 1981] est employée pour trouver rapidement un sous-ensemble d'amers de la carte qui permette une reconstruction cohérente de la position. Cette approche, basée sur le critère du maximum de vraisemblance, repose essentiellement sur la robustesse de l'association de données entre les primitives courantes et les amers de la carte. Dans [Williams *et al.*, 2007b], une approche similaire est mise en œuvre dans un algorithme de SLAM afin de retrouver la position de la caméra lorsque son suivi est interrompu, dû à une occultation du capteur ou un déplacement brutal. Des triplets d'amers de la carte ressemblant aux primitives extraites de l'image courante sont recherchés. La position métrique de la caméra est alors inférée à partir des coordonnées 3D du triplet d'amers. La méthode d'inférence, *i.e.* l'algorithme des « trois points » [Fischler et Bolles, 1981] est implémentée dans une procédure RANSAC visant à générer plusieurs hypothèses de localisation. L'hypothèse permettant d'assurer un maximum de projections cohérentes d'amers dans l'image est choisie. Pour assurer le fonctionnement temps-réel, la méthode précitée a été améliorée dans [Williams *et al.*, 2007a] à l'aide des Randomized Tree [Lepetit et Fua, 2006] pour la reconnaissance de triplets d'amers présents dans l'image courante. Les Randomized Trees reposent sur une forêt d'arbres binaires pour la reconnaissance de primitives visuelles. Il s'agit d'une méthode d'apprentissage nécessitant d'entraîner les arbres de décision sur la base d'exemples obtenus au préalable.

Dans ce cas, l'apprentissage est effectué en ligne : à chaque ajout d'une primitive visuelle, un ensemble de 400 primitives différentes est généré par déformation artificielle. La méthode originellement développée pour la relocalisation suite à un kidnapping de robot a été adaptée pour la détection de fermeture de boucle dans [Williams *et al.*, 2008]. Plutôt que de chercher à localiser la caméra dans une partie connue de l'environnement uniquement lorsque le suivi de position est interrompu, une telle procédure est mise en œuvre périodiquement. À chaque nouvelle acquisition, une recherche d'appariement entre les primitives de l'image courante et les zones éloignées de la carte est effectuée en utilisant le graphe de covisibilité des amers. Ce graphe, construit de manière incrémentale, lie entre eux les amers qui ont été observés simultanément dans la même image. La détection de fermeture de boucle repose alors sur l'appariement entre les primitives de l'image courante et les amers distants de ceux actuellement observés. En cas de succès, une fermeture de boucle est détectée. La carte obtenue grâce à cet algorithme est affichée sur la figure 1.2. L'algorithme présente l'avantage d'avoir une implémentation incrémentale permettant un traitement temps-réel à 30Hz. Toutefois, la méthode est sensible à l'aliasing perceptuel (le terme est expliqué dans la section 1.2.3 traitant de cet aspect).

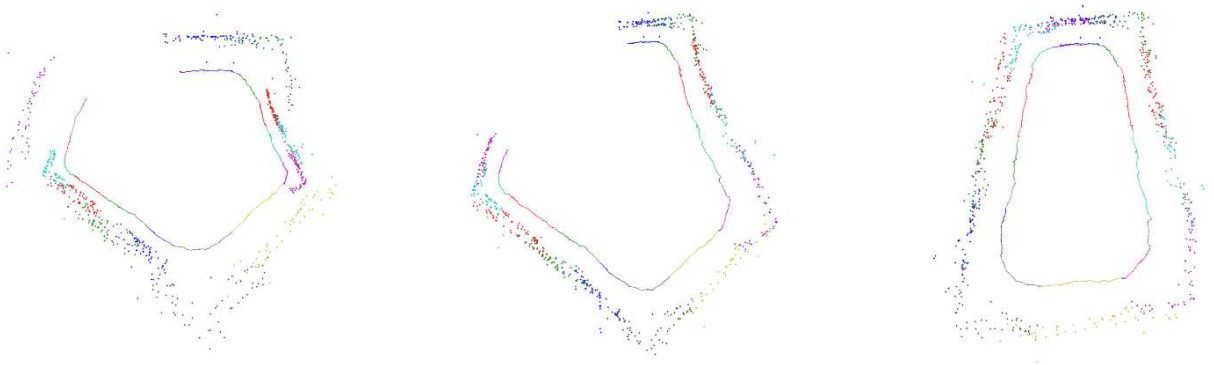


FIG. 1.2 – La carte est obtenue avec une caméra monoculaire (à gauche) puis mise à l'échelle automatiquement (au milieu). La carte est réajustée lorsqu'une fermeture de boucle est détectée (à droite). Source : [Williams *et al.*, 2008].

Les auteurs de [Kumar *et al.*, 2008] utilisent également des arbres de décision pour effectuer une détection de fermeture de boucle purement basée sur l'apparence. Dans ce cas il s'agit d'Extremely Randomized Trees qui est un ensemble de Randomized Trees. Les auteurs utilisent des primitives locales pour représenter les images obtenues à partir d'une caméra omnidirectionnelle suivant le principe du sac de mots visuels. Ces primitives sont enregistrées dans les Randomized Trees. La mesure de si-

milarité entre les images repose alors sur l'utilisation de leurs représentations sous forme de primitives locales. La construction des arbres contenant les primitives locales est effectuée lors d'une première phase hors-ligne d'apprentissage du modèle de l'environnement. L'utilisation des arbres de décision permet ensuite un appariement rapide entre les primitives de l'image courante et celles du modèle de l'environnement. La méthode présente de bons résultats et notamment une invariance à l'orientation (*cf.* figure1.3). Toutefois, de nombreux faux positifs apparaissent lors de la détection de fermeture de boucle. Ces derniers sont supprimés par vérification de géométrie épipolaire entre les deux images. Comme pour la méthode précédente, il semble que les méthodes utilisant des arbres de décision soient sensibles au phénomène d'aliasing perceptuel.

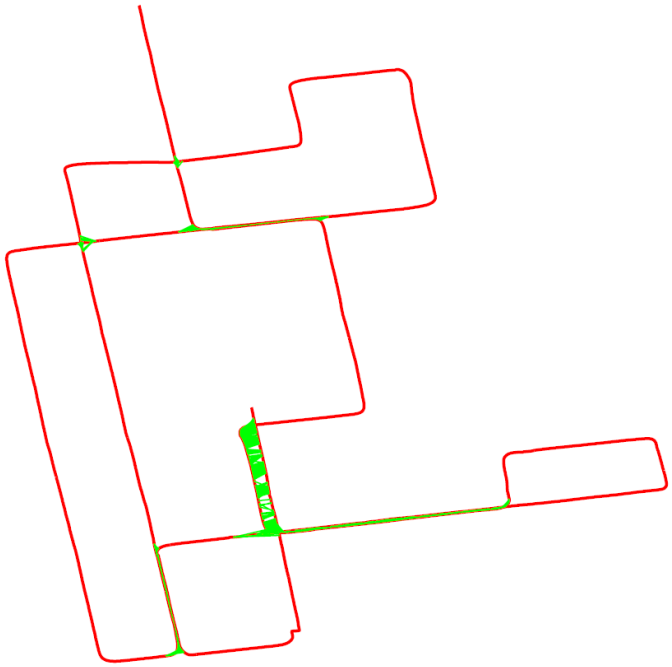


FIG. 1.3 – La trajectoire suivie par le robot est obtenue à l'aide du GPS et est représentée en rouge. Les images de fermetures de boucles sont repérées par des lignes vertes les liant. Source : [Kumar *et al.*, 2008].

La méthode des sacs de mots mise œuvre par [Nistér et Stewénius, 2006] est utilisée dans le contexte du SLAM visuel par les auteurs [Konolige *et al.*, 2009]. Les primitives locales sont extraites d'images stéréo et enregistrées dans un arbre de mots visuels modélisant l'environnement. La structure en arbre permet, comme précédemment, une recherche rapide des mots visuels déjà rencontrés. La mise en

correspondance rapide des mots visuels de chaque image permet de créer une carte de l'environnement et de détecter les fermetures de boucle en temps-réel. De plus, la méthode lie les images acquises entre elles par des contraintes créant ainsi un squelette représentant la trajectoire du robot. Ce squelette permet de contraindre la création de la carte, constituée d'images, afin de maximiser sa cohérence. Lors de la détection de fermeture de boucle, plusieurs vues sont mises en correspondance au sein du squelette. La fermeture de boucle n'est alors acceptée que si les squelettes de deux trajectoires coïncident. La méthode présente d'excellentes caractéristiques : construction de carte en temps-réel, mise à jour de la carte lors du déplacement du robot (carte dynamique), méthode incrémentale de construction et de détection de fermeture de boucle, très bonne consistance des cartes de vues créées, utilisable sur de très grands environnements. Les limitations principales proviennent du mécanisme d'optimisation du squelette. Au delà de 1000 arêtes au sein du squelette, les temps de calcul deviennent trop élevés pour réaliser un algorithme temps-réel. D'autre part, les auteurs utilisent des caméras monoculaires ne permettant pas de créer des contraintes à 6 degrés de liberté dans le squelette. Il s'agit du problème de l'indépendance à l'orientation du robot (le problème est décrit dans la section 1.2.4).

Certaines approches s'attachent à définir un cadre de représentation des images optimisé en fonction des caractéristiques de l'environnement. Le principe de l'Analyse en Composantes Principales (ACP) mis en œuvre dans les travaux de [Gaspar *et al.*, 2000], [Kröse *et al.*, 2000] et [Sim et Dudek, 1999] est d'apprendre, à partir d'une base de données d'images représentatives de l'environnement, une base optimale pour la projection ultérieure des images acquises par le robot. La base peut être apprise à partir de toute l'information contenue dans des images omnidirectionnelles [Gaspar *et al.*, 2000], à partir d'une version réduite de cette information dans des images omnidirectionnelles [Kröse *et al.*, 2000] ou bien même à partir de primitives extraites dans des images provenant d'une caméra monoculaire [Sim et Dudek, 1999]. La base obtenue ne retient que les dimensions de l'espace d'entrée, *i.e.* l'espace de représentation des images, qui apportent une information pertinente, les autres dimensions étant négligées. Le nombre final de dimensions est alors limité rendant les traitements ultérieurs rapides. La base optimisée est en fait constituée des vecteurs propres les plus pertinents obtenus à partir de la matrice de covariance formée par toute l'information de l'espace d'entrée disponible dans la base d'apprentissage. Dans [Gaspar *et al.*, 2000], par exemple, l'apprentissage étant directement réalisé sur les images omnidirectionnelles, les vecteurs propres obtenus au final sont des images propres (*cf.* figure 1.4). Une fois la base construite, l'image courante y est projetée afin de calculer la distance qui

la sépare des images d'apprentissage. Les coordonnées de l'image courante dans la base optimisée sont alors simplement comparées avec celles des images d'apprentissage. Les auteurs [Gaspar *et al.*, 2000] considèrent que l'image courante provient du même lieu que l'image d'entraînement la plus proche dans l'espace de représentation optimisé. Dans le cas de [Kröse *et al.*, 2000] et de [Sim et Dudek, 1999], un modèle génératif de l'apparence de l'environnement est également appris lors de la phase d'entraînement, à partir de positions données par une vérité terrain. Il est alors possible d'inférer une position métrique précise pour l'image courante sans reconstruction 3D explicite. Si ces méthodes présentent un avantage indéniable en ce qui concerne les aspects de calcul temps-réel et les réductions de bases de données, elles sont par contre sensibles à l'aliasing perceptuel. Réduire une image, ou ses primitives, à son information la plus pertinente engendre la perte de l'information complémentaire moins pertinente mais qui permet de distinguer deux endroits presque semblables (partageant de fait la même information pertinente).

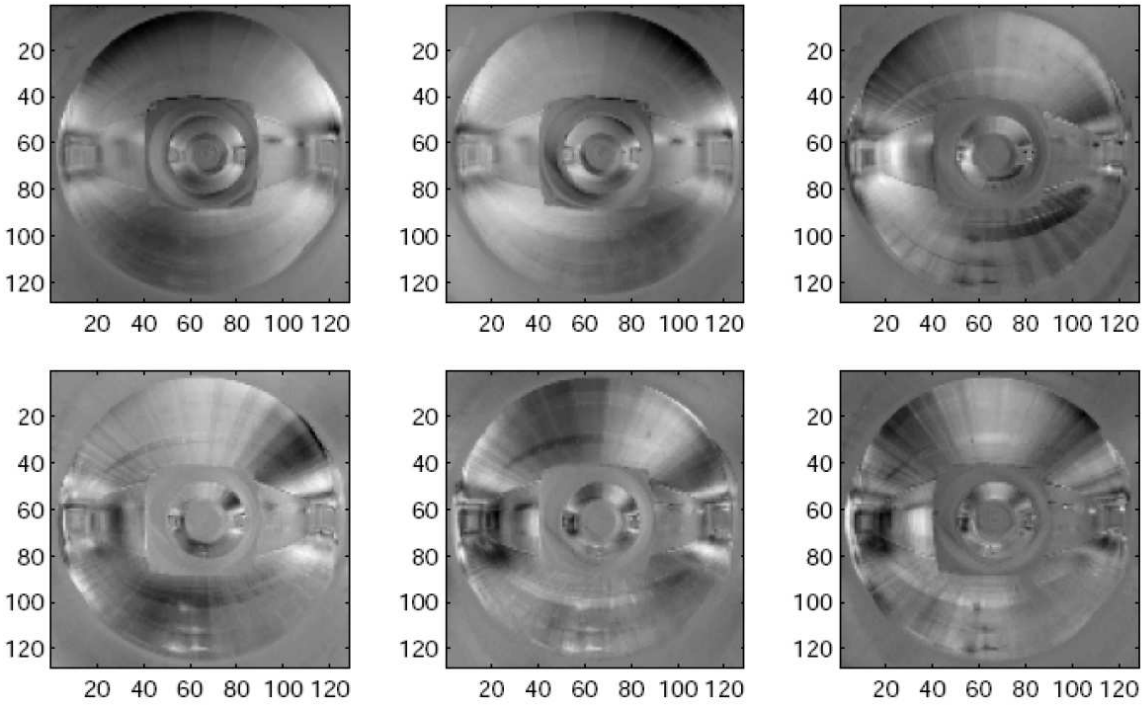


FIG. 1.4 – Exemples d'images propres constituant la base optimisée utilisée dans les méthodes d'ACP pour la comparaison des images. Ces 6 images correspondent aux 6 vecteurs propres les plus pertinents obtenus à partir de la matrice de covariance formée par toutes les images omnidirectionnelles de la base d'apprentissage. Source : [Gaspar *et al.*, 2000].

Les auteurs [Valgren *et al.*, 2007] adoptent une méthode nommée « *incremental spectral clustering* » groupant les images provenant d'un même lieu dans le cadre du SLAM topologique. Toute image acquise à partir d'une caméra omnidirectionnelle est alors comparée au représentant de chaque groupe du modèle de l'environnement obtenu jusque-là. La comparaison est effectuée sur la base d'une mesure de similarité entre les primitives locales, *i.e.* un simple comptage des primitives communes. Une matrice d'affinité renseignant sur les similarités entre l'image courante et l'ensemble des groupes est mise à jour grâce à cette mesure. L'algorithme *incremental spectral clustering* permet ensuite de déterminer le nombre optimal de groupes compte tenu des entrées de cette matrice, sur la base de l'apparence uniquement. L'opération consiste alors en la mise à jour d'un groupe existant avec la nouvelle image ou bien en la création d'un nouvel ensemble de groupes. Dans tous les cas, l'algorithme cherche à maximiser la proximité des images en fonction de l'apparence uniquement. L'image courante est simplement comparée au représentant de chaque groupe pour optimiser les performances. Par contre, tous les individus sont mémorisés afin de permettre une réorganisation des groupes si l'algorithme le requiert. La méthode obtenue est incrémentale avec des traitements réalisés en-ligne mais avec une complexité élevée (l'algorithme *incremental spectral clustering* nécessite plusieurs décompositions en valeurs singulières de la matrice d'affinité).

1.2.2 Robustesse en présence d'objets dynamiques

La qualité de la robustesse en présence d'objets dynamiques est liée à la méthode de traitement de l'information contenue dans l'image. Les objets dynamiques sont, contrairement aux objets statiques, des objets soit qui sont en mouvement lors du passage du robot soit qui sont apparus, ou ont disparus, lors de la revisite du robot dans un endroit. Une voiture qui se déplace dans un environnement urbain est un exemple d'objet dynamique en mouvement lors du passage du robot. Les tables et les chaises sont des exemples d'objets qui peuvent avoir disparu lors d'une revisite du robot. L'inconvénient de la présence de ces objets est qu'ils fournissent une information inutilisable pour la détection de fermeture de boucle et, de par leur présence dans l'image, ils occultent une partie de l'information éventuellement exploitable. Les différentes approches pour comparer des images seront donc plus ou moins sensibles aux objets dynamiques.

Les approches [Ulrich et Nourbakhsh, 2000] et [Wang *et al.*, 2006] définissent une mesure de similarité entre images en employant de simples méthodes de vote. Chaque nouvelle image acquise est

mise en correspondance avec toutes les images du modèle, et son lieu de provenance est déterminé comme étant le lieu de l'image du modèle avec lequel elle partage le plus de similarités, *i.e.* celle qui reçoit le plus de votes. Le nombre de votes dépend ainsi de la quantité de correspondances existant entre les primitives des images comparées. Avec les méthodes de vote et comptage, la caractérisation des images et le type de caméra sont libres tant qu'il existe une mesure de similarité qui puisse être calculée. Les auteurs [Ulrich et Nourbakhsh, 2000] caractérisent les images d'une caméra monoculaire avec des primitives globales alors que les auteurs [Wang *et al.*, 2006] utilisent le concept du sac de mots visuels, également à partir d'une caméra monoculaire. La différence majeure entre ces deux approches repose sur la capacité à gérer les objets dynamiques. L'approche [Ulrich et Nourbakhsh, 2000] utilisant des primitives globales est sensible aux objets dynamiques dans la mesure où ces objets sont enregistrés dans la caractérisation de l'image rendant impossible son exclusion lors du comptage des votes. Le cadre des sacs de mots visuels permet, quant à lui, une meilleure robustesse vis-à-vis des objets dynamiques. L'image étant caractérisée par un ensemble de primitives visuelles locales, il est possible de discerner les primitives du lieu de celles de l'objet dynamique. La mesure de similarité s'appliquant entre les primitives, lors du comptage des votes, les primitives appartenant aux lieux seront prises en compte tandis que celles appartenant aux objets dynamiques pourront être ignorées.

Dans le cadre de la localisation globale d'un robot, les auteurs [Pronobis *et al.*, 2006] emploient une technique de Séparateur à Vastes Marges (Support Vector Machine, SVM) pour apprendre un classifieur qui permet de prédire le lieu de l'image courante. La SVM est apprise lors d'une phase hors-ligne à partir d'une collection d'images étiquetées manuellement et décrivant un ensemble restreint de lieux. Dans le cadre de ces travaux, il s'agit de quelques pièces dans un environnement d'intérieur. L'apprentissage consiste à projeter les images d'entraînement dans un espace où il est possible de séparer chacune des classes entre elles par des séparations linéaires. Pour construire cet espace, les images sont encodées sous la forme de primitives globales. Une fois le classifieur appris, la classe de l'image courante est déterminée en projetant sa représentation dans l'espace de discrimination. La distance qui sépare cette image de l'ensemble des surfaces de séparation est calculée pour déterminer à quelle catégorie elle appartient. La méthode a été améliorée dans [Pronobis et Caputo, 2007] pour pouvoir prédire la classe d'une image de manière graduelle ; l'image est traitée petit à petit en essayant de déterminer son lieu de provenance à partir d'une information partielle. Dans certains cas, il est possible d'obtenir la classe de l'image sans nécessiter l'analyse complète de celle-ci. Les traitements sont alors effectués plus

rapidement. La méthode a également été adaptée par [Luo *et al.*, 2007] pour pouvoir mettre à jour le classifieur SVM avec de nouveaux exemples en-ligne. La robustesse aux changements de conditions dans l'environnement (illumination, déplacement d'objets) est alors accrue mais il n'est toutefois pas possible d'ajouter de nouveaux lieux dans le modèle. Malgré de bonnes performances obtenues dans les résultats expérimentaux présentés dans chacune des méthodes, l'inconvénient majeur des approches SVM réside dans la nécessité d'avoir une phase hors-ligne d'apprentissage du modèle. Il est intéressant de noter la robustesse de ces algorithmes aux changements de condition de l'environnement malgré l'utilisation d'une représentation de l'image par une primitive globale. Cette robustesse peut en partie s'expliquer par un apprentissage des SVM à partir de l'environnement sous diverses conditions.

1.2.3 Robustesse à l'aliasing perceptuel

L'aliasing perceptuel apparaît lorsque plusieurs endroits sont très similaires engendrant alors une ambiguïté dans la localisation du robot. Un exemple d'aliasing perceptuel est la similarité entre les différents couloirs d'un même bâtiment. Si l'algorithme n'est pas robuste à l'aliasing perceptuel, il sera difficile de déterminer précisément dans quel couloir se trouve le robot. La position réelle du robot dans l'environnement est perdue rendant alors les tâches de navigation et de cartographie impossibles à réaliser correctement. De manière générale, il s'agit de remplacer une estimation de fermeture de boucle basée sur le maximum de vraisemblance par une estimation basée sur le maximum a posteriori. L'avantage de cette deuxième méthode est qu'elle permet une intégration de l'information dans le temps, contrairement au maximum de vraisemblance qui évalue l'hypothèse la plus probable seulement à partir de l'observation courante. De ce fait, la robustesse aux erreurs passagères est accrue. Dans le cas de l'aliasing perceptuel, plusieurs lieux distincts de l'environnement se ressemblent. Ceci résulte en autant d'hypothèses de fermeture de boucle. En utilisant le maximum a priori, il est probable que les observations au cours du temps permettent de lever l'ambiguïté. Cette méthode permet d'accroître la robustesse à l'aliasing perceptuel mais ne garantit pas un taux de faux positifs nul.

La méthode présentée dans [Williams *et al.*, 2007a] et dans [Williams *et al.*, 2008] bénéficie de nombreux avantages du fait de son implémentation incrémentale et de son traitement temps-réel mais possède une importante faiblesse vis-à-vis de l'aliasing perceptuel. En effet, les Randomized Trees ne semblent pas robustes à l'aliasing perceptuel et comme décrit dans [Williams *et al.*, 2008], ils produisent un nombre élevé de faux positifs ensuite écartés par l'algorithme des « trois points ». Il en

est de même pour les travaux de [Kumar *et al.*, 2008], les faux positifs sont cette fois éliminés par contrainte de géométrie épipolaire. Présentant déjà une faiblesse à l'aliasing perceptuel, ces algorithmes sont difficilement utilisables dans les environnements présentant un aliasing perceptuel fort.

L'association de données dans un cadre probabiliste de filtrage particulaire, permettant d'intégrer l'information au cours du temps, garantit une meilleure robustesse face à l'aliasing perceptuel. Le modèle d'estimation repose sur le critère de maximum a posteriori. Ce type de modèle a été appliqué avec succès au problème de la localisation globale par la méthode de *Monte Carlo Localization* (MCL) [Dellaert *et al.*, 1999] (*cf.* figure 1.5) et du SLAM par celle du *Rao-Blackwellised particle filter* [Montemerlo *et al.*, 2003]. D'autres versions basées sur la méthode de Monte Carlo Localisation ont ensuite été développées [Andreasson *et al.*, 2005], [Wolf *et al.*, 2005]. Les approches [Barfoot, 2005], [Eade et Drummond, 2006], [Elinas *et al.*, 2006], [Karlsson *et al.*, 2005], [Pupilli et Calway, 2006], [Sim *et al.*, 2005], quant à elles, reposent sur le principe de la méthode du Rao-Blackellwised particle filter. Le principe de base du cadre probabiliste commun aux applications précitées consiste à approximer la distribution de probabilité de la position du robot dans son environnement grâce à un ensemble fini d'échantillons appelés particules (une particule correspond à une hypothèse de position), selon un processus récursif de tirages aléatoires avec remise. Lors de l'acquisition d'une nouvelle mesure en provenance des capteurs, chaque particule est pondérée en fonction de la pertinence de l'hypothèse qu'elle représente face à la mesure obtenue. Plus l'hypothèse est pertinente et plus le poids de la particule est renforcé. Ce calcul de vraisemblance correspond globalement à une mesure de similarité entre les primitives de l'image courante et les amers visibles compte tenu de la position représentée par la particule considérée. Une étape de ré-échantillonnage permet de concentrer les particules ayant les poids les plus élevés dans les lieux les plus vraisemblables. La distribution discrète de probabilité recherchée est obtenue par normalisation des poids calculés. Avant la prochaine mesure effectuée par les capteurs, l'ensemble des particules sera déplacé relativement à un modèle d'évolution temporelle de la position du robot. En utilisant le critère du maximum a posteriori, les méthodes *Monte Carlo Localization* et *Rao-Blackwellised particle filter* présentent une certaine robustesse face au phénomène d'aliasing perceptuel.

Les méthodes précédentes utilisées dans le cadre de l'association de données ont aussi été adaptées au cadre de la reconnaissance d'image pour la détection de fermeture de boucle. L'approche *Monte*

FIG. 1.5 – Illustration de la convergence du processus de localisation globale MCL : au départ, les particules sont dispersées sur toute la carte, avant de se concentrer au fur et à mesure des déplacements et des observations sur la position réelle. Source : [Dellaert *et al.*, 1999].

Carlo Localization est abordée dans les travaux de [Menegatti *et al.*, 2004] tandis que l'approche *Rao-Blackwellised particle filter* est utilisée dans les travaux de [Weiss *et al.*, 2007]. La distribution de probabilité de la position du robot est toujours discrétisée sous forme d'un ensemble de particules pondérées. Il ne s'agit plus de comparer les primitives de l'image courante avec les amers de la carte mais de faire un appariement de l'image courante avec l'ensemble des images faisant partie du modèle de l'environnement. Le modèle est construit lors d'une phase préalable hors-ligne au cours de laquelle chaque image considérée est associée à une position métrique précise obtenue grâce à une vérité terrain. Dans [Menegatti *et al.*, 2004], chaque image est caractérisée par une primitive globale localisée par une position métrique absolue tandis que dans [Weiss *et al.*, 2007], ce sont les primitives locales de chaque image qui sont localisées par leurs points de vue. À partir de ce modèle, lorsqu'une nouvelle mesure en provenance des capteurs est obtenue, le poids de chaque hypothèse de position est mis à jour en fonction des ressemblances trouvées entre l'image courante et les images du modèle. L'information métrique de position est alors obtenue à partir du point de vue associé à chacune des images du modèle.

Les auteurs [Newman *et al.*, 2006] proposent un cadre de gestion des hypothèses au cours du temps pour assurer une cohérence temporelle de l'estimation. Une matrice de similarité est construite sur la base de la comparaison de l'image courante avec l'ensemble des images traitées jusque-là. Chaque entrée $M(i, j)$ de cette matrice stocke une valeur proportionnelle à la mesure de similarité entre les images i et j. En cas de fermeture de boucle, les images correspondantes forment une suite consécutive d'entrées non-nulles sur un axe parallèle à la diagonale. La détection de fermeture de boucle consiste alors à extraire les éléments hors-diagonaux non nuls de la matrice. L'approche décrite propose une détection de fermeture de boucle en-ligne. Elle repose sur une méthode de sacs de mots visuels ap-

pliquée lors d'une phase hors-ligne d'acquisition du modèle de l'environnement. Les auteurs [Zivkovic *et al.*, 2005] utilisent également une matrice de similarité pour construire une carte topologique de l'environnement lors d'un processus hors-ligne.

D'autres approches utilisant estimation et filtrage ont été proposées afin de pallier les limitations du critère du maximum de vraisemblance. Les auteurs de [Goedemé *et al.*, 2007] proposent une construction de carte topologique reposant sur l'apparence uniquement. La détection de fermeture de boucle est modélisée à l'aide d'un formalisme mathématique dérivé de la théorie de l'évidence. Lors d'une première phase hors-ligne d'apprentissage non supervisé, une carte topologique de l'environnement est construite à partir d'une collection d'images acquises depuis une caméra omnidirectionnelle, qui permet de satisfaire les besoins d'indépendance à la rotation du robot. L'acquisition est effectuée à intervalle de temps régulier afin d'obtenir un recouvrement partiel entre les images. L'algorithme de construction de carte détermine ensuite les images provenant du même lieu. La ressemblance entre les images est déterminée à l'aide d'une mesure de similarité basée à la fois sur les primitives globales et locales. Les groupes d'images sont ainsi définis uniquement sur la base de l'apparence. À l'intérieur de chaque groupe, il est possible de définir plusieurs sous-groupes grâce à l'information de voisinage temporel : les images d'un même groupe acquises à des instants proches dans le temps peuvent être rassemblées. Il est ainsi possible de distinguer, parmi les images qui se ressemblent, celles qui ont été acquises au même moment. La fermeture de boucle correspond alors à une fusion de sous-groupes : deux sous-groupes d'images similaires prises à des instants éloignés correspondent à deux passages au même endroit à deux instants différents. Toutefois, ces sous-groupes peuvent également correspondre à deux lieux distincts qui se ressemblent. Une méthode de collection de l'évidence basée sur la théorie de Dempster-Shafer permet alors de lever l'ambiguïté. Le support de l'hypothèse résultante est mesuré dans le cas de la combinaison de deux sous-groupes. Si le résultat est supérieur à un certain seuil de probabilité, la fermeture de boucle est acceptée dans la carte construite. Dans le cas contraire, chacun des deux sous-groupes est considéré comme un groupe à part entière. Une fois la carte construite, elle est utilisée lors d'une seconde phase pour la localisation globale du robot (voir figure 1.6). Pour cette dernière tâche, une méthode classique de filtrage bayésien permet d'estimer la probabilité de la position du robot à partir de la même mesure de similarité que pour la construction de la carte.

Le formalisme probabiliste de filtrage bayésien décrit dans les travaux de [Cummins et Newman,

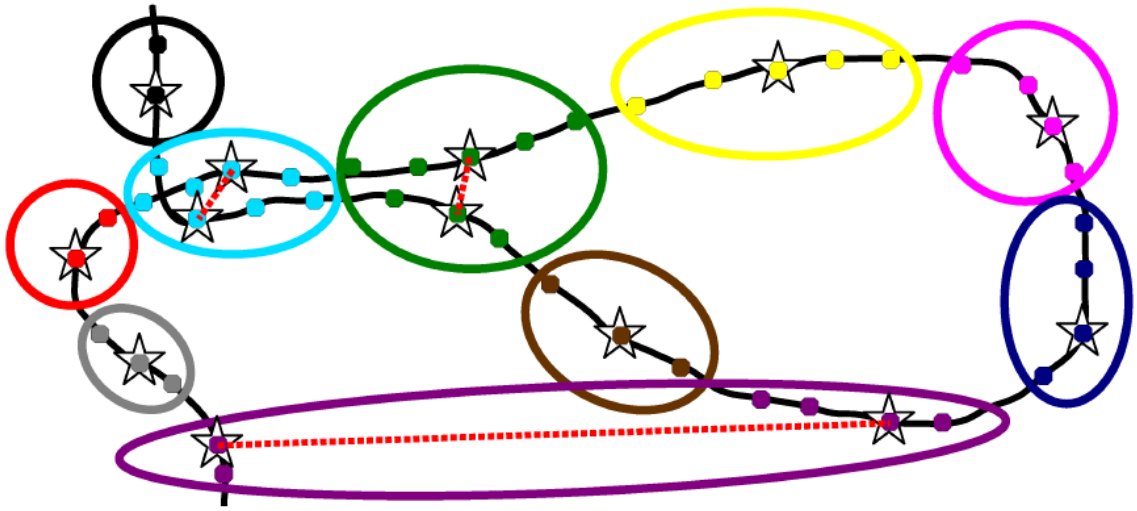


FIG. 1.6 – Illustration de la méthode de regroupement d'images pour la caractérisation des lieux. Chaque ellipse regroupe les images (*i.e.* les points dans la figure) sur la base de la similarité uniquement. A l'intérieur de chaque ellipse, les sous-groupes sont définis sur la base de la proximité temporelle d'acquisition des images. Un représentant pour chaque sous-groupe est désigné et caractérisé dans la figure par une étoile. La théorie de Dempster-Shafer permet alors de décider si un groupe doit être scindé ou fusionné. La fusion de sous-groupes correspond à une détection de fermeture de boucle. Source : [Goedemé *et al.*, 2007].

2007] propose une solution au problème du SLAM topologique sur la base de l'apparence uniquement et reposant sur l'utilisation d'une simple caméra monoculaire. La méthode permet de déterminer la probabilité que deux images viennent du même endroit dans le cadre d'une estimation au sens du maximum a posteriori. Les images et les lieux sont encodés par des collections de primitives locales d'après le principe des sacs de mots visuels. Pour évaluer la probabilité de fermeture de boucle, un modèle génératif de l'apparence de l'environnement est appris lors d'une première phase hors-ligne. Les probabilités de co-occurrence des primitives visuelles locales extraites des images d'entraînement sont estimées lors de cette première phase. Les images d'entraînement sont recueillies sur une vaste zone représentative du type d'environnement dans lequel le robot évoluera par la suite. Lors d'une seconde phase d'exploitation en-ligne, ces probabilités permettent de déterminer la probabilité de provenance de l'image courante. L'algorithme obtenu présente des caractéristiques intéressantes, notamment une robustesse impressionnante face à l'aliasing perceptuel. Toutefois, l'aspect non incrémental lié à l'apprentissage du modèle de l'environnement est regrettable. De même, les temps de calcul sont importants et empêchent des traitements en temps-réel. Dans [Cummins et Newman, 2008], les auteurs

présentent une version améliorée de l'algorithme sans toutefois atteindre des performances temps-réel. Les résultats sont illustrés par la figure 1.7.

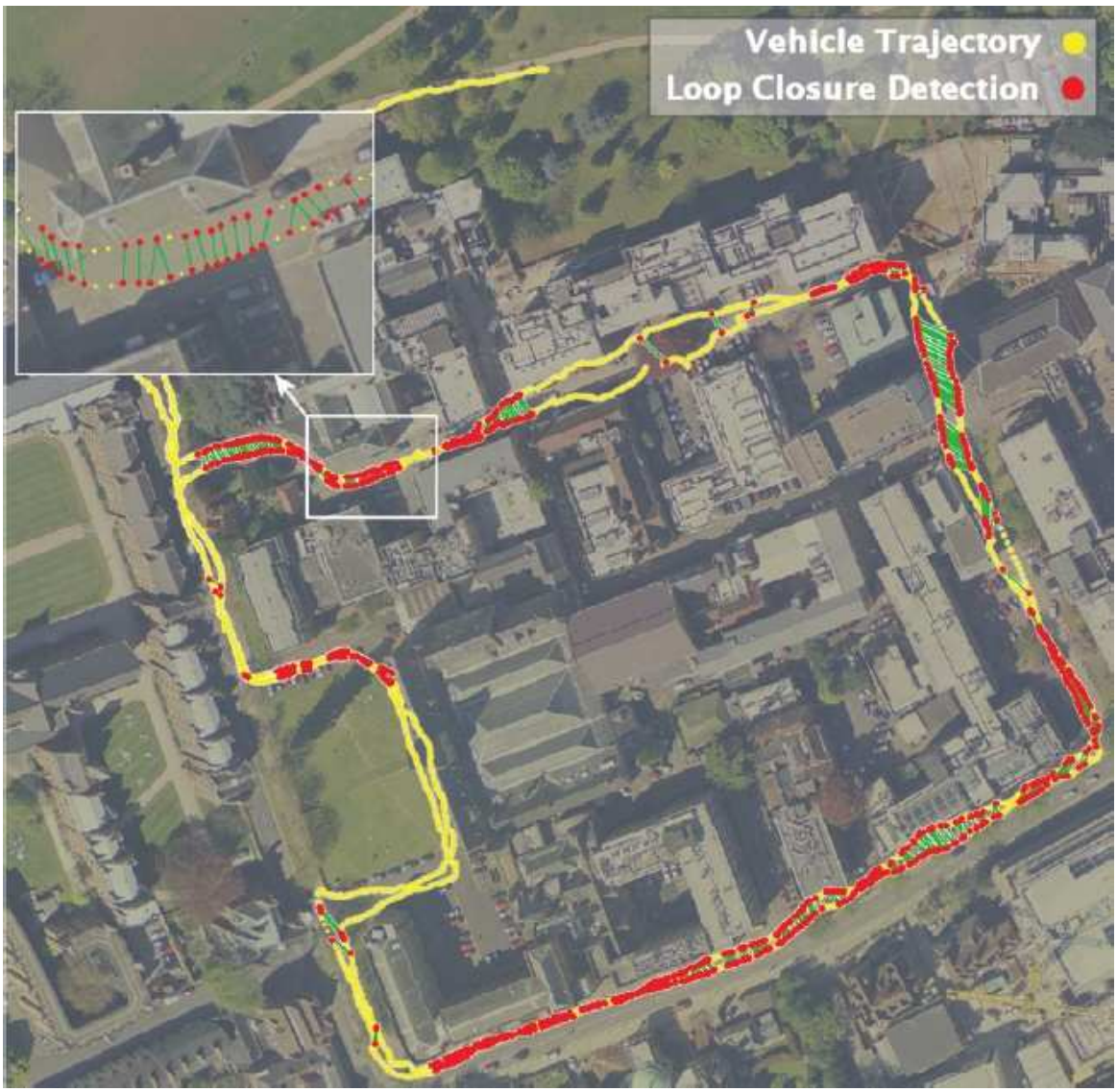


FIG. 1.7 – Résultats de la reconnaissance de lieu basée sur l'apparence superposés sur une photographie aérienne. Le robot traverse deux fois la boucle avec une trajectoire totale de 2km, collectionnant 2474 images. Les positions (corrigées manuellement à partir d'un GPS) auxquelles le robot a acquis des images sont marquées par des points jaunes. Deux images qui ont reçu une probabilité $p \geq 0.99$ de provenir du même endroit (à partir de l'apparence uniquement) sont marquées en rouge et sont jointes par une ligne. Aucun faux positif n'a atteint ce seuil de probabilité. Source : [Cummins et Newman, 2008].

Afin de limiter la sensibilité à l'aliasing perceptuel, les auteurs de [Košecká *et al.*, 2005] utilisent

un Modèle de Markov Caché (*i.e.* Hidden Markov Model, HMM) dans le contexte de la localisation globale. La méthode repose sur l'utilisation de primitives locales extraites des images suivant le paradigme du sac de mots visuels. Les images sont ensuite groupées en lieux, chaque lieu est alors représenté par un ensemble de vues du modèle associées à leurs mots visuels. Le groupement d'images se fait simplement par un système de comptage des primitives communes aux différentes images. Cela revient à grouper les images dont la covisibilité des primitives locales est très élevée. Le modèle de Markov Caché intervient pour modéliser les relations de voisinage entre les différents lieux et maintient ainsi une consistance globale de l'environnement. Le processus de localisation repose sur un système à deux étages. Le premier consiste en un simple système de classification de l'image courante suivant un mécanisme de vote. Le deuxième exploite le modèle de Markov de l'environnement permettant d'introduire les relations de voisinage dans la détection de fermeture de boucle. Ce deuxième étage permet ainsi de limiter les effets de l'aliasing perceptuel. La méthode permet d'obtenir de bons taux de reconnaissance entraînant une bonne localisation globale dans des environnements à grande échelle.

Dans [Filliat, 2007], l'auteur développe une méthode d'apprentissage supervisée pour aborder le problème de la localisation globale qualitative. Pour cela, il utilise une méthode incrémentale d'apprentissage basée sur l'apparence et en interaction avec un superviseur. Cette méthode caractérise les images par des primitives locales et les lieux par les collections de primitives locales des images correspondantes. Il s'agit d'une version incrémentale du formalisme du sac de mots visuels. Lorsque le robot acquiert une nouvelle image, il cherche à en déterminer le lieu grâce à un simple mécanisme de vote à deux étages. Le premier vote permet de déterminer, en analysant l'image dans différents espaces de représentation, les lieux de provenance les plus probables dans chacun de ces espaces séparément. Le second étage du vote permet de fusionner les notes obtenues à l'issue du premier étage. À la fin de la procédure, si le niveau de confiance dans le vote final n'est pas satisfaisant, une nouvelle image est acquise. Si après avoir traité un certain nombre d'images ce seuil n'est toujours pas atteint, le superviseur indique le lieu actuel. Si le seuil de confiance est dépassé, le superviseur vérifie le lieu prédit par le robot afin de corriger en cas d'erreur. Après chaque interaction avec le superviseur, le modèle de l'environnement est mis à jour : l'apprentissage est donc réalisé en-ligne en mémorisant simplement les lieux dans lesquels chaque primitive visuelle a été vue. L'approche proposée présente de nombreuses qualités et notamment sur le traitement complètement incrémental, *i.e.* depuis la construction du modèle de l'environnement jusqu'à l'apprentissage des lieux correspondants mais repose sur

l'interaction régulière avec un superviseur. Dans [Angeli *et al.*, 2008b], la méthode est modifiée dans le cadre de la détection de fermeture de boucle. L'image est cette fois représentée seulement par un type de primitives locales. La détermination de la fermeture de boucle repose sur un mécanisme de vote permettant d'attribuer un score de similarité entre l'image courante et les images du modèle. Un processus d'inférence bayésienne permet de mettre à jour les hypothèses a posteriori de fermeture de boucle à partir des résultats du vote. Le superviseur est remplacé par un mécanisme de vérification par géométrie épipolaire. La fermeture de boucle détectée est validée si la transformation entre les images est cohérente. Les auteurs introduisent aussi la notion d'image virtuelle qui contient toutes les primitives locales les plus communes créant ainsi une image présentant le plus de similarité avec l'ensemble des images du modèle. Si, lors du processus de détection de fermeture de boucle, l'image virtuelle est l'image la plus similaire avec l'image courante, alors l'hypothèse de fermeture de boucle est rejetée. La méthode présente les avantages décrits dans [Filliat, 2007] mais sans faire appel à un superviseur. De plus, les expériences montrent une excellente robustesse à l'aliasing perceptuel. Diverses approches ont alors été élaborées à partir de cet algorithme. [Angeli *et al.*, 2008c] adaptent la méthode à l'utilisation de plusieurs espaces de représentations de l'image comme dans les travaux originaux de [Filliat, 2007]. L'objectif est de permettre une robustesse encore accrue face à l'aliasing perceptuel. La méthode montre d'ailleurs de très bons résultats en environnement d'intérieur et d'extérieur. La méthode est adaptée au SLAM topologique dans [Angeli *et al.*, 2008a]. La carte topologique obtenue est affichée sur la figure 1.8 Dans [Angeli *et al.*, 2009], les auteurs ajoutent l'odométrie du robot permettant de connaître le déplacement relatif entre les images afin d'estimer une position 2D absolue pour la position des nœuds. La cohérence de la carte construite est ainsi améliorée. La solution de détection de fermeture de boucle est aussi adaptée au cadre plus restreint de la localisation globale. La méthode a été utilisée avec succès dans des environnements d'intérieur et d'extérieur présentant un fort aliasing perceptuel.

1.2.4 Indépendance à l'orientation du robot

Lorsque l'information de l'environnement est extraite de l'image, elle est souvent dépendante de l'orientation du robot. C'est-à-dire que l'information est liée non seulement au lieu où se trouve le robot, mais aussi au point de vue. Une fermeture de boucle, lors de la revisite du robot dans un même endroit, n'est alors détectable que sous des conditions de prises de vues similaires. Une légère variation

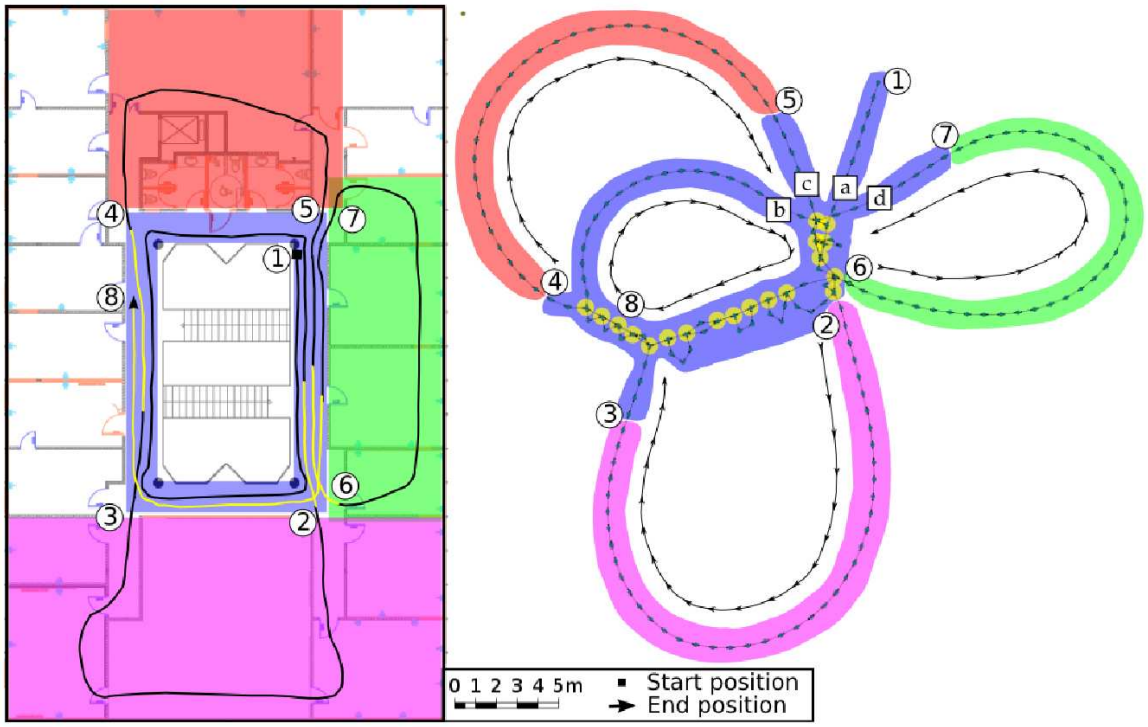


FIG. 1.8 – Plan de masse du bâtiment exploré avec la trajectoire de la caméra superposée (à gauche). La carte topologique correspondante (à droite) est obtenue en utilisant un modèle de relaxation à ressorts. Source : [Angeli *et al.*, 2008a].

de la prise de vue est toutefois acceptable en fonction de la robustesse aux transformations affines de l'information extraite. La dépendance à l'orientation du robot limite alors grandement les cas pratiques d'utilisation. Soit le robot est contraint à revisiter les lieux sous les mêmes conditions d'orientation (contrainte difficile à réaliser dans le cadre de la navigation et de la cartographie automatiques), soit la localisation et la cartographie sont sujettes à l'erreur. L'indépendance de la détection de fermeture de boucle à l'orientation du robot est une caractéristique indispensable pour réaliser des tâches de navigation et de cartographie de manière robuste quels que soient le type d'environnement et la trajectoire suivie par le robot. Deux solutions sont généralement utilisées pour pallier le problème de la dépendance de l'orientation du robot :

- Soit en utilisant une connaissance approximative de la position du robot.
- Soit en effectuant un appariement panoramique indépendant de l'orientation.

Dans le cadre applicatif du SLAM métrique, les auteurs [Eustice *et al.*, 2004] et [Lemaire *et al.*, 2007] proposent d'implémenter de simples méthodes de comptage de similarités afin de permettre la détection de fermeture de boucle. L'algorithme de filtrage employé pour le SLAM dans [Eustice *et al.*, 2004] repose explicitement sur la comparaison de l'image courante avec l'ensemble des images passées. Dans le delayed-state SLAM, la trajectoire de la caméra est estimée à chaque acquisition d'image comme autant de points de passage, en définissant des contraintes entre ces points de passage. Ces contraintes sont obtenues par un algorithme de géométrie multi-vues et renseignent sur les transformations relatives entre images. La détection de fermeture de boucle revient à chercher, parmi les images passées de la trajectoire, celle qui ressemble le plus à l'image courante et avec laquelle une transformation relative peut-être calculée. Pour limiter la recherche et la rendre plus efficace, seules les images passées dont les points de vue sont proches de la position actuellement estimée sont prises en compte, rendant la détection de fermeture de boucle dépendante du processus d'estimation. Dans un cadre plus classique, les auteurs de [Lemaire *et al.*, 2007] construisent une carte métrique d'amers de l'environnement. Une mémoire visuelle liant chaque amer de la carte aux images dans lesquelles il a été vu est maintenue en parallèle. Les images enregistrées dans cette mémoire visuelle sont par ailleurs associées à leur point de vue. L'image courante est alors régulièrement comparée aux images de la mémoire visuelle dont le point de vue est proche de la position actuellement estimée. En cas d'un nombre significatif d'appariements entre les primitives locales de ces images, il est possible de forcer l'observation des amers correspondants dans la carte pour provoquer la fermeture de boucle dans l'algorithme de SLAM. La méthode proposée est donc très simple mais la détection de fermeture de boucle dépend de la position estimée. Dans [Lemaire et Lacroix, 2007], ces travaux ont été adaptés au cadre de la vision panoramique permettant de rendre la méthode indépendante à l'orientation du robot.

Les auteurs [Sim et Dudek, 2004] utilisent dans un cadre d'inférence probabiliste un système d'apprentissage de modèle génératif de l'apparence de l'environnement. Il est alors possible d'inférer une position métrique pour le robot à partir d'une simple observation d'amers, sans recourir à une reconstruction explicite à partir des positions des amers. Pour y parvenir, une phase d'apprentissage hors-ligne est effectuée afin d'associer chaque amer de la carte aux positions de la caméra à partir desquelles il a été perçu. La fonction liant l'observation des amers à la position de la caméra est approximée en interpolant entre les différentes positions de la caméra. Cette fonction, qui constitue le modèle d'observation, permet de retrouver instantanément le point de vue de la caméra à partir des

amers perçus. Bien que présentant une bonne robustesse vis-à-vis du point de vue du robot, cette approche a pour principal défaut de reposer sur une phase d'apprentissage lourde. Une importante quantité d'images prises à des points de vues proches doit être analysée pour obtenir le modèle de l'environnement. De plus, la carte est construite au préalable et il n'est pas possible de la mettre à jour, limitant ainsi l'approche à des tâches de navigation dans des environnements connus.

Dans [Fraundorfer *et al.*, 2007], les auteurs réalisent du SLAM topologique à partir d'une base de données d'images représentatives de l'environnement. Les images sont ensuite caractérisées selon le formalisme des sacs de mots visuels. La méthode mise en œuvre pour réaliser l'apprentissage est similaire à celle employée dans les travaux de [Nistér et Stewénius, 2006]. Une fois cette phase achevée, une carte topologique de l'environnement est inférée en-ligne au cours du déplacement du robot sur la base de la comparaison des images acquises. Chaque nouvelle image est comparée aux images traitées jusqu'à l'instant présent grâce à une méthode de vote pour déterminer l'image qui ressemble le plus à l'image courante. Un algorithme de géométrie multi-vues est employé pour valider le résultat du vote si la transformation entre les images est cohérente. La méthode utilise une caméra standard mais présente malgré tout une certaine invariance à l'orientation. Le robot cherche toujours à retrouver une position et orientation connues le remettant dans une situation lui permettant d'observer de manière similaire les images de la base de données. Pour cela, les auteurs limitent les rotations du robot afin de garantir de toujours avoir une partie de l'environnement connu dans le champ de vision du robot. De ce fait, lors des expériences de homing, le robot retrouve efficacement son point de départ mais avec un déplacement en marche arrière. Il ne s'agit donc pas d'une véritable invariance à l'orientation du robot.

Dans un contexte de localisation globale par reconnaissance d'images, les auteurs [Booij *et al.*, 2007] utilisent une carte topologique construite au cours d'une première phase hors-ligne d'apprentissage non supervisé du modèle de l'environnement. Toutes les images d'entraînement sont comparées les unes aux autres de façon à les lier en fonction de leurs ressemblances. Chaque arête de la carte correspond alors à une mesure de similarité entre les images qu'elle relie. La mesure de similarité est obtenue d'abord par un comptage de primitives locales communes aux deux images. Un algorithme de géométrie multi-vues est ensuite appliqué à partir des correspondances obtenues afin de déterminer la transformation exprimant le changement de point de vue entre ces deux images. La caméra omnidirectionnelle est utilisée pour rendre les mécanismes de construction de la carte et de localisation indépendants de l'orientation

du robot. La distance choisie ici pour construire les arêtes de la carte topologique correspond à une mesure de la qualité de la transformation (*i.e.*, il s'agit du pourcentage de primitives locales qui peuvent être correctement projetées d'une image à l'autre compte tenu de la transformation calculée). La carte topologique ainsi construite renseigne sur la similarité entre nœuds : plus les images correspondantes sont semblables, plus le lien est fort. La carte de l'environnement obtenue est affichée sur la figure 1.9. Le mécanisme de localisation dans la carte utilise la même procédure que lors de la construction de la carte, le lieu de l'image courante est le nœud avec lequel elle partage la plus forte similarité. À chaque étape de localisation, l'image courante est donc comparée exhaustivement à tous les nœuds de la carte.

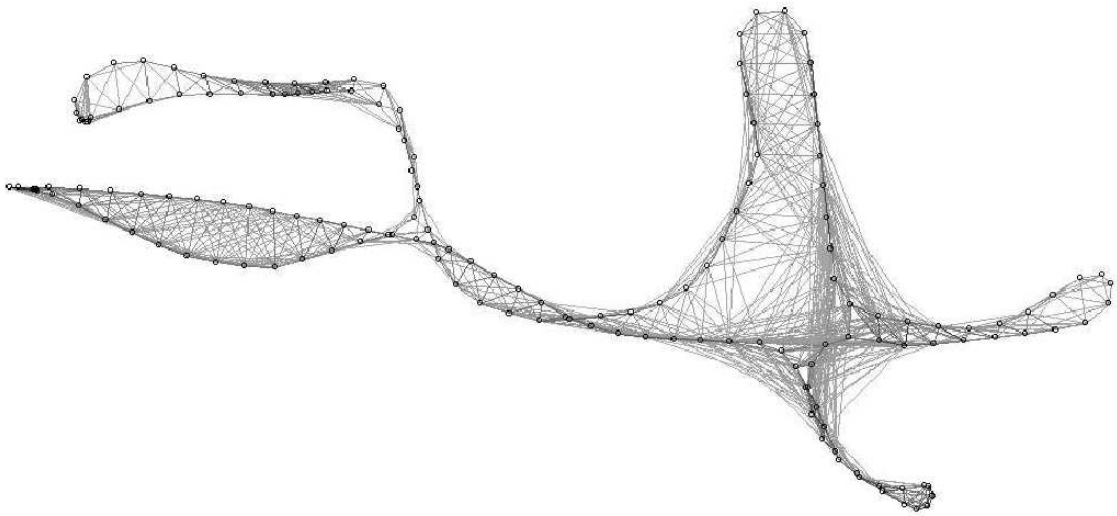


FIG. 1.9 – Le graphe d'apparence obtenu dans [Booij *et al.*, 2007]. Les cercles dénotent les positions approximées des images et les lignes les connectant dénotent les images appariées. La valeur de gris des lignes correspond à la valeur de similarité de l'appariement.

Les auteurs [Hübner et Mallot, 2007] mettent en œuvre une méthode simple de SLAM topologique en utilisant une caméra panoramique. Chaque image est caractérisée par une primitive globale très simple. Elle est constituée d'un vecteur de 72 pixels en niveau de gris pris sur la ligne d'horizon. Les images sont comparées en utilisant la norme L2 séparant les vecteurs associés. La carte quant à elle est construite par ajout d'un nœud décrit par l'image courante lorsqu'il se distingue assez des autres nœuds. L'image courante est simplement mise en correspondance avec le dernier nœud ajouté dans la carte ou bien le dernier nœud de fermeture de boucle afin d'éviter une comparaison exhaustive avec tous les nœuds de la carte. La détection de fermeture de boucle repose sur la comparaison de l'image

courante avec tous les nœuds dont la position est proche de l'estimation de la pose courante. Cette estimation est obtenue grâce à un algorithme de relaxation contraint par les relations d'adjacence entre nœuds. L'information de position relative des nœuds est obtenue à l'aide de l'odométrie du robot. Malgré des résultats positifs présentés dans le cadre d'une simple arène sans aliasing perceptuel, l'information visuelle prise en compte est très simpliste et pauvre. L'algorithme bénéficie d'une invariance à l'orientation du fait de l'utilisation d'une caméra panoramique. La primitive globale simpliste permet du calcul temps-réel mais présente une importante faiblesse vis-à-vis des objets dynamiques se trouvant sur la ligne d'horizon. Enfin, la pauvreté de la primitive rend l'algorithme très sensible à l'aliasing perceptuel.

Comme pour l'approche précédente, les auteurs [Murillo et Košecká, 2009] utilisent une primitive globale de relativement faible dimension, un vecteur de 320 dimensions. Cette primitive globale est calculée sur chacune des 4 images constituant la vue panoramique formant ainsi un vecteur contenant les 4 primitives. Afin de réduire la base de données pour optimiser le temps de recherche de l'image la plus proche de l'image courante dans l'espace de représentation, les auteurs adoptent un mécanisme de partitionnement de type k-means pour grouper les primitives globales des panoramas les plus semblables (*cf.* figure 1.10). Ce groupement est rendu possible par la nature même de la primitive globale qui varie peu lorsque les images sont prises dans un voisinage relativement large (une rue complète par exemple dans le cas d'un environnement urbain). Pour effectuer une localisation globale du robot, une simple mesure de similarité (*i.e.* mesure de distance euclidienne) entre l'ensemble de primitives globales de l'image panoramique courante et l'ensemble des images représentatives du modèle (*i.e.* les primitives caractéristiques de chaque partition) est effectuée. L'approche utilisée présente de bons résultats quant à localisation globale dans des environnements à grande échelle. L'utilisation d'une caméra panoramique associée à un mécanisme de permutation des primitives globales de l'image courante permet d'assurer une invariance à l'orientation du robot lors de la reconnaissance de lieu pour la localisation. Toutefois, cette représentation de l'environnement reste sensible à l'aliasing perceptuel obligeant les auteurs à ajouter des primitives locales pour vérifier la consistance de la transformation géométrique entre les lieux fermant la boucle et ainsi supprimer les faux positifs.

FIG. 1.10 – Vue moyenne obtenue à partir des clusters de primitives globales et servant de vue de référence pour les algorithmes de localisation globale dans [Murillo et Košecká, 2009]. Chaque vue est une moyenne d'environ 1000 panoramas (4000 images).

1.2.5 Robustesse de la détection dans les expériences à long-terme

Le problème du changement de l'environnement apparaît dans les expérimentations dites « long term experiments » (expériences de longue durée) ou encore « time varying experiments » (expériences dans des environnements variants avec le temps). Il s'agit de la capacité à gérer des environnements qui varient avec le temps et dont l'information extraite est différente. Par exemple, si un environnement est cartographié en été, quelle sera la capacité du robot à se localiser dans ce même environnement contenant de la neige en hiver ? Le problème de ce genre de situation est que l'information dite fiable et robuste des éléments statiques des scènes est modifiée au cours du temps. Le changement s'effectue sur une durée bien plus longue que la variation d'information due aux objets dynamiques. L'algorithme de détection de fermeture de boucle nécessite un système qui peut : soit utiliser de l'information robuste aux variations dues au temps, soit n'utiliser que l'information ne changeant pas malgré les modifications de l'environnement, soit adopter un mécanisme de mise à jour de l'information. Utiliser une information robuste à ces changements revient à conduire des expérimentations dans différentes conditions. Le travail d'apprentissage serait alors long et fastidieux.

Les auteurs de [Dayoub et Duckett, 2008] introduisent une méthode permettant de mettre à jour les vues de référence dans une carte topologique pour la localisation dans les environnements changeants.

Pour cela, ils utilisent les concepts de mémoire à court-terme et à long-terme basés sur le modèle de multi-enregistrement de la mémoire de l'être humain proposé par [Atkinson et Shiffrin, 1968]. Le modèle est constitué de trois niveaux : la mémoire sensorielle (MS), la mémoire à court-terme (MCT) et la mémoire à long-terme (MLT). La MS contient alors les primitives locales extraites de l'image courante. Un mécanisme d'attention sélectionne les primitives qui vont être déplacées dans la MCT. Cette dernière sert de mémoire intermédiaire où les nouvelles observations sont gardées pour une courte durée. Pendant ce temps, le système utilise un mécanisme d'entraînement pour sélectionner les primitives les plus stables à transférer à la MLT. Afin d'éviter une surcharge de la mémoire du système et pour s'adapter aux changements de l'environnement, le système contient aussi un mécanisme qui oublie les primitives inutilisées de la MLT en supprimant ces primitives du nœud de la carte. La MLT est alors utilisée par le mécanisme d'attention pour sélectionner la nouvelle information sensorielle afin de mettre à jour la carte. La méthode utilisée montre de bons résultats pour les expérimentations à long terme avec un système de mise à jour efficace. Dans [Dayoub *et al.*, 2009], les auteurs étendent la méthode à la localisation métrique du robot dans l'environnement. L'objectif est de montrer la possibilité d'utiliser le mécanisme de mémoire, entraînant la suppression et l'ajout de primitives dans les vues de référence tout en gardant la précision de la localisation dans des expérimentations à long-terme. Les expérimentations ont été effectuées sur une période d'approximativement 9 semaines dans un environnement changeant régulièrement. Il s'agit du restaurant étudiant de l'Université Lincoln dans lequel de nombreuses activités étudiantes sont organisées. Les résultats montrent une excellente adaptation aux changements de l'environnement permettant une navigation précise du robot.

Dans le cadre de la localisation topologique globale, les auteurs [Valgren et Lilienthal, 2007] effectuent des tests de robustesse de la reconnaissance d'images dans un environnement changeant au cours des saisons. Les images de test ont été acquises sur une période de 9 mois. La méthode repose sur une simple recherche de l'image la plus proche à partir des primitives locales SIFT ou SURF. Si les résultats avec les primitives SURF sont meilleurs, les auteurs démontrent qu'il n'est pas possible de faire de la reconnaissance d'image dans des expériences à long terme à partir d'une seule image et sur la simple comparaison des primitives les plus robustes. Pour améliorer la reconnaissance, le seuil de comparaison des primitives est abaissé afin d'obtenir plus de primitives communes entre deux images. Le taux de reconnaissance est ainsi amélioré mais le taux de faux positifs est aussi accru. Pour diminuer le nombre de faux positifs, une contrainte de vérification de géométrie épipolaire est ajoutée.

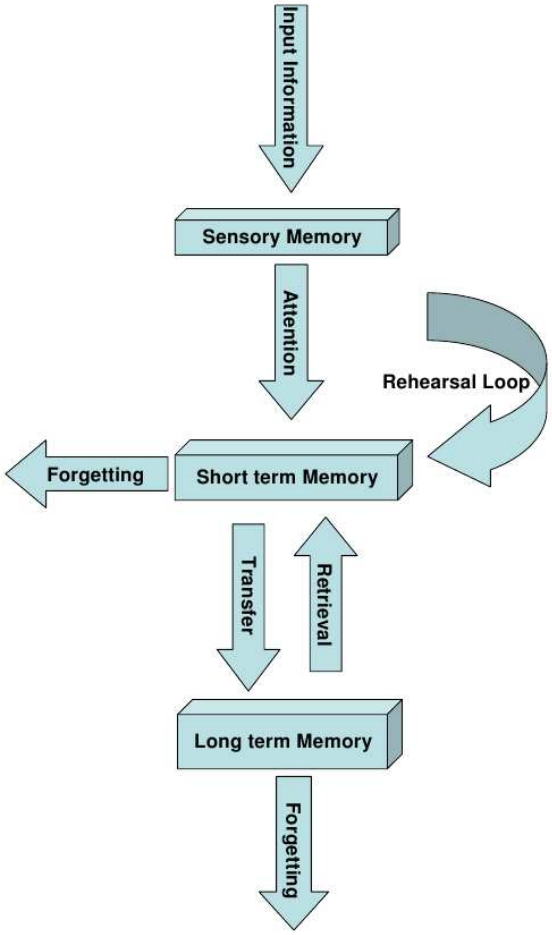


FIG. 1.11 – Mécanisme de mémoire utilisé dans [Dayoub et Duckett, 2008] pour la mise à jour de la carte dans des environnements présentant de forts changements dans les expérimentations à long-terme.

Les auteurs réalisent ainsi une méthode efficace permettant un bon taux de reconnaissance des lieux malgré le changement d'apparence important dû aux différentes saisons.

1.3 Conclusion

Au travers des méthodes élaborées pour résoudre le problème de la détection visuelle de fermeture de boucle, de nombreux aspects ont été abordés pour améliorer la robustesse et la fiabilité des algorithmes. Notamment, de nombreux travaux se sont focalisés sur la création de méthodes de détection de fermeture de boucle en temps-réel que ce soit en réduisant la taille de la base de données ([Gaspar *et al.*, 2000], [Kröse *et al.*, 2000], [Sim et Dudek, 1999]), en utilisant des structures de recherche efficaces comme les Randomized Trees ([Williams *et al.*, 2007a], [Williams *et al.*, 2008], [Kumar *et al.*, 2008])

ou encore en utilisant des algorithmes incrémentaux ([Williams *et al.*, 2007a], [Williams *et al.*, 2008], [Nistér et Stewénius, 2006], [Valgren *et al.*, 2007], [Filliat, 2007], [Angeli *et al.*, 2008b], [Angeli *et al.*, 2008c], [Angeli *et al.*, 2008a]). Ces dernières années, le problème de l'aliasing perceptuel était au centre des études. Les auteurs [Cummins et Newman, 2008], [Angeli *et al.*, 2008b] et [Angeli *et al.*, 2008c], en développant des mécanismes robustes d'appariement entre les primitives des lieux visités, sont parvenus à obtenir une robustesse impressionnante face à l'aliasing perceptuel. L'étude de la robustesse de la détection de boucle dans les expérimentations à long-terme, telles celles de [Dayoub et Duckett, 2008] et [Dayoub *et al.*, 2009], semble être la caractéristique sur laquelle se focalise actuellement la recherche de l'amélioration des algorithmes de détection de fermeture de boucle. En ce qui concerne l'aspect de la robustesse à l'orientation du robot, les auteurs emploient le plus souvent une caméra panoramique ou omnidirectionnelle pour pallier le problème. La méthode fournit en général des résultats suffisants dans les contextes étudiés, souvent le déplacement de robots dans des environnements plans. Les deux limitations qui apparaissent sont :

- Les méthodes ne sont pas réutilisables dans le cadre de robots de type drone.
- Les méthodes se contentent d'utiliser l'information supplémentaire fournie par un champ de vision plus large mais n'exploitent pas les propriétés de la représentation.

La dépendance à l'orientation du robot pour la détection de fermeture de boucle provient du fait que les primitives extraites des images sont dépendantes du point de vue du robot. L'exemple le plus simple est de considérer l'expérience où le robot fait le tour toujours dans le même sens d'un ensemble de maisons. Le mécanisme de détection de fermeture de boucle fonctionnera alors très bien. Par contre, si le robot fait d'abord un tour dans un sens puis revient en sens inverse, alors peu ou pas de fermetures de boucle vont être détectées. Cela se conçoit assez bien en comprenant que, dans ce cas, il s'agit d'un effet miroir. L'image 1.12 permet de visualiser des primitives lorsque le robot passe pour la première fois dans un sens et ces mêmes primitives vues par le robot lorsqu'il revient mais en sens inverse. Ce que nous considérons naturellement comme étant les mêmes primitives sont fait des primitives différentes. Il aurait fallu pour qu'elles soient identiques prendre en compte la symétrie perçue mais dépendante de l'environnement lui-même et de l'orientation du robot vis-à-vis de ces primitives. Il s'agit d'un exemple simple où il est facile de comprendre le problème. En reportant ce cas à l'ensemble des orientations possibles et des configurations des primitives envisageables, nous voyons

aisément l'impact du problème posé. L'utilisation de caméras panoramiques ou omnidirectionnelles permet d'enregistrer plus d'information provenant de l'environnement et donc d'enregistrer les primitives sous différents points de vue.

Dans le cadre de cette première partie de thèse, notre objectif est d'utiliser les acquis en termes d'algorithmes temps-réel et robustes face à l'aliasing perceptuel et d'y ajouter la robustesse face à l'orientation du robot. Les travaux de [Angeli *et al.*, 2008b] et [Angeli *et al.*, 2008c] nous servent de base pour les acquis. En ce qui concerne l'ajout de la robustesse à l'orientation du robot, nous utilisons la représentation sphérique ego-centrée de l'environnement abordée dans le chapitre suivant. L'ajout de cette représentation nous permet d'utiliser efficacement les propriétés spécifiques aux capteurs à large champ de vision telles les caméras panoramiques et omnidirectionnelles. L'algorithme final est utilisable pour les robots mobiles mais aussi pour les robots de type drone nécessitant une invariance à l'orientation suivant les trois axes de rotation.

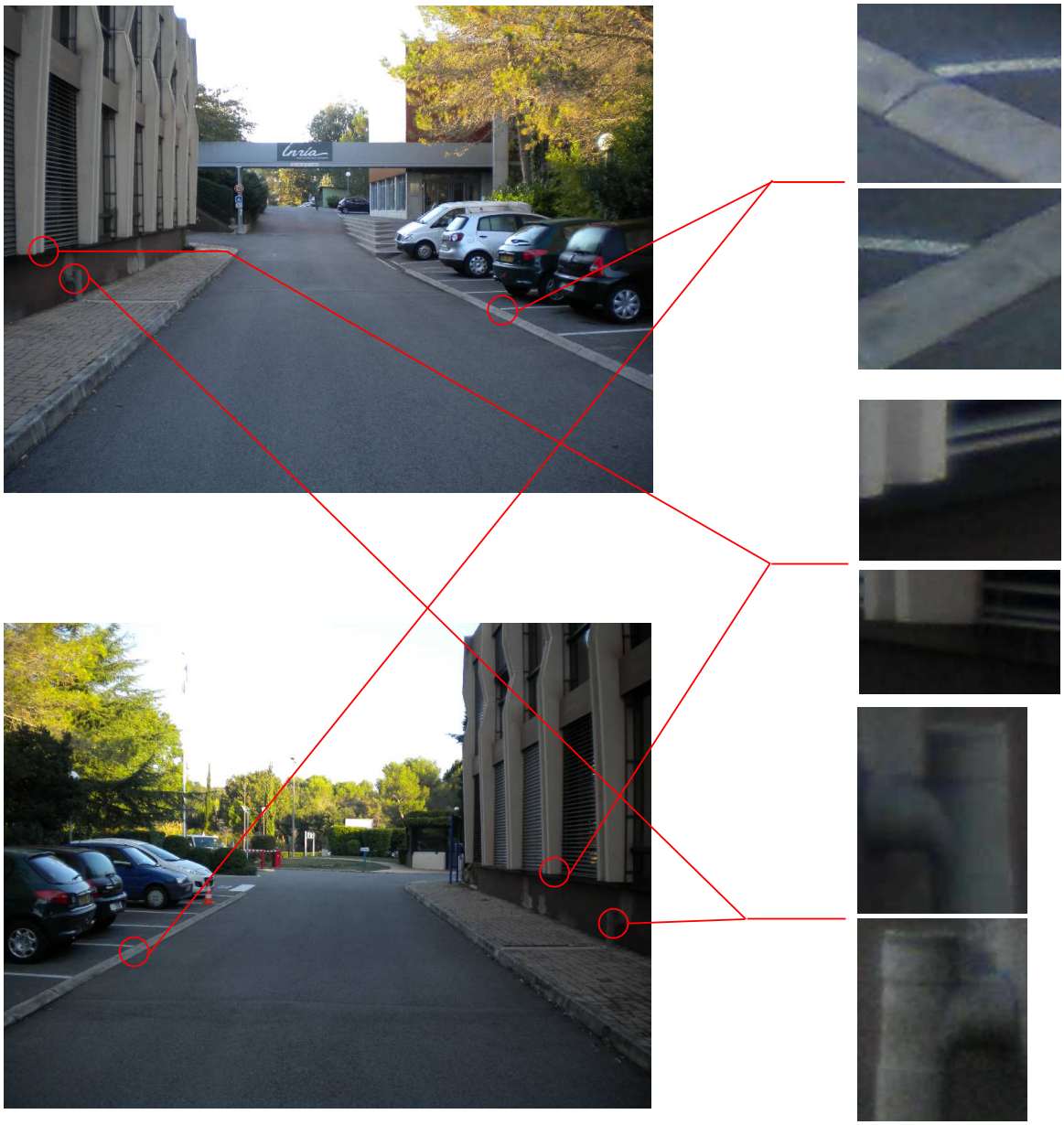


FIG. 1.12 – L'image du haut et l'image du bas sont deux images prises au même endroit mais en sens opposé l'une de l'autre. Trois primitives « identiques » sont repérées manuellement dans les deux images et mises en correspondance. Sur la partie droite, un zoom est effectué sur le contenu de chacune des primitives. Les primitives sont groupées par lot de deux, à chaque fois contenant en haut la primitive provenant de l'image du haut et en dessous la primitive correspondante provenant de l'image du bas. En comparant les contenus des primitives associées, nous observons que l'information est très différente, d'où l'impossibilité de les apparier dans un algorithme.

Chapitre 2

Représentation de l'environnement

Sommaire

2.1 Introduction

L'état de l'art sur la détection de fermeture de boucle a permis de mettre en avant l'importance de la représentation de l'environnement au sein des algorithmes. En effet, cela influe sur les caractéristiques de l'algorithme en terme de capacité de détection mais aussi sur la robustesse, la fiabilité et le temps de calcul. Dans le cadre de cette thèse, seules les représentations visuelles sont considérées et plus particulièrement la représentation sphérique (vue sphérique). Comme cela est expliqué dans la section 3.2, cette représentation permet de faire de la détection visuelle de fermeture de boucle indépendamment de l'orientation du robot. L'objectif de ce chapitre est alors de présenter le modèle de représentation sphérique de l'environnement. Ensuite, sont exposées les méthodes de description de l'information contenue dans une image. Le modèle de représentation du « sac de mots visuels » est présenté dans la dernière section.

2.2 Le modèle de représentation sphérique

2.2.1 Systèmes existants

2.2.1.1 Sphère photométrique panoramique

Alors que les travaux de [Krishnan et Nayar, 2009] présentent un vrai capteur sphérique, l'acquisition d'images panoramiques de bonne qualité est encore un problème non résolu. En effet, seule la construction de capteurs planaires (CCD ou CMOS) est maitrisée, et donc de caméras à champ de vue limité. Bien que certains objectifs permettent l'acquisition d'images très grand angle (*e.g.* objectif fisheye, *cf.* 2.1), la vision panoramique à 360° est d'une manière générale simulée. Deux techniques sont majoritairement utilisées : les caméras omnidirectionnelles catadioptriques et les systèmes multi-caméras.

2.2.1.2 Caméra omnidirectionnelle catadioptrique

Les caméras omnidirectionnelles [Nayar, 1997] permettent d'acquérir des images avec un champ de vision horizontal de 360° et avec un centre de projection unique. Ce capteur est en général composé d'une caméra perspective classique ou orthographique, à laquelle vient se greffer un miroir convexe de forme le plus souvent hyperbolique ou parabolique (voir figure 2.2), placé dans l'axe optique de la caméra.

Ce type de caméra peut être modélisé par le modèle de projection unifié proposé par [Mei et Rives, 2007], qui est une extension de [Geyer et Daniilidis, 2000]. Ce modèle effectue deux projections

FIG. 2.1 – Objectif grand angle et image fisheye.

FIG. 2.2 – Caméra omnidirectionnelle catadioptrique.

successives : une projection sphérique suivie d'une projection perspective (*cf.* figure 2.3). Un point $\mathbf{P} \in \mathbb{R}^3$ de l'espace Euclidien est projeté sur la sphère unitaire par une projection sphérique :

$$\mathbf{q}_S = \frac{\mathbf{P}}{\|\mathbf{P}\|} \tag{2.1}$$

Le point $\mathbf{q}_S$ est exprimé dans le repère $\mathcal{F}_M$ par :

$$\mathbf{q}_M = \mathbf{q}_S + \mathbf{e}_3\xi \tag{2.2}$$

où le paramètre $\xi \in [0;1]$ dépend de la géométrie du miroir. Le point $\mathbf{q}_M$ est alors projeté dans le

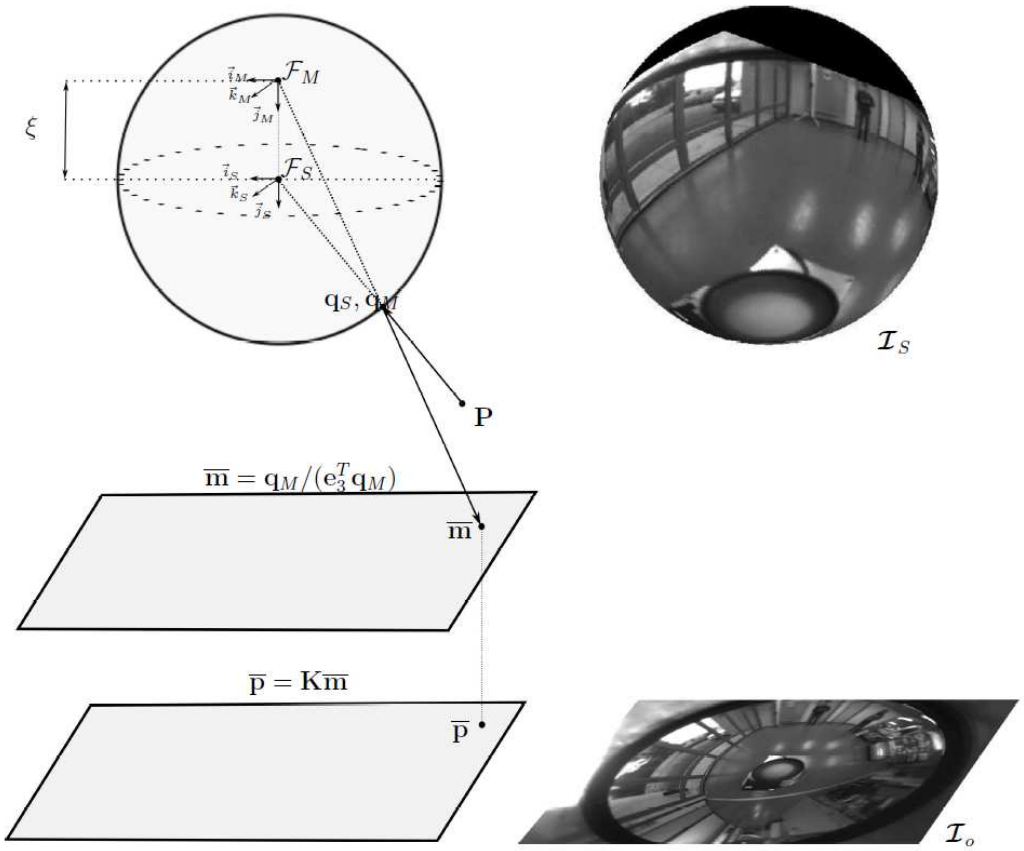


FIG. 2.3 – Modèle de projection unifié.

plan normalisé de la caméra par :

$$\overline{\mathbf{m}} = \frac{\mathbf{q}_M}{\mathbf{e}_3^T \mathbf{q}_M} \tag{2.3}$$

et dans le plan image normalisé par :

$$\overline{\mathbf{p}} = \mathbf{K}\overline{\mathbf{m}} = \begin{bmatrix} \gamma_1 & \gamma_1 s & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \overline{\mathbf{m}} \tag{2.4}$$

où $\mathbf{K}$ est la matrice des paramètres intrinsèques de la caméra et (γ_1, γ_2) sont les distances focales généralisées et dépendent de la forme du miroir. Les valeurs théoriques de γ et ξ sont détaillées dans [Mei et Rives, 2007], et peuvent être déterminées lors d'une phase de calibration.

L'image omnidirectionnelle $\boldsymbol{\mathcal{I}}_o$ peut ainsi être re-projetée sur une image sphérique $\boldsymbol{\mathcal{I}}_S$ au moyen d'un warping stéréographique inverse :

$$\boldsymbol{\mathcal{I}}_S(\mathbf{q}_S) = \boldsymbol{\mathcal{I}}_o(w(\mathbf{K}, \xi, \mathbf{q}_S)) \tag{2.5}$$

Les inconvénients majeurs de ce genre de caméras sont, d'une part, une faible résolution (360° sont projetés sur un seul capteur perspectif) et, d'autre part, une résolution spatiale non uniforme : la qualité de l'image diminue en direction des bords du capteur (*cf.* figure 2.3). De plus, en fonction de la forme du miroir, la largeur en champ vertical est limitée (demi-sphère, $< 90°$), ce qui n'est pas idéal pour cartographier des environnements urbains : la partie saillante et stable de l'information se trouvant souvent en hauteur (façade des bâtiments).

2.2.1.3 Systèmes multi-cameras

Il est également possible de construire une image panoramique assemblée à partir de plusieurs images capturées simultanément par plusieurs caméras reliées rigidement [Baker *et al.*, 2001] ou issues d'une séquence d'images [Lovegrove et Davison, 2010]. Les images, dont la position doit être parfaitement connue sont alors recalées, projetées et fusionnées sur une sphère virtuelle tangente aux capteurs par une technique de *mosaicing* [Szeliski, 2006]. Les N images $\boldsymbol{\mathcal{I}}_i$ peuvent être transformées et fusionnées sur une sphère par une fonction de warping des intensités des images perspectives vers l'image sphérique $\boldsymbol{\mathcal{I}}_S$:

$$\boldsymbol{\mathcal{I}}_S(\mathbf{q}_S) = \boldsymbol{\alpha}_1 \boldsymbol{\mathcal{I}}_1\Big(w(\mathbf{K}_1, \mathbf{R}_1, \mathbf{q}_S)\Big) + \ldots + \boldsymbol{\alpha}_N \boldsymbol{\mathcal{I}}_N\Big(w(\mathbf{K}_N, \mathbf{R}_N, \mathbf{q}_S)\Big) \tag{2.6}$$

où les coefficients $\boldsymbol{\alpha}$ sont les coefficients de fusion des intensités, les matrices $\mathbf{K}_i$ les paramètres intrinsèques des caméras et les matrices $\mathbf{R}_i$ représentent la rotation des images par rapport à la sphère. Puisque l'information de profondeur n'est pas connue, les translations $\mathbf{t}_i$, entre les images et la sphère sont obligatoirement négligées. La fonction $\overline{\mathbf{p}} = w(\mathbf{K}, \mathbf{R}, \mathbf{q_S})$ transfère le point de la sphère unitaire $\mathbf{q}_S \in \mathcal{S}^2$ dans l'image par une projection perspective (*cf.* figure 2.4(c)) :

$$\overline{\mathbf{p}} = \frac{\mathbf{K}\mathbf{R}\mathbf{q}_S}{\mathbf{e}_3^T \mathbf{K}\mathbf{R}\mathbf{q}_S} \tag{2.7}$$

Grâce à l'utilisation de plusieurs images issues de capteurs perspectifs, ce genre de technique permet de construire des images sphériques de très grande résolution (> 10 millions de pixels). Néanmoins, le fait de négliger les translations, revient à assumer un centre de projection commun à toutes les caméras afin d'aligner les images en rotation uniquement (la rotation étant indépendante de la géométrie de la scène). Dans ce cas les centres optiques doivent être le plus proche possible les uns des autres, ce qui

peut être problématique en terme de conception mécanique car le centre optique d'une caméra est un point virtuel.

Dans certains cas, notamment lorsque des objets de la scène sont proches des capteurs, la translation entre les centres optiques n'est pas négligeable. L'hypothèse du centre de projection unique n'est alors pas valable : des artéfacts liés aux effets de parallaxe sont visibles dans les images panoramiques reconstruites.

Pour minimiser cet effet de parallaxe, [Li, 2006] a proposé une caméra sphérique composée de deux objectifs fisheye placés dos à dos. L'image finale est formée sur un seul capteur à l'aide d'un miroir. Ce système permet de minimiser la translation entre les deux caméras virtuelles et ainsi obtenir un centre de projection quasiment unique. Cependant, le fait de n'utiliser qu'un seul capteur ne permet pas d'obtenir des sphères de grande résolution.

2.2.2 Système d'acquisition de sphères

2.2.2.1 Système de caméras à multi-baselines

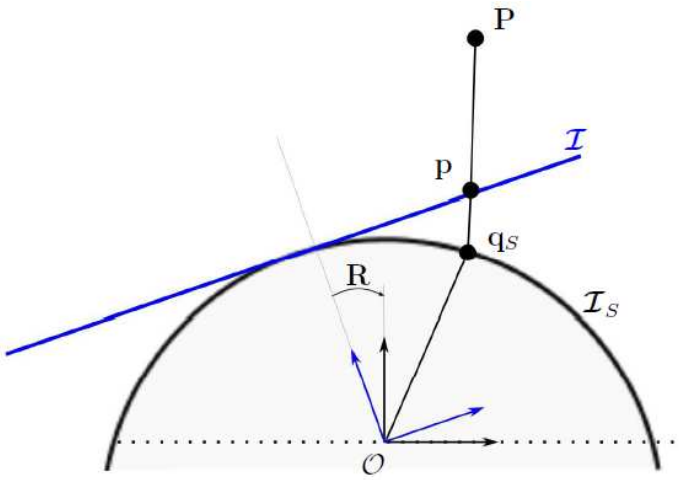
Le système d'acquisition développé par [Meilland *et al.*, 2010], [Meilland *et al.*, 2011], utilise six caméras grand angle placées sur un cercle dont les centres optiques sont éloignés volontairement les uns des autres. Contrairement aux systèmes multi-caméras classiques, cette configuration permet de générer de la disparité entre chaque image et ainsi utiliser des techniques de mise en correspondance dense stéréo pour extraire la profondeur, directement sur les images du système. Cette information est d'une part indispensable pour une localisation à 6 degrés de liberté, et d'autre part pour créer une sphère à centre de projection unique. En effet, l'information de profondeur permet de reprojeter correctement la photométrie issue des images, ce qui évite les artéfacts liés aux effets de parallaxe dont souffrent les systèmes panoramiques multi-caméras classiques.

La figure 2.5 montre le système monté sur un véhicule. Dans cette configuration, les 360° du champ de vision horizontal sont visibles par le système. Grâce à la large baseline séparant chaque caméra, et aux objectifs grands angles utilisés, une disparité peut être extraite dans chaque image.

Pour plus de renseignements sur le système, notamment l'estimation de profondeur dans l'image et la reconstruction dense de l'image en prenant en compte l'information de profondeur, le lecteur intéressé pourra se rapporter à la thèse de A. Meilland [Meilland, 2012]. Nous avons simplement présenté le concept de représentation sphérique et la méthode de construction de la sphère par mosaicing étant donné que les algorithmes développés dans cette thèse sont basés dessus.

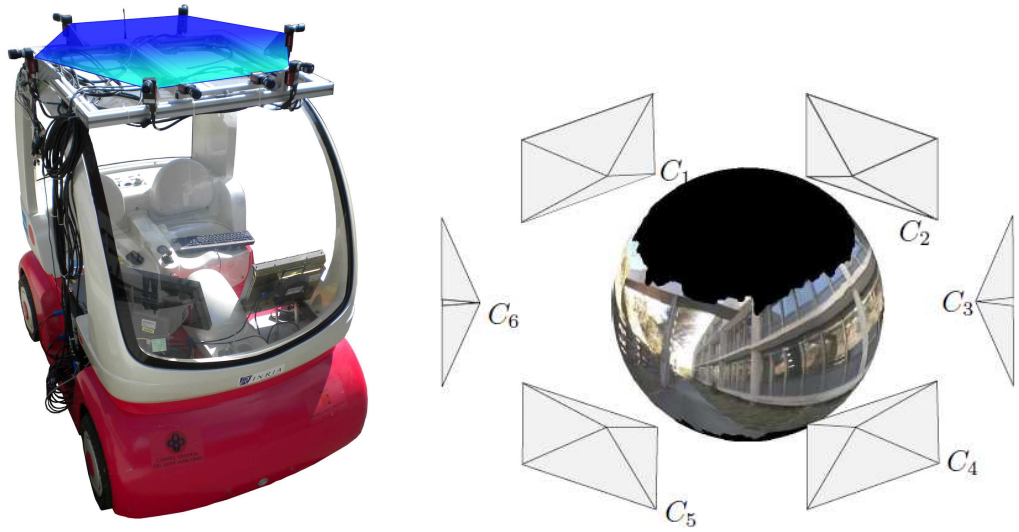
(a) Caméras sphériques. *PointGrey,[Pfeil* et al.*, 2011] et Nikon.*

(b) Image panoramique. *LadyBug PointGrey.*



(c) *Mosaicing*

FIG. 2.4 – (a) : Exemples de systèmes d'acquisition d'images panoramiques multi-caméra .(b) Image panoramique reconstruite. (c) : Transformation d'une image perspective sur la sphère.



(a) Système monté sur un véhicule. (b) Sphere reconstruite.

FIG. 2.5 – Système d'acquisition de sphères. La disposition hexagonale des caméras permet l'utilisation de techniques de mise en correspondance dense.

2.2.2.2 Étalonnage

Avant de pouvoir effectuer la mise en correspondance dense et ainsi reconstruire des sphères visuelles, il est important de calibrer le système. C'est à dire extraire les paramètres extrinsèques (position relative des caméras) et les paramètres intrinsèques des caméras (focale, centre optique, polynôme de distorsions). Le système multi-caméras peut être représenté comme 6 paires de caméras stéréo, où chaque paire stéréo est reliée rigidement à la paire suivante.

La particularité de ce dispositif est que l'angle formé entre les axes optiques de chaque caméra est divergent (60°). Dans ces conditions, une mire classique de calibration (échiquier) ne peut être observée que par deux caméras simultanément, ce qui ne permet pas d'utiliser des techniques de calibration multi-caméras telles que [Svoboda *et al.*, 2005], [Zaharescu *et al.*, 2006] qui assument l'objet de calibration visible par toutes les caméras. D'autres techniques [Li, 2006] utilisent une seule mire rigide englobant le système afin de calibrer toutes les caméras simultanément. Cependant ce genre d'approche est difficilement applicable à ce système, en particulier à cause de l'échelle : une mire rigide de plusieurs mètres est nécessaire.

Afin d'assurer une mise en œuvre simple, la méthode de calibration utilise une mire simple (objet plan). La technique la plus basique consiste à estimer successivement les paramètres extrinsèques des caméras avec une calibration stéréo classique [Bouguet, 2000]. Dans ce cas, les erreurs sont cumulées,

résultant en une calibration inconsistante : la dernière paire stéréo contiendra toute la dérive intégrée sur chaque paire stéréo.

Cependant la configuration circulaire du système présente une fermeture de boucle (*cf.* Figure 2.6). Dans ces conditions il est possible de formuler le problème en une optimisation globale des paramètres extrinsèques du système ainsi que des poses des mires de calibration. Cela permet de corriger la dérive et de répartir correctement les erreurs de reprojection sur toutes les caméras.

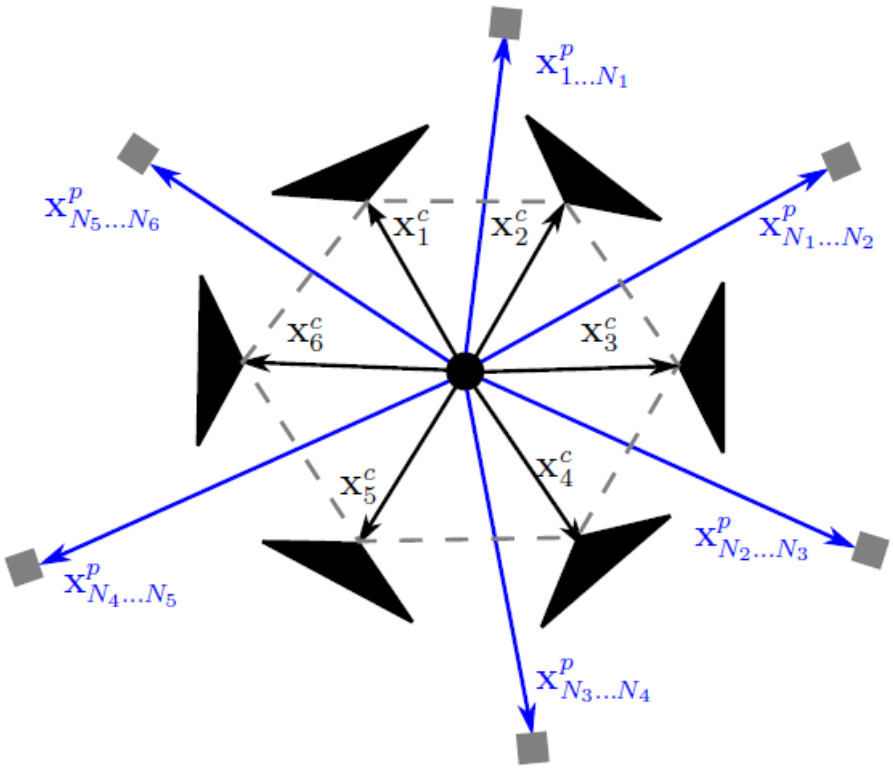


FIG. 2.6 – Calibration du système d'acquisition sphérique.

Le vecteur des inconnues du système est défini tel que :

$$\mathbf{x}^{\Sigma} = \left(\mathbf{x}^c_1, ..., \mathbf{x}^c_M, \mathbf{x}^p_1, ..., \mathbf{x}^p_N\right)^{\top} \tag{2.8}$$

où $\mathbf{x}^c_M$ représente les poses des caméras ($M = 6$) et $\mathbf{x}^p_N$ représente les N poses des mires.

Le critère d'optimisation global est défini (avec abus de notation) par l'erreur entre le vecteur des points de la mire projetée $w(\boldsymbol{\mathcal{P}}_p)$ et le vecteur des points détectés dans les images $\boldsymbol{\mathcal{P}}_m$:

$$\mathbf{e}(i,j) = \boldsymbol{\mathcal{P}}_m - w\left(\mathbf{T}(\mathbf{x}^c_i)\mathbf{T}(\mathbf{x}^p_j), \boldsymbol{\xi}_i; \boldsymbol{\mathcal{P}}_p, \boldsymbol{\mathcal{Z}}_p\right) \tag{2.9}$$

où i et j sont respectivement l'indice de la caméra et l'indice de la mire (voir Figure. 2.6). La matrice $\mathbf{K}(\boldsymbol{\xi}_i) \in \mathbb{R}^{3\times3}$ contient les paramètres intrinsèques de la caméra i. Dans ce cas, la fonction $w(.)$ est une projection perspective qui transfère les points de la mire j sur la caméra i :

$$\overline{\mathbf{p}} = \frac{\boldsymbol{K} \left[\begin{array}{cc} \mathbf{R} & \mathbf{t} \end{array} \right] \mathbf{P}}{\mathbf{e}_3^T \boldsymbol{K} \left[\begin{array}{cc} \mathbf{R} & \mathbf{t} \end{array} \right] \mathbf{P}} \tag{2.10}$$

où $\mathbf{P}$ est un point 3D Euclidien appartenant à la mire de calibration.

À partir de cette fonction d'erreur, il est possible de trouver un jeu de paramètres optimal $\widehat{\mathbf{x}}^{\Sigma}$ en minimisant l'erreur de re-projection pour chaque caméra et chaque mire :

$$\widehat{\mathbf{x}}^{\Sigma} = \arg\min \mathbf{x}^{\Sigma} \sum_{i=1}^{6} \left(\sum_{j=1}^{N} \|\mathbf{e}(i,j)\|^2 . \eta(i,j) \right) \tag{2.11}$$

$$\eta(i,j) = \left\{ \begin{array}{ll} 1 & \text{si la mire j est vue par la caméra i} \\ 0 & \text{sinon} \end{array} \right.$$

Minimiser itérativement la fonction de coût 2.11 permet d'estimer la pose de chaque caméra $\mathbf{x}_i^c$ en respectant la contrainte de fermeture de boucle. En pratique, afin d'éviter certains minimas locaux, la minimisation est initialisée avec les paramètres extrinsèques obtenus successivement par calibration stéréo [Bouguet, 2000]. Les paramètres intrinsèques $\boldsymbol{\xi}_k$ quand à eux peuvent être obtenus indépendamment et précisément pour chaque caméra, ils ne sont donc pas re-estimés dans la minimisation.

2.3 Description de l'information contenue dans l'image

Cette section se concentre sur une présentation générale de l'existant en terme d'extraction de primitives à partir d'une image. Cette présentation est non exhaustive et ne constitue pas un état de l'art. L'intérêt de cette partie est qu'elle permet d'introduire et d'expliquer le principe des primitives locales comme le SIFT [Lowe, 1999] (que nous utilisons dans cette première partie de thèse) et des primitives globales comme celle que nous créons dans notre système de détection de fermeture de boucle ou le GIST [Oliva et Torralba, 2001] (que nous utilisons dans la deuxième partie). Comme évoqué à maintes reprises dans l'état de l'art sur la détection visuelle de fermeture de boucle, les algorithmes reposent sur l'extraction de primitives à partir des images. L'importance des primitives visuelles pour la reconnaissance est illustrée par le simple exemple de la figure 2.7. Ces primitives

sont ensuite appariées suivant une mesure de similarité afin de déterminer les primitives semblables. L'extraction de primitives de l'image repose sur l'utilisation d'un détecteur qui est en charge de trouver les points d'intérêt de l'image suivant un certain nombre de critères. Une fois les points d'intérêt déterminés, ils sont associés à un descripteur qui décrit l'information de la primitive. L'étape de comparaison sur base de mesure de similarité s'effectue sur les descripteurs des primitives. Ce mécanisme est valable dans le cas de primitives locales qu'il faut identifier dans une image. Dans le cas de primitives globales, il n'y a pas de détecteur mais seulement un mécanisme d'élaboration du descripteur. Au travers de cette présentation nous aborderons les qualités requises pour élaborer un détecteur efficace. Quelques détecteurs intéressants sont présentés afin de comparer leurs performances vis-à-vis des qualités évoquées. Pour une étude plus détaillée des primitives locales, le lecteur intéressé pourra se référer aux travaux de [Tuytelaars et Mikolajczyk, 2008].

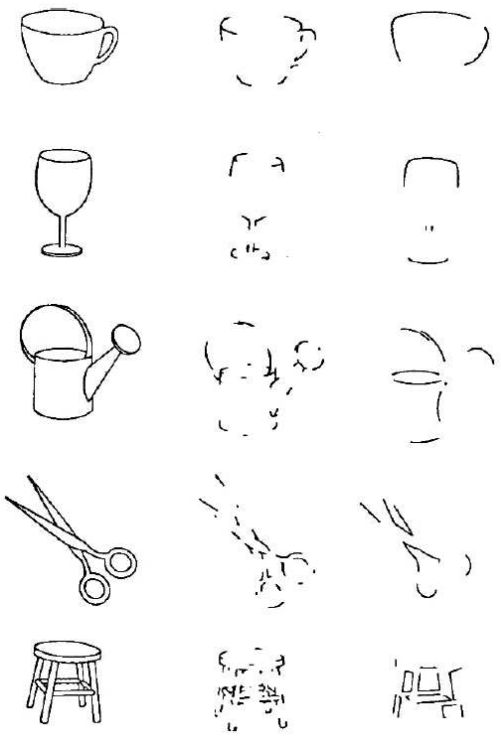


FIG. 2.7 – Illustration de l'importance des coins et des jonctions pour la reconnaissance d'objets. Source : [Biederman, 1987].

2.3.1 Propriétés du détecteur « idéal »

Tel que le définissent les auteurs [Tuytelaars et Mikolajczyk, 2008], un détecteur « idéal » doit satisfaire les propriétés suivantes :

- **Répétabilité :** étant données deux images de la même scène, suivant différentes conditions de vue, un pourcentage élevé des primitives détectées dans la partie visible de l'image doit être détecté dans les deux images. Il s'agit de la propriété la plus importante, elle peut être obtenue de deux manières :

  Invariance : lorsque des déformations importantes sont attendues, il est préférable de les modéliser mathématiquement. Ensuite, il est nécessaire de développer des méthodes de détection de primitives qui ne soient pas affectées par ces transformations mathématiques.

  Robustesse : dans le cadre de petites déformations, il est souvent suffisant de rendre les méthodes de détection de primitives moins sensibles à ces déformations ; ainsi, la précision est réduite mais pas de manière drastique. Les déformations typiques qui sont traitées par la robustesse sont le bruit, les effets de discrétisation, les artéfacts de compression, le flou. De même, les déviations photométriques et géométriques des modèles mathématiques sont souvent corrigées en incluant plus de robustesse.
- **Distinctivité/ information :** les primitives doivent présenter suffisamment de variations afin de pouvoir être distinguées et mises en correspondance.
- **Localité :** les primitives doivent être au maximum locales afin de réduire les risques d'occlusion et de permettre des modèles simples d'approximations des déformations géométriques et photométriques entre deux images prises dans des conditions de vue différentes. (Hypothèse de planéité locale)
- **Quantité :** le nombre de primitives détectées doit être suffisamment important pour qu'un nombre raisonnable de primitives soit aussi détecté même sur de petits objets. Idéalement, le nombre de primitives détectées devrait être paramétrable sur une large plage par l'intermédiaire d'un simple seuillage. La densité de primitives devrait refléter l'information contenue dans l'image afin de donner une représentation compacte de cette image.
- **Précision :** les primitives détectées doivent être précisément localisées dans l'image quelque soit l'échelle et la forme.
- **Efficacité :** la détection des primitives dans une nouvelle image devrait pouvoir être faite en temps-réel.

Bien que cette définition ait été établie pour les primitives locales, les mêmes propriétés (outre celles de localité, quantité et de précision) doivent être vérifiables pour établir un descripteur global performant.

2.3.2 Présentation des détecteurs les plus classiques

Les détecteurs les plus classiques, et aussi les plus souvent utilisés du fait de leurs bonnes propriétés d'invariance, sont rapidement présentés dans cette section.

2.3.2.1 Points de Harris / FAST

Les détecteurs de Harris [Harris et Stephens, 1988], SUSAN [Smith et Brady, 1997] et FAST [Rosten et Drummond, 2006] sont des détecteurs de coins. Le premier, le plus connu et le plus utilisé, fonctionne suivant une méthode de calcul de gradient par l'intermédiaire de matrices d'autocorrélation. Le mécanisme de détection est illustré sur la figure 2.8. Il présente l'avantage d'être invariant à la translation, à la rotation et aux changements de luminosité. De même, il présente des taux de répétabilité et d'information très élevés. Toutefois, il reste sensible au bruit et au changement de point de vue (points qu'essaie de corriger le détecteur SUSAN sans obtenir pour autant une robustesse élevée).
Le détecteur FAST (Features from Accelerated Segment Test) est basé sur une approche similaire au détecteur SUSAN. Le mécanisme repose sur la comparaison de l'intensité d'un pixel central avec l'ensemble des intensités des pixels d'un voisinage proche résultant en une fraction du nombre de pixels similaires par rapport au nombre de pixels considérés. Le résultat contient une information importante quant à la structure locale de l'image. La conception du détecteur FAST est plus élaborée de façon à pouvoir être utilisé en temps réel. Les résultats obtenus avec ce détecteur sont tout à fait satisfaisants ; malgré le fait qu'il faille souvent faire un choix entre la précision et le temps de calcul. Les résultats sont aussi bons voire meilleurs que des approches plus courantes.

2.3.2.2 DoG / Laplace / SIFT

Sans décrire précisément chacun de ces détecteurs, ils sont tous les trois basés sur le même principe : une analyse à partir de la différence de Gaussiennes (DoG : Différence of Gaussians) afin de pouvoir obtenir des primitives invariantes à l'échelle. Comme cela a été prouvé dans [Lindeberg, 1994], sous réserve de certaines hypothèses assez générales, les noyaux Gaussiens et leurs dérivées sont les seuls noyaux lissants possibles pour une analyse des échelles. Les détecteurs DoG et Laplace sont des détecteurs dont le but essentiel est de passer dans l'espace des échelles afin d'apporter une in-

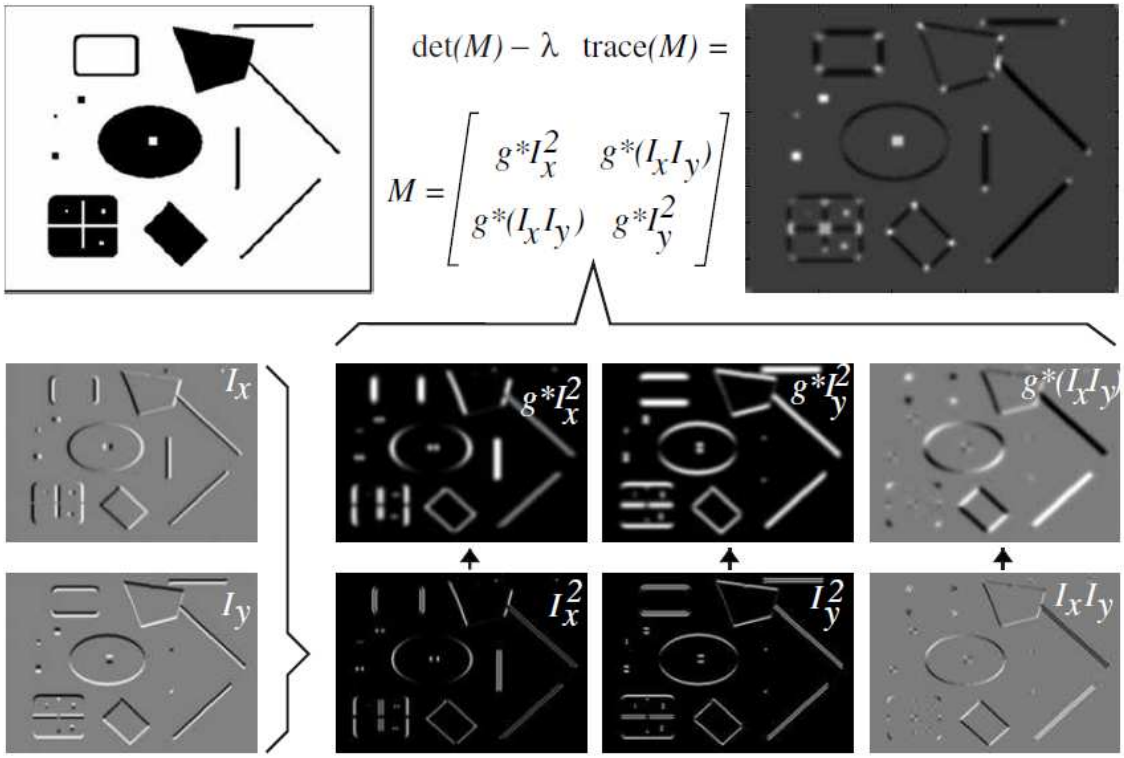


FIG. 2.8 – Illustration du mécanisme de détection des coins de Harris. Source : [Tuytelaars et Mikolajczyk, 2008].

variance à celle-ci. Ils sont alors souvent combinés avec un autre détecteur d'où certains noms tels que le détecteur Harris-Laplace. SIFT [Lowe, 1999], quant à lui, est un détecteur basé sur la détection d'angles (détecteur de Harris) sur lequel est ajoutée une approche DoG pour l'invariance à l'échelle. Un de ses points forts est son mode de fonctionnement de type simulation. Ceci lui confère une approche moins mathématique, formelle, et donc plus de souplesse (robustesse telle que vu précédemment). Les points de Harris sont détectés à plusieurs échelles et ensuite mis en relation via des principes de gradient d'intensité et d'orientation du gradient. Outre cette invariance à l'échelle, il présente aussi une certaine invariance à la rotation suivant les trois axes (ceci obtenu par l'enregistrement de l'orientation du gradient des points de Harris). Le processus de détermination du descripteur est illustré en figure 2.9. Ceci fait de ce détecteur l'un des plus utilisés actuellement. Toutefois, bien que le calcul soit rapide pour un petit nombre de points, pour un nombre important de primitives, le calcul devient long et ne permet pas de faire du temps réel.

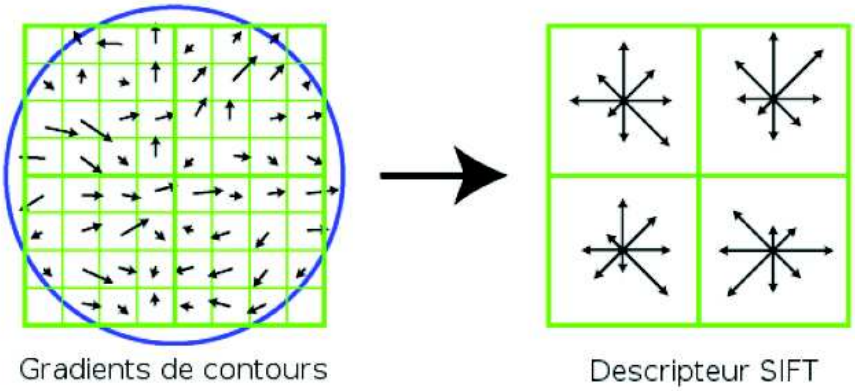


FIG. 2.9 – Illustration du principe de construction du descripteur SIFT. Le descripteur est construit en deux étapes. Dans un premier temps, les orientations et les amplitudes des gradients pris aux alentours d'un point d'intérêt dans l'image (partie gauche de la figure) sont calculées. Ces informations sont alors pondérées par des coefficients Gaussiens (le cercle sert à délimiter la zone où ces coefficients sont non nuls), avant d'être accumulées sous la forme d'histogrammes d'orientations regroupant l'information par sous-régions de 4x4 pixels. Le résultat de cette accumulation est illustré dans la partie droite de la figure, où la taille de chaque flèche dépend des amplitudes des gradients. Pour les besoins de la figure, seulement 4 sous-régions de 4x4 pixels chacune sont montrées, alors que normalement 16 sous-régions de cette taille sont utilisées. Source : [Lowe, 2004].

2.3.2.3 SURF

Le détecteur SURF (Speeded Up Robust Features) [Bay *et al.*, 2006] est une amélioration du détecteur SIFT. En effet, il pallie le manque d'efficience de ce dernier en modifiant le calcul de la différence de Gaussiennes long à effectuer. Il est remplacé par une approximation de la matrice Hessienne d'un noyau Gaussien calculée rapidement à partir du principe de l'image intégrale (illustrée figure 2.10) introduite par [Viola et Jones, 2001]. L'enregistrement de l'information pour le calcul du descripteur est aussi différent : l'orientation de la primitive est analysée au travers d'un échantillonnage du voisinage sous forme circulaire. La primitive est ainsi rendue invariante à la rotation. Le résultat est soumis à une transformée en ondelettes de Haar. De manière générale, le détecteur SURF présente les mêmes avantages que le SIFT. L'approximation de la matrice Hessienne n'engendre pas d'erreur importante d'estimation et permet un calcul en temps réel.

2.3.2.4 SuperPixel / MSER

Le détecteur SuperPixel ([Ren et Malik, 2003], [Mori *et al.*, 2004]), tel que décrit dans [Tuytelaars et Mikolajczyk, 2008], est un détecteur de régions nommées superpixels et obtenues par segmentation. L'image est alors considérée comme un ensemble de superpixels ; il y a abstraction de l'unité de bas

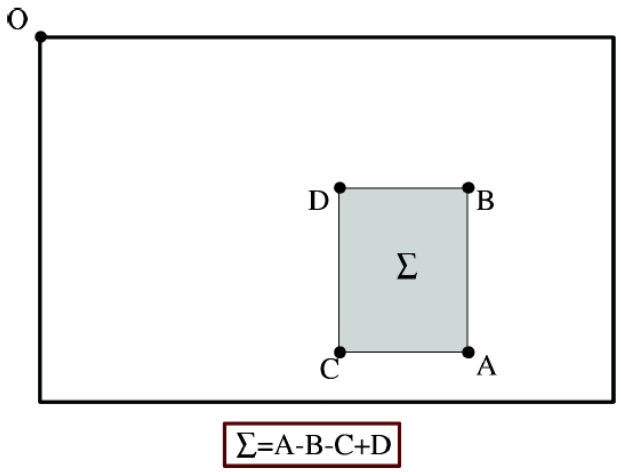


FIG. 2.10 – En utilisant les images intégrales, seulement 4 opérations sont nécessaires pour calculer l'intégrale des intensités d'une région rectangulaire de n'importe quelle taille. Source : [Viola et Jones, 2001].

niveau qu'est le pixel. Ce détecteur présente l'avantage d'avoir un bon compromis entre la localité de la primitive et sa distinctivité (généralement basée sur un large voisinage). D'autre part, il est un des rares à couvrir toute l'image sans chevauchement entre les régions. Le résultat de la segmentation d'une image est illustré sur la figure 2.11. Toutefois, il n'est pas invariant à l'échelle et n'est pas adapté pour l'appariement ou la reconnaissance d'objets. Ce détecteur est surtout utilisé du fait que les régions obtenues ont beaucoup plus de sens que les pixels en eux mêmes, elles présentent donc un intérêt sémantique.

Le détecteur MSER (Maximally Stable Extremal Regions) [Matas *et al.*, 2004] est aussi un détecteur de régions. Basé sur l'intensité des pixels, il ne présente pas les mêmes avantages que le détecteur superpixel. En effet, il ne couvre pas toute l'image dans la solution apportée et ne garantit pas que les régions ne se chevauchent pas. Par contre, il est invariant aux transformations affines géométriques et photométriques. Il peut aussi être rendu invariant aux changements d'échelle. Cependant, comme le superpixel, il n'est pas adapté pour l'appariement ou la reconnaissance d'objets.

2.3.2.5 Synthèse des différents détecteurs classiques

Le tableau 2.1, extrait de [Tuytelaars et Mikolajczyk, 2008] et complété avec le détecteur SIFT, récapitule les performances des différents détecteurs de primitives locales.

Détecteur	Coins	Zones locales	Régions	Invariance à la rotation	Invariance à l'échelle	Invariance aux transformations affines	Répétabilité	Précision de la localisation	Robustesse	Efficience
Harris	✓			✓			+++	+++	+++	++
Hessian		✓		✓			++	++	++	+
SUSAN	✓			✓			++	++	++	+++
Harris-Laplace	✓	(✓)		✓	✓		+++	+++	++	+
Hessian-Laplace	(✓)	✓		✓	✓		+++	+++	+++	+
DoG	(✓)	✓		✓	✓		++	++	++	++
SIFT	(✓)	✓		✓	✓		+++	+++	+++	++
SURF	(✓)	✓		✓	✓		++	++	++	+++
Harris-Affine	✓	(✓)		✓	✓	✓	+++	+++	++	++
Hessian-Affine	(✓)	✓		✓	✓	✓	+++	+++	+++	++
Régions Saillantes	(✓)	✓		✓	✓	(✓)	+	+	++	+
Bordures	✓			✓	✓	✓	+++	+++	+	+
MSER			✓	✓	✓	✓	+++	+++	++	+++
Basé intensité			✓	✓	✓	✓	++	++	++	++
Superpixels			✓	✓	(✓)	(✓)	+	+	+	+

TAB. 2.1 – Synthèse des détecteurs classiques

FIG. 2.11 – Illustration de la segmentation d'une image en superpixels. Source : [Ren et Malik, 2003], [Mori *et al.*, 2004].

2.3.3 Détecteurs récents présentant de bonnes performances

2.3.3.1 ASIFT

Tel que décrit dans [Yu et Morel, 2009], le détecteur ASIFT (Affine SIFT) est une extension du détecteur SIFT [Lowe, 1999]. Basé sur le principe de simulation efficace du détecteur SIFT, ce détecteur étend la simulation aux transformations affines et notamment le changement de prise de vue. Les prises de vue sont alors simulées pour des angles échantillonnés et enregistrées sous forme de descripteur SIFT. Le détecteur, restant basé sur l'approche du SIFT, bénéficie de ses qualités. Il ajoute la robuste pour des angles de prises de vue allant jusqu'à 80° (*cf.* figure 2.12) ; élément très intéressant pour la répétabilité et la robustesse. Il permettrait alors de représenter une grande scène de manière simple et compacte tout en ayant un taux d'appariement élevé. Toutefois, il pâtit d'un inconvénient majeur : il ne peut pas s'exécuter en temps réel (le temps de simulation est trop important).

2.3.3.2 DAISY

DAISY, élaboré par les auteurs [Tola *et al.*, 2009], est un détecteur mixant les avantages de deux détecteurs très performants : SIFT [Lowe, 1999] et GLOH [Mikolajczyk et Schmid, 2005]. Ainsi, ce détecteur est basé sur un noyau permettant de créer une information invariante à l'échelle par l'in-

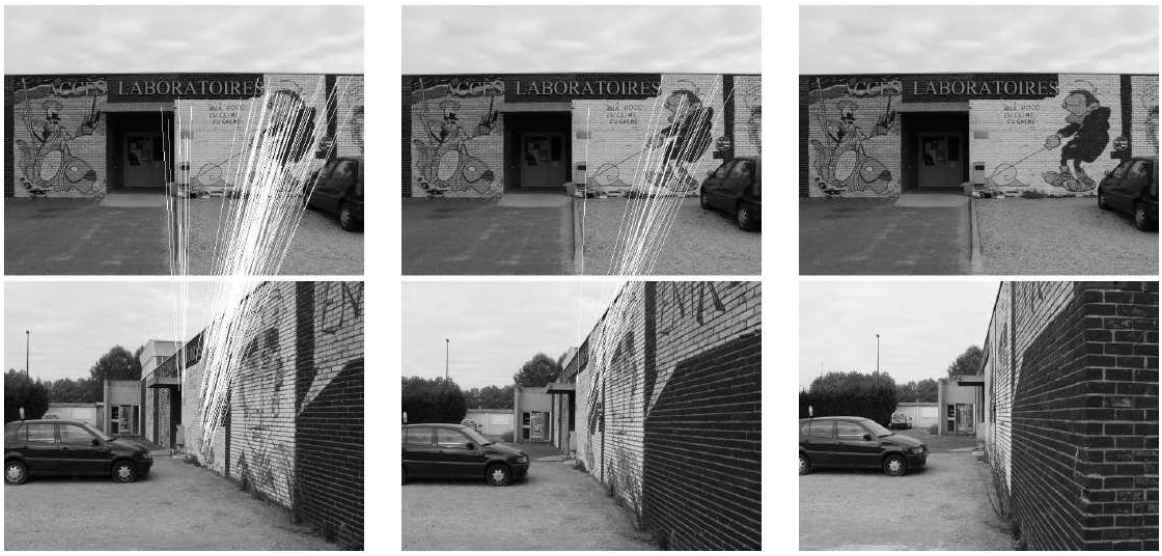


FIG. 2.12 – Illustration de la robustesse aux transformations affines du détecteur ASIFT. Source : [Yu et Morel, 2009].

termédiaire d'une analyse multi-échelle sur un voisinage proche du point étudié. Le mécanisme de calcul du descripteur est illustré sur la figure 2.13. De plus, il est invariant à la rotation et à la translation, plutôt résistant aux occlusions et aux transformations photométriques et géométriques. Il apporte donc un taux de répétabilité élevé adjoint d'une bonne information permettant la distinction tout en conservant la précision. Il s'agit d'un très bon compromis entre toutes les propriétés d'un détecteur. De plus, la précision du détecteur permet la reconstruction de cartes de profondeurs proches de la vérité terrain. Enfin, il améliore l'approche calculatoire du SIFT en utilisant des simplifications par convolution de Gaussiennes, réduisant le nombre d'opérations élémentaires nécessaires et ainsi son temps d'exécution ; DAISY peut donc être utilisé en temps réel.

À la différence de nombreux détecteurs vus précédemment, DAISY est un détecteur dense. C'est-à-dire qu'il s'agit d'une méthode de calcul d'un descripteur mais qu'il n'y a pas de détecteurs pour sélectionner les primitives dans l'image. Il est dense dans le sens où il est destiné à être calculé pour chacun des pixels de l'image (opération longue). Toutefois, il est possible de le combiner avec un détecteur de primitives locales comme Harris et ensuite de calculer les descripteurs aux localisations des points de Harris.

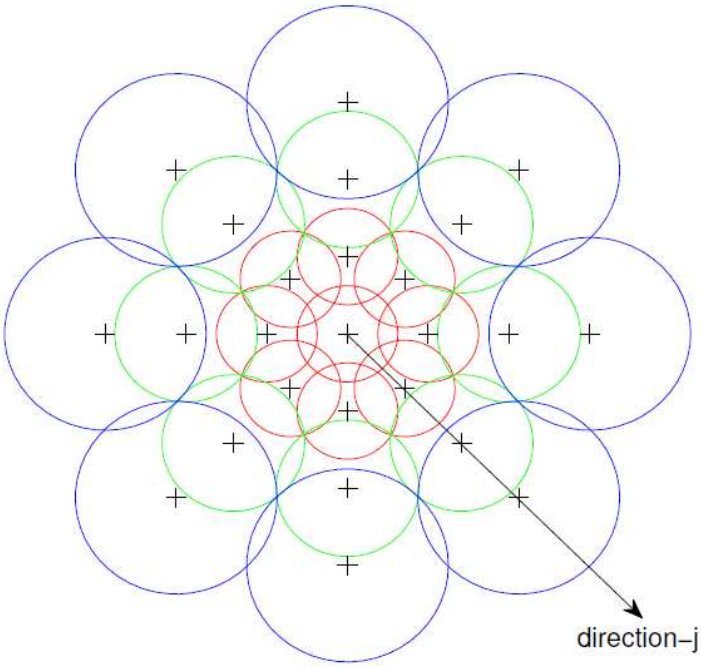


FIG. 2.13 – Le descripteur DAISY. Chaque cercle représente une région dont le rayon est proportionnel à l'écart-type du noyau gaussien. Le signe + indique les endroits où sont échantillonnés les centres des cartes convoluées d'orientations. Ces centres sont les endroits où le descripteur est calculé. La superposition des régions permet une transition lisse entre les régions et un certain degré de robustesse à la rotation. Les rayons des régions extérieures sont plus grands pour avoir un échantillonnage égal par rapport à l'axe de rotation. Ceci est nécessaire pour obtenir une robustesse face à la rotation. Source : [Tola *et al.*, 2009].

2.3.3.3 GIST

Tel que décrit dans [Oliva et Torralba, 2006] et [Oliva et Torralba, 2001], l'idée principale de ce détecteur global est d'extraire l'information la plus importante d'une image, ce qui est aperçu à première vue, sans un réel souci de précision. Son intérêt réside dans la forte interprétation sémantique de la scène. À partir des informations extraites par le détecteur, il est possible de différencier les lieux, voire scènes, de par leur sémantique plutôt que par des primitives invariantes à un certain nombre de transformations. Il en résulte qu'il est a priori le détecteur idéal pour faire de la localisation topologique. Toutefois, il est montré dans [Douze *et al.*, 2009] que le manque de précision de ce détecteur -grande information sémantique mais faible distinctivité locale, liée aux primitives- engendre des erreurs lors d'une recherche dans une base de données importante ; l'erreur étant qu'il permet d'obtenir une image proche de celle demandée mais pas l'image demandée en elle-même. La signature (descripteur) est

simple et de longueur constante quelque soit la taille de l'image et la quantité d'information. Le calcul du descripteur peut être fait en temps réel. Ce détecteur sera abordé plus en détails dans la section 6.2 du fait que nous l'ayons utilisé et modifié pour les besoins de notre algorithme de détection de changement de lieu topologique.

2.3.4 Les descripteurs

Les descripteurs sont la représentation de l'information obtenue à partir du détecteur. Chaque détecteur est généralement élaboré avec son propre descripteur car il est nécessaire lors de sa conception de pouvoir l'enregistrer en mémoire. Toutefois, certains descripteurs sont plus performants que d'autres et présentent un certain nombre d'avantages comme un temps de calcul restreint pour retrouver une primitive à partir de son descripteur. Par exemple, le descripteur SIFT encode l'information sous forme de clé et l'enregistre dans une table de hashage. Ceci permet de condenser l'information et de faire la recherche par clé ; il s'agit d'une méthode d'indexation qui diminue le temps de recherche et qui améliore ainsi le temps de calcul lors des phases d'appariement. Le descripteur GIST présente l'avantage d'être relativement petit et de taille constante pour n'importe quelle taille d'image et quelque soit la quantité d'information. Le descripteur associé au détecteur présente une importance dans les processus de représentation de l'image et de recherche de similarité. Il est alors nécessaire de trouver un compromis entre l'espace mémoire occupé, le temps de recherche et la distinctivité du descripteur (quantité d'information contenue nécessaire pour la différenciation) afin d'obtenir un résultat fiable en temps réel.

2.4 Le sac de mots visuels : une représentation de l'image

La méthode des sacs de mots visuels est une approche courante dans le cadre de la catégorisation d'images([Csurka *et al.*, 2004], [Nilsback et Zisserman, 2006], [Nistér et Stewénius, 2006]). Elle repose sur une représentation des images sous la forme d'un ensemble non-ordonné de primitives locales, les mots, choisies à partir d'un dictionnaire (ou vocabulaire). Généralement, le dictionnaire est appris lors d'une phase préalable hors-ligne, à partir d'images représentatives pour la tâche à accomplir. La construction du dictionnaire consiste en une clusterisation (selon la méthode des k-means par exemple) des descripteurs visuels associés aux primitives extraites dans des images d'entraînement. Cela permet notamment d'améliorer la robustesse au bruit local dans l'image lors de l'appariement ultérieur des primitives. Une fois un dictionnaire appris, il est possible d'inférer la classe d'une image simplement

sur la base de la fréquence des mots qu'elle contient. La structure globale est ainsi ignorée, améliorant alors la robustesse aux occultations partielles. Les étapes de construction hors-ligne et d'utilisation du dictionnaire pour encoder les images sont illustrées sur la figure 2.14.

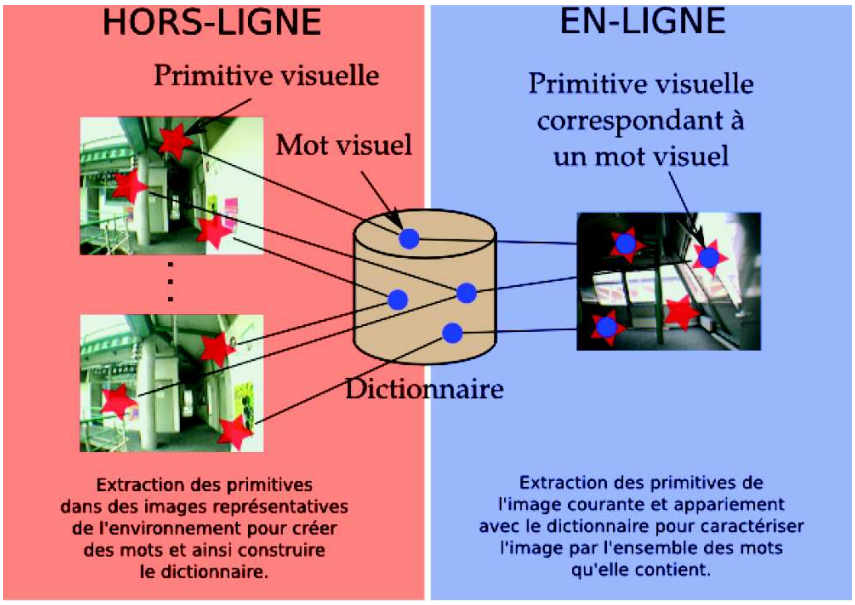


FIG. 2.14 – Construction hors-ligne et utilisation du dictionnaire. Le dictionnaire est construit lors d'une phase préalable hors-ligne par agrégation de primitives visuelles extraites dans des images d'entraînement. Une fois la construction achevée, chaque image traitée est caractérisée par l'occurrence des mots trouvés dans cette image. Source [Angeli, 2008].

Dans le cadre des travaux présentés dans cette thèse, une variante incrémentale [Filliat, 2007] de la méthode des sacs de mots visuels est utilisée. Au lieu d'apprendre le vocabulaire au cours d'une phase préalable hors-ligne, cet apprentissage est effectué en-ligne, à partir d'une structure initialement vide qui est graduellement remplie au fil de la découverte de l'environnement. Pour cela, lorsqu'une image est traitée, les primitives visuelles qui n'ont pas d'équivalent dans le dictionnaire sont ajoutées à celui-ci comme nouveaux mots (voir figure 2.15).

Le mécanisme d'ajout d'un mot dans le dictionnaire repose sur le calcul de la distance entre les descripteurs de la primitive visuelle considérée et tous les mots du dictionnaire. Les mots du dictionnaire sont considérés comme des sphères de rayon fixe dans l'espace des descripteurs. Ainsi, si la distance d'une primitive au centre d'une sphère est inférieure au rayon de cette sphère, la primitive est appariée au mot correspondant. Si la primitive ne tombe dans aucun des mots du dictionnaire,

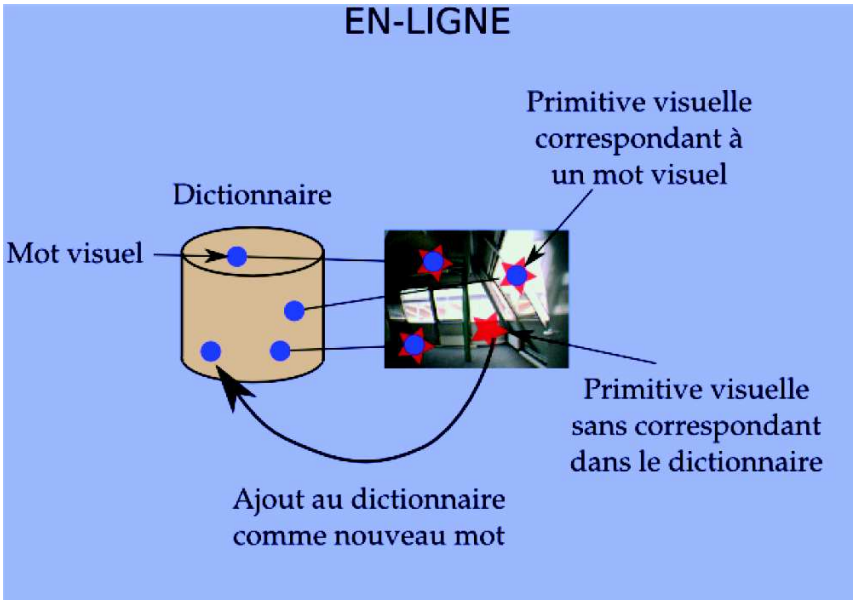


FIG. 2.15 – Construction en-ligne et utilisation du dictionnaire. Le dictionnaire est construit en-ligne au fur et à mesure de la découverte de l'environnement : chaque primitive extraite dans l'image courante qui ne trouve pas d'équivalent dans le dictionnaire (*i.e.* qui ne correspond à aucun mot) est ajoutée à celui-ci comme nouveau mot. Source [Angeli, 2008].

un nouveau mot est ajouté dans le dictionnaire en créant une nouvelle sphère centrée sur la primitive dans l'espace des descripteurs. Grâce à la construction incrémentale du vocabulaire, le système ne fait aucune hypothèse préalable sur le type d'environnement (*i.e.* intérieur ou extérieur) dans lequel le robot va évoluer. Cela offre ainsi une meilleure capacité d'adaptation à tout type d'environnement. Les mécanismes de recherche et d'ajout de mots dans le dictionnaire, ainsi que la structure arborescente du dictionnaire utilisé, sont décrits en détails dans la section 3.3.

Chapitre 3

Détection de fermeture de boucle basée sur la représentation sphérique

Sommaire

3.1 Système d'inférence bayésienne

Notre système de détection visuelle de fermeture de boucle utilise l'approche décrite dans les travaux de [Angeli *et al.*, 2008c], [Angeli *et al.*, 2008b] et plus en détail dans [Angeli, 2008]. Le système repose notamment sur le mécanisme d'estimation incrémentale de la probabilité a posteriori de fermeture de boucle ; mécanisme inscrit dans un processus d'inférence bayésienne. Avant de présenter le système et sa modélisation, quelques rappels théoriques sont fournis. Les concepts de probabilité conditionnelle sont directement présentés sans faire de rappel préalable sur le concept des probabilités et leurs règles fondamentales. Le lecteur n'ayant pas besoin de ces rappels pourra se rendre directement à la section 3.1.2.

3.1.1 Théorie de l'inférence bayésienne

L'inférence bayésienne est un concept très simple reposant sur la règle de Bayes traitant des probabilités conditionnelles :

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{3.1}$$

Dans le cadre de l'inférence bayésienne, nous utiliserons la notation suivante afin de simplifier l'explication et d'expliciter le sens de chaque composante.

$$p(H|E) = \frac{p(E|H)p(H)}{p(E)} \tag{3.2}$$

La notation précédente permet de mettre en avant H en tant qu'hypothèse et E en tant qu'évidence. H est l'ensemble des hypothèses concurrentes qui pourront être modifiées par l'observation de données qui sont alors les évidences E. L'inférence bayésienne permet de déterminer la probabilité a posteriori $p(H|E)$ comme conséquence d'une probabilité a priori $p(H)$ et d'une fonction de vraisemblance $p(E|H)$, notée $\mathcal{L}(H|E)$. Chaque terme de l'équation est expliqué ci-après :

- La probabilité a priori $p(H)$ est la probabilité de H avant que E ne soit observée et correspond donc aux hypothèses du modèle avant observation du comportement du système.
- La probabilité a posteriori $p(H|E)$ est la probabilité de chaque hypothèse après observation de l'évidence.
- La fonction de vraisemblance $\mathcal{L}(H|E) = p(E|H)$ est la probabilité d'observer une évidence E sous une hypothèse H. Elle dénote alors la compatibilité entre une évidence et une hypothèse.

Sa détermination repose sur le modèle de probabilité des données observées.

– $p(E)$ est la vraisemblance marginale ou encore l'évidence du modèle. Elle est la même quelle que soit l'hypothèse H.

En résumé, la probabilité a posteriori d'une hypothèse est déterminée par une vraisemblance intrinsèque, dénotée par la probabilité a priori, et une compatibilité entre l'évidence et l'hypothèse déterminée par la fonction de vraisemblance.

Une définition plus formelle de l'inférence bayésienne est donnée ci-après :

$$p(\theta|X,\alpha) = \frac{p(X|\theta)p(\theta|\alpha)}{p(X|\alpha)} \tag{3.3}$$

X est un vecteur d'observations $x_1, x_2, ..., x_n$. θ représente les paramètres de la distribution des observations tel que $x \sim p(x|\theta)$. α représente les hyperparamètres des paramètres tel que $\theta \sim p(\theta|\alpha)$, θ et α pouvant être des vecteurs. L'avantage de cette formulation est qu'elle permet d'expliciter ce que comprend chacun des termes :

– La probabilité a priori $p(\theta|\alpha)$ est la distribution des paramètres de la distribution des observations avant que celles-ci ne soient faites. C'est-à-dire que la probabilité a priori $p(H)$ susmentionnée représente la probabilité des hypothèses H avec H étant une distribution. Les hypothèses sont caractérisées par les paramètres de distribution θ conditionnés par des hyperparamètres α. Les hyperparamètres α permettent alors de prendre en compte dans l'équation toutes les hypothèses possibles. Chaque hypothèse est caractérisée par un jeu de paramètres θ et les hyperparamètres α permettent de déterminer quelle hypothèse doit être prise en compte, et par conséquent le jeu de paramètres θ.

– La distribution d'échantillonnage ou distribution des échantillons ou encore fonction de vraisemblance $p(X|\theta) = \mathcal{L}(\theta|X)$ est la distribution des observations conditionnellement aux paramètres, à savoir si les échantillons correspondent à la distribution paramétrée par θ. Ceci correspond bien à la notation précédente de la fonction de vraisemblance $p(E|H)$ où l'hypothèse H est bien caractérisée par θ et l'évidence E est bien l'ensemble des observations faites.

– La vraisemblance marginale ou évidence $p(X|\alpha)$ est la distribution marginale des observations par rapport aux hyperparamètres tel que $p(X|\alpha) = \int_\theta p(X|\theta)p(\theta|\alpha)d\theta$. Comme stipulé précédemment, il s'agit de l'évidence du modèle étant donné que cela ne dépend pas de l'hypothèse H.

– La probabilité a posteriori $p(\theta|X,\alpha)$ est la distribution des paramètres θ après avoir pris en compte les observations X. Cela correspond bien à la notation précédente annonçant qu'il s'agissait de la probabilité d'une hypothèse en prenant en compte l'évidence.

L'évidence $p(X|\alpha)$ n'est généralement pas estimée. La probabilité a posteriori est alors considérée comme proportionnelle à la probabilité a priori mise à jour par l'évidence. L'équation 3.3 devient :

$$p(\theta|X,\alpha) \sim p(X|\theta)p(\theta|\alpha) \tag{3.4}$$

Pour conserver l'exactitude de l'équation, la formulation suivante est adoptée :

$$p(\theta|X,\alpha) = \eta p(X|\theta)p(\theta|\alpha) \tag{3.5}$$

où η est un paramètre de normalisation assurant $\sum p(\theta|X,\alpha) = 1$. Afin d'établir un système d'inférence bayésienne, permettant la mise à jour des hypothèses à partir de l'évidence des observations, il est alors nécessaire de déterminer la probabilité a priori $p(\theta|\alpha)$ et la fonction de vraisemblance $\mathcal{L}(\theta|X)$.

3.1.2 Modélisation du système

Les travaux de détection visuelle de fermeture de boucle reposent sur les travaux décrits en détail dans [Angeli, 2008]. Afin d'appliquer la théorie de l'inférence bayésienne à notre système de détection de fermeture de boucle, il est nécessaire de modéliser le système. Il est notamment important de bien définir quels sont les données disponibles et les paramètres, distributions à estimer.

Soit la variable aléatoire S_t représentant les hypothèses de fermeture de boucle à l'instant t. t est l'instant courant et représente l'écoulement du temps en termes d'indices, il prend successivement les valeurs de 0 à t. Les hypothèses de fermeture de boucle sont tous les instants passés, c'est-à-dire toutes les valeurs prises par t jusqu'à l'instant présent. L'hypothèse de non fermeture de boucle est représentée par la valeur -1. Il en résulte $S_t \in [\![-1, t]\!]$. L'ensemble des mots visuels présents uniquement à l'instant t, extrait de l'image courante, est représenté par la variable z_t. La variable z^t représente la séquence des mots visuels extraits à partir de la séquence d'images : $z^t = [z_0, z_1, ..., z_t]$. L'équation de détection de fermeture de boucle s'écrit alors :

$$p(S_t|z^t) = p(S_t|z_t, z^{t-1}) \tag{3.6}$$
$$= \frac{p(z_t|S_t)p(S_t|z^{t-1})}{p(z_t|z^{t-1})} \tag{3.7}$$
$$= \eta p(z_t|S_t)p(S_t|z^{t-1}) \tag{3.8}$$

Afin de vérifier la validité de l'expression, les termes de l'équation sont analysés du point de vue de l'expression théorique. S_t représente l'ensemble des hypothèses H. Les mots visuels z_t représentent les observations à l'instant t ; ils sont l'évidence E. La probabilité a priori $p(S_t|z^{t-1})$ est la probabilité de chacune des hypothèses à l'instant $t-1$, soit la probabilité de chacune des hypothèses du modèle pour l'estimation à l'instant t. La formule théorique de l'inférence bayésienne $p(H|E) = \eta p(E|H)p(H)$ est bien vérifiée.

En considérant l'équation plus formelle, X est l'évidence apportée par l'observation à l'instant t. X correspond alors à l'ensemble des mots visuels z_t de l'image courante. θ correspond aux paramètres de la distribution suivie par la variable aléatoire S_t. Toutefois, S_t suit une distribution évoluant suivant l'apport d'évidence mais ne correspond pas à une distribution dont l'expression est connue, et exprimable à l'aide de paramètres. θ est dans ce cas une paramétrisation inconnue de la distribution de probabilités. Les hyperparamètres α permettent de considérer l'ensemble des paramètres θ possibles pour décrire la distribution des hypothèses S_t. De même, les hyperparamètres sont une paramétrisation inconnue dans ce cas. Bien que les paramètres θ et α ne soient pas modélisables dans ce cas, ils dénotent l'ensemble des hypothèses S_t possibles.

Le principe de mise à jour au travers du mécanisme d'inférence bayésienne peut alors être mis en œuvre. Avant de résoudre cette équation, la signification de chacun des termes est donnée dans le cadre présent de notre système :

- $p(S_t|z^t)$ est la probabilité a posteriori de fermeture de boucle.
- $p(z_t|S_t)$ est la fonction de vraisemblance $\mathcal{L}(S_t|z_t)$ permettant de calculer la compatibilité entre les observations courantes et les hypothèses de fermeture de boucle déterminées à l'instant précédent. Le mécanisme de calcul de la fonction de vraisemblance est un système de vote détaillé dans la section suivante 3.1.3.
- $p(S_t|z^{t-1})$ est la probabilité a priori de fermeture de boucle.

L'équation précédente peut être reformulée afin de présenter des propriétés intéressantes pour le processus de mise à jour. Il sera alors possible d'évaluer la probabilité a posteriori de fermeture de boucle au travers d'un mécanisme incrémental de mise à jour. La probabilité a priori $p(S_t|z^{t-1})$ peut se réécrire de la manière suivante :

$$p(S_t|z^{t-1}) = \sum_{j=-1}^{t-p} p(S_t|S_{t-1} = j)p(S_{t-1} = j|z^{t-1}) \tag{3.9}$$

L'avantage de cette écriture est qu'elle fait apparaître la probabilité a priori de fermeture de boucle comme un calcul à partir d'un modèle d'évolution temporelle $p(S_t|S_{t-1} = j)$ et d'un terme $p(S_{t-1} = j|z^{t-1})$. Ce dernier terme se trouve être exactement la probabilité a posteriori de fermeture de boucle à l'instant $t-1$, soit l'instant précédent. Un nouveau paramètre p est introduit dans l'équation, celui-ci a pour fonction de ne pas prendre en compte les p derniers instants dans le processus de détection de fermeture de boucle. Cela revient à mettre une probabilité nulle pour l'hypothèse de fermeture de boucle dans les p derniers instants. Ce paramètre est important dans la mesure où le robot évoluant dans un environnement continu, les p dernières images seront forcément très similaires à l'image courante. Il en résulte de fortes probabilités de fermeture de boucles aux alentours de ces instants. Le principe de la détection de fermeture de boucle étant de déterminer si un endroit a déjà été visité par le passé, trouver une boucle sur les derniers instants n'est pas cohérent. Les p derniers instants sont donc supprimés du processus d'évaluation. L'équation complète devient alors :

$$p(S_t|z^t) = \eta p(z_t|S_t) \sum_{j=-1}^{t-p} p(S_t|S_{t-1} = j)p(S_{t-1} = j|z^{t-1}) \tag{3.10}$$

où la probabilité a posteriori de fermeture de boucle à l'instant t dépend uniquement de la probabilité a posteriori de fermeture de boucle l'instant précédent $t-1$. Les probabilités de fermeture de boucle antérieures à $t-1$ sont incluses dans la probabilité de fermeture de boucle à l'instant $t-1$. Il s'agit d'un processus markovien où l'instant futur ne dépend que de l'instant présent. Les termes restants sont la fonction de vraisemblance qui ne dépend que de l'instant présent et le modèle d'évolution temporelle qui ne dépend pas de l'instant auquel il est calculé. La mise à jour de la probabilité de fermeture de boucle est alors un processus incrémental.

En ce qui concerne le modèle d'évolution temporelle, il permet de maintenir la cohérence du système lors de l'estimation des nouvelles hypothèses de fermeture de boucle à partir des anciennes hypothèses.

Cela traduit, par exemple, le fait que si une fermeture de boucle est fort probable à l'instant $t-1$, elle l'est aussi à l'instant t. Il en est de même avec une fermeture de boucle peu probable. Outre la cohérence temporelle, elle permet de maintenir une cohérence spatiale de la localisation de la fermeture de boucle. Si une fermeture de boucle est fort probable à un endroit donné à l'instant $t-1$, la probabilité de fermeture de boucle à l'instant t doit forcément se trouver dans un voisinage proche de la fermeture de boucle éventuelle à l'instant précédent. Le modèle d'évolution temporelle se traduit alors par un ensemble de probabilités et de distributions indépendantes de l'instant de calcul. L'ensemble de ces règles est décrit ci-après :

- $p(S_t = -1|S_{t-1} = -1) = 0.9$, la probabilité de non fermeture de boucle à l'instant t est élevée étant donné qu'il n'y avait pas fermeture de boucle à l'instant $t-1$.
- $p(S_t = i|S_{t-1} = -1) = \frac{0.1}{t-p+1}$ avec $i \in [\![0, t-p]\!]$, la probabilité de fermeture de boucle à l'instant t est faible étant donné qu'il n'y avait pas de fermeture de boucle à l'instant $t-1$. Elle est inversement proportionnelle au nombre d'hypothèses de fermeture de boucle considérées. Il s'agit simplement d'une distribution uniforme sur l'ensemble des hypothèses. Plus le nombre d'hypothèses est élevé et plus la probabilité d'obtenir une fermeture de boucle à un instant donné est faible. Il faut noter qu'il s'agit d'une distribution marginale car conditionnée par $S_{t-1} = -1$ avec $\sum_{i=0}^{t-p} p(S_t = i|S_{t-1} - 1) = 0.1$.
- $p(S_t = -1|S_{t-1} = j) = 0.1$ avec $j \in [\![0, t-p]\!]$, la probabilité de non fermeture de boucle à l'instant t est faible sachant qu'il y avait une forte probabilité de fermeture de boucle à l'instant $t-1$.
- $p(S_t = i|S_{t-1} = j)$ avec $(i, j) \in [\![0, t-p]\!]^2$, correspond à une gaussienne centrée sur la différence entre i et j avec un écart-type tel que la probabilité soit différente de 0 pour exactement 4 voisins (*i.e.* $i = j-2, ..., j+2$). Le voisinage est adapté en fonction de la fréquence d'acquisition de la caméra et de la vitesse de déplacement de celle-ci. Il s'agit en fait d'une diffusion de la probabilité a posteriori de fermeture de boucle afin de prendre en compte la similarité entre les images voisines. L'utilisation d'une gaussienne est la méthode la plus simple. Une somme de deux gaussiennes centrées sur les indices voisins est plus adaptée afin de prendre en compte le déplacement d'hypothèse de fermeture de boucle aux proches voisins dans le cas où le processus prend en charge des images relativement différentes les unes des autres. Pour plus de détails, le lecteur pourra se référer à la thèse de A. Angeli [Angeli, 2008].

La figure 3.1 montre une schématisation du modèle d'évolution temporelle sous forme de graphe

d'états.

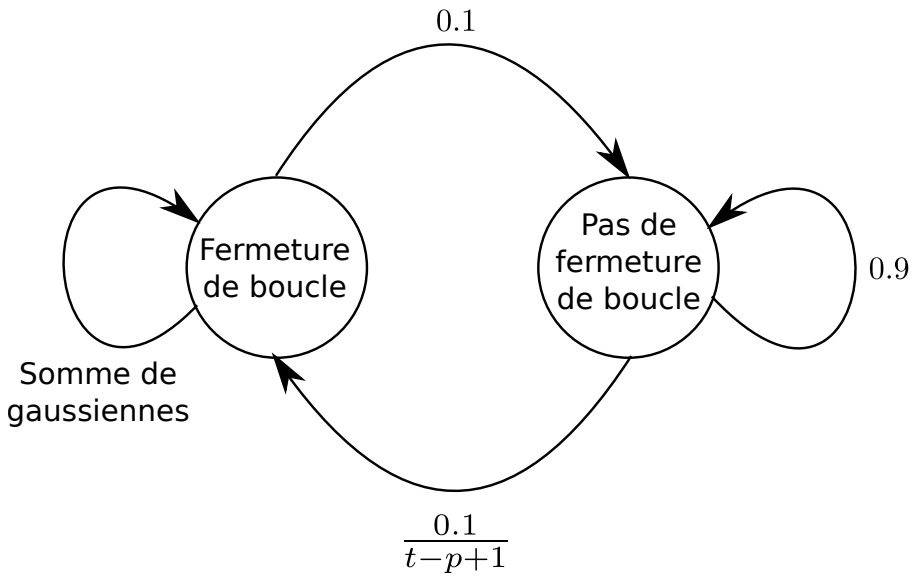


FIG. 3.1 – Modèle d'évolution temporelle représenté sous la forme d'un graphe d'états. Le modèle proposé peut être qualifié de « stationnaire » : la probabilité de rester dans le même état est plus forte que la probabilité de changer d'état.

3.1.3 Estimation de la vraisemblance

L'évaluation de la fonction de vraisemblance, tel qu'annoncé précédemment, repose sur l'utilisation d'un système de vote. Pour rappel, la fonction de vraisemblance $p(z_t|S_t) = \mathcal{L}(S_t|z_t)$ mesure la compatibilité entre les hypothèses de fermeture de boucle S_t et l'évidence obtenue des mots visuels z_t à l'instant t. Pour déterminer la compatibilité, il suffit d'évaluer le nombre de mots visuels communs entre l'image courante et les potentielles images de fermeture de boucle. Il s'agit d'un système de vote permettant d'attribuer un score de similarité entre chaque image déjà observée et l'image courante. Ce système de calcul de similarité correspond pour ainsi dire à une fonction de vraisemblance : une hypothèse de fermeture de boucle avec une image déjà observée est réévaluée en fonction de l'évidence apportée par les mots visuels de l'image courante. La somme de tous les scores ne valant pas 1, il n'est pas complètement possible de considérer le calcul de similarité comme une fonction de vraisemblance. Le calcul ne donne pas lieu à une évaluation de probabilité. Pour pallier ce problème, le paramètre de normalisation η est aussi utilisé pour assurer que le système de calcul de similarité donne une fonction de probabilité (*i.e* la fonction de vraisemblance). Il suffit alors de déterminer sa valeur afin que la somme des probabilités fasse 1.

Avant de préciser comment le calcul du score est effectué, il est nécessaire de préciser l'architecture du système autour du dictionnaire de mots visuels, de l'index inversé et le mode opératoire du mécanisme de vote. Le dictionnaire contient l'ensemble des mots visuels connus. L'index inversé est un répertoire contenant pour chaque mot visuel du dictionnaire l'ensemble des images dans lesquelles il a été observé. Le système fonctionne de la façon suivante :

1. Les mots visuels de l'image courante sont extraits à l'aide du détecteur choisi (détecteur de point SIFT [Lowe, 1999] dans notre cas).
2. Les mots visuels obtenus sont comparés avec les mots visuels contenus dans le dictionnaire.
3. Pour chaque mot visuel de l'image courante trouvé dans le dictionnaire, l'index inversé permet de retrouver les images déjà observées qui contiennent le mot visuel.
4. Pour calculer le score le plus simple qui soit, il suffit de calculer le nombre de mots communs entre l'image courante et l'image concernée, c'est-à-dire faire un cumul du nombre de mots visuels de l'image courante retrouvés dans chacune des images passées.
5. Les scores sont normalisés pour obtenir une fonction de vraisemblance (*i.e.* somme des scores valant 1).

L'image virtuelle d'indice -1, caractérisant la non fermeture de boucle, est créée virtuellement à partir d'un ensemble de mots visuels les plus communs à toutes les images. Un score de similarité avec l'image courante lui est associé pour qu'elle soit prise en compte dans le processus de mise à jour des hypothèses de fermeture de boucle. Cela évite alors de détecter des fermetures de boucle non significatives car basées sur des mots visuels peu discriminants.

Le schéma de la figure 3.2 représente le fonctionnement du système de vote. Ce simple système de calcul de score n'est toutefois pas idéal : la même importance est accordée aux mots apparaissant régulièrement dans les images (outre le principe de l'image virtuelle) et aux mots très peu fréquents (donc très significatifs). Pour améliorer le résultat du score, le système de cumul est remplacé par un système fréquemment utilisé en recherche de documents : le *term frequency - inverse document frequency (i.e. tf-idf)*, introduit dans le domaine de la reconnaissance d'images par [Sivic et Zisserman, 2003]. Ce mécanisme permet d'accorder d'avantage d'importance à un mot récurrent dans une même image qu'à un mot présent peu de fois. Il permet aussi d'accorder plus d'importance à un mot présent

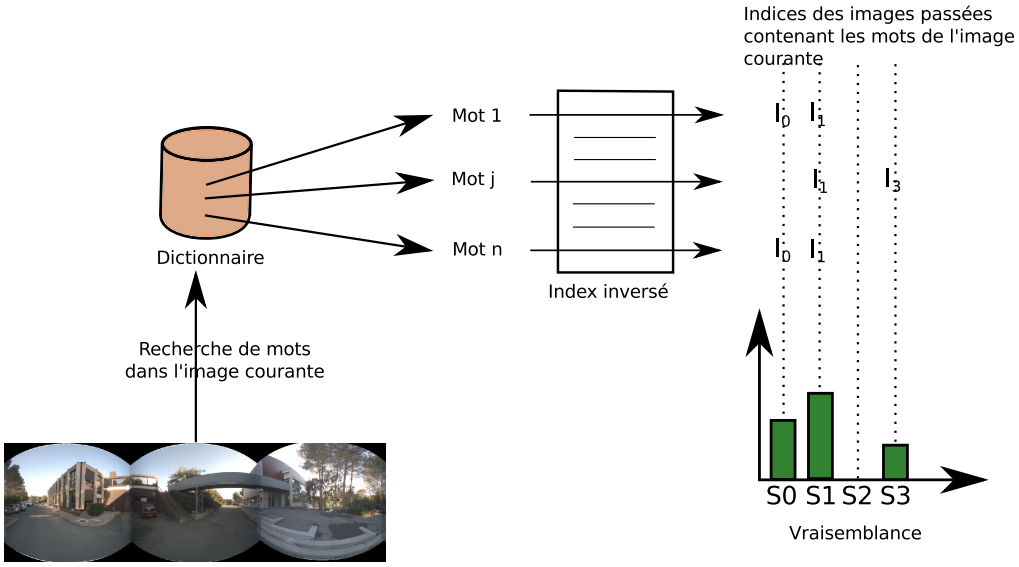


FIG. 3.2 – Schéma du système de vote utilisé avec prise en compte de l'image virtuelle pour la non fermeture de boucle.

dans peu d'images qu'à un mot présent dans beaucoup d'images. Ce calcul permet de prendre l'aspect discriminant d'un mot visuel par rapport à une image et par rapport à l'ensemble des images. Le calcul du score s'exprime donc de la façon suivante :

$$tf\text{-}idf = \frac{n_{wi}}{n_i} log \frac{N}{n_w} \tag{3.11}$$

où n_{wi} est le nombre d'occurrences du mot w dans l'image i, n_i est le nombre total de mots dans l'image i, N est le nombre total de mots dans le dictionnaire et n_w est le nombre d'images contenant le mot w. La formule précédente est valable pour la mise à jour du score pour un mot w appartenant à l'image courante et une image passée i. Afin d'obtenir le score complet pour une image, il suffit de sommer les scores individuels de chacun de mots contenus dans l'image courante. Ce calcul implique une fonction de log-vraisemblance et une indépendance des mots visuels. La fonction de vraisemblance, pour une fermeture de boucle éventuelle avec l'image d'indice i, s'exprime par l'équation suivante (au coefficient de normalisation près) :

$$p\left(z_t | S_t = i\right) \sim \sum_{w \in z_t} \frac{n_{wi}}{n_i} log \frac{N}{n_w} \tag{3.12}$$

La valeur du score, par le biais de la fonction de vraisemblance, permet de mettre à jour les

hypothèses de fermeture de boucle. Il a toutefois été choisi de ne prendre en compte pour la mise à jour des hypothèses seulement les cas où le score est suffisamment significatif. C'est-à-dire qu'il modifie de manière non négligeable l'hypothèse de fermeture de boucle avec l'image i. Les autres scores sont simplement ignorés ; pour cela, il suffit de multiplier la probabilité de fermeture de boucle a priori du score concerné par 1. Ne sont donc considérées pour la mise à jour que les hypothèses dont le score est significativement supérieur à la moyenne des scores. Ceci revient à sélectionner les hypothèses dont le coefficient de variation particulier (*i.e.* l'écart à la moyenne du score normalisé par la moyenne) est supérieur au coefficient de variation standard (*i.e.* l'écart-type normalisé par la moyenne). La valeur utilisée pour la mise à jour de la probabilité de fermeture de boucle associée à l'image i est la différence entre le coefficient de variation particulier correspondant et le coefficient de variation standard plus 1. L'expression de la fonction de vraisemblance finale s'écrit :

$$\mathcal{L}\left(S_t = i | \left(z_k\right)_t\right) = \begin{cases} \dfrac{s_i - \mu}{\mu} - \dfrac{\sigma}{\mu} + 1 = \dfrac{s_i - \sigma}{\mu} \text{ si } s_i \geq \mu + \sigma \\ 1 \text{ sinon} \end{cases} \tag{3.13}$$

Le système de détection visuelle de fermetures de boucle est désormais complètement déterminé. En résumé, nous avons un système incrémental dont la probabilité a priori de fermeture de boucle est la probabilité a posteriori à l'instant précédent. Le modèle d'évolution temporelle permet de maintenir une cohérence temporelle et spatiale des hypothèses de fermetures de boucle. Ensuite, le mécanisme de vote permet l'évaluation de la fonction de vraisemblance utilisée pour la mise à jour des hypothèses à partir de l'évidence de l'observation courante, *i.e.* les mots visuels courants. La normalisation permet d'assurer que les probabilités a posteriori soient cohérentes, c'est-à-dire sommant à 1. Les mécanismes de vote et d'estimation de la probabilité a posteriori de fermeture de boucle sont schématisés dans la figure 3.3.

Bien que ce système soit efficace, il existe encore des fausses alarmes, à savoir des fermetures de boucle qui sont détectées alors qu'elles n'existent pas. Afin de remédier à ces fausses détections, un système de vérification de consistance géométrique a été ajouté. Le mécanisme n'est pas détaillé ici mais il s'agit de déterminer la transformation de corps rigide (*i.e.* rotation et translation) entre l'image courante et l'image supposée de fermeture de boucle. L'algorithme de calcul de la transformation se base sur les mots visuels agrémentés de leurs coordonnées dans chacune des deux images. Si l'algorithme est capable de déterminer une transformation géométrique alors la fermeture de boucle est jugée

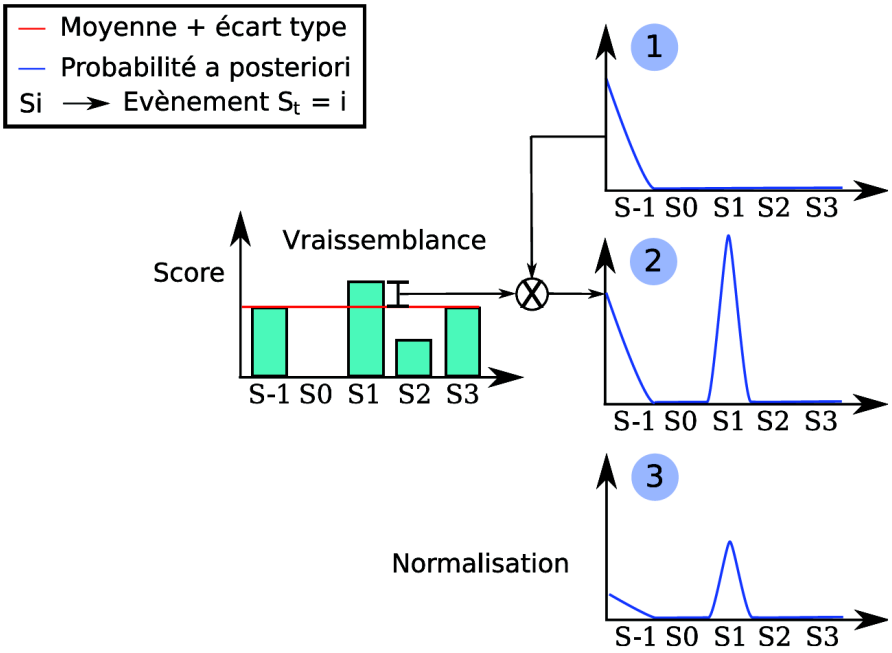


FIG. 3.3 – Schéma du mécanisme de mise à jour des hypothèses de fermeture de boucle à partir du score de similarité obtenu pour chaque image candidate. Le résultat final est la probabilité a posteriori de fermeture de boucle. Source : thèse de A. Angeli [Angeli, 2008]

recevable. Par contre, si aucune transformation n'est calculable, cela signifie alors qu'il y a similitude entre les deux images en terme de mots visuels mais que les lieux ne sont pas semblables en terme de structure. La fermeture de boucle est donc rejetée. Ce système de filtrage est très efficace et permet d'obtenir au final que peu ou pas de fausses fermetures de boucle.
Les mécanismes de ces deux sections sont expliqués plus en détail dans la thèse de A. Angeli [Angeli, 2008].

3.2 Descripteur global sphérique

Le système précédemment décrit dans la section 3.1 fonctionne très bien. Il présente des avantages incontestables en terme de robustesse à l'aliasing perceptuel, de calcul incrémental et d'évaluation de la fermeture de boucle en temps-réel. Toutefois, comme de nombreuses approches tentant de résoudre le problème de la détection de fermeture de boucle, il reste un point problématique : l'indépendance de l'algorithme vis-à-vis du point de vue du robot, ou angle de vue de la caméra. Le système précédent utilisant des images perspectives, il fonctionne lorsque le lieu précédemment visité est revisité dans des conditions d'observation semblables ; c'est-à-dire dans le même sens de parcours ou alors avec un angle

relativement faible par rapport à l'orientation du robot lors de la précédente visite. Les problèmes engendrés sont alors ceux évoqués dans la section 1.3 sur les limitations des méthodes actuelles.
Dans cette section, nous proposons une solution reposant sur l'utilisation du modèle de représentation sphérique de l'environnement introduit dans la section 2.2. L'objectif est d'introduire le modèle de représentation sphérique dans le système de détection de fermeture de boucle tout en exploitant les propriétés particulières offertes par la vue sphérique. La solution est donc une méthode généralisée de détection de fermeture de boucle indépendante à l'orientation du robot. Il ne s'agit pas d'une simple utilisation de caméra panoramique ou sphérique. Un exemple de vue sphérique utilisée est représenté sur la figure 3.4.

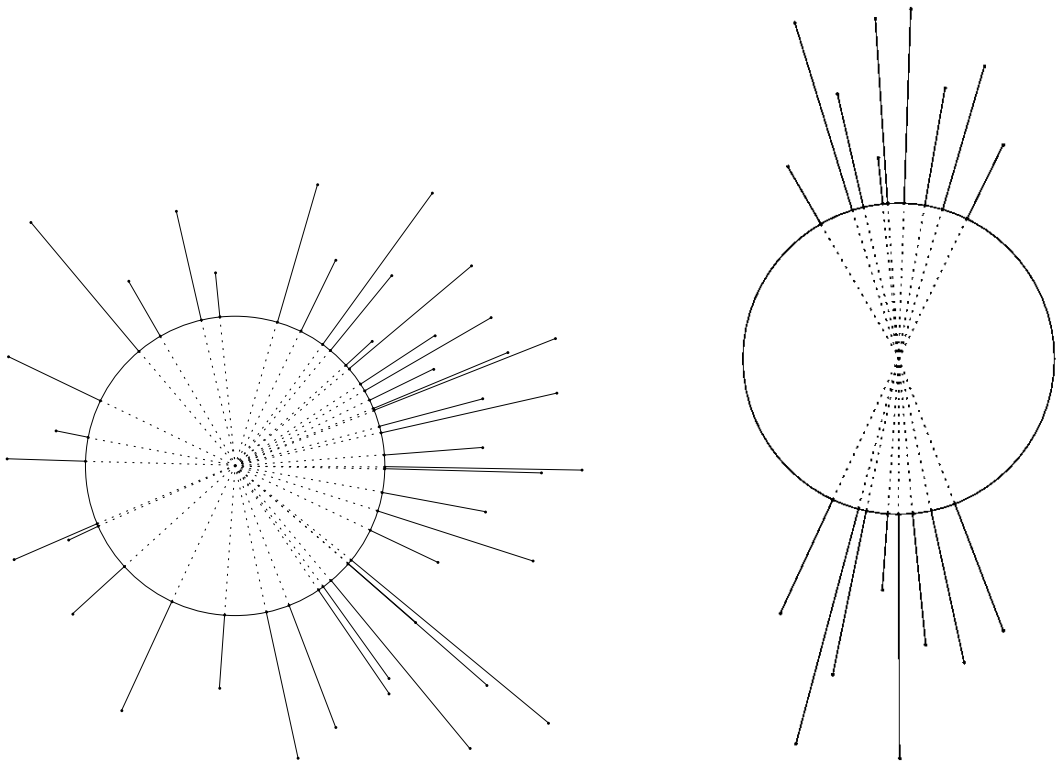
FIG. 3.4 – Exemple de vue sphérique avec projection des points d'intérêt SIFT sur la surface de la sphère. Les points SIFT sont les mots visuels considérés dans cette approche.

3.2.1 Intérêt d'un nouveau descripteur

L'avantage d'utiliser une représentation sphérique de l'environnement est qu'elle fournit une vue omnidirectionnelle. Ainsi, quelle que soit l'orientation du robot, nous obtenons l'information sur tout

l'environnement autour du robot. Sans résoudre le problème de la symétrie des points d'intérêt évoqué dans les limitations de méthodes actuelles (section 1.3), cela permet à chaque nouvelle acquisition d'enregistrer l'information dans toutes les directions. Cela permet notamment d'enregistrer les points d'intérêt qui n'auraient pas été vus au premier passage avec une vue perspective et qui sont découverts lors du passage en sens inverse. La représentation est alors bien plus riche que la représentation perspective.

La représentation sphérique permettant d'obtenir de l'information tout autour du robot, la localisation de celui-ci dans l'environnement est alors plus précise. En effet, les points d'intérêt sont en général robustes aux faibles variations de prise de vue (transformations affines). De ce fait, les points d'intérêt extraits d'une image perspective vont être similaires si le robot effectue une translation suivant l'axe de la prise de vue (problème de parallaxe). La localisation du robot suivant cet axe sera alors imprécise puisque, pour des conditions de prise de vue différentes, la même information est extraite. Dans le cas d'une représentation sphérique, les points d'intérêt extraits suivant l'axe de déplacement du robot (dans ce cas, il s'agit des points d'intérêt extraits suivant la direction de déplacement et dans les deux sens : devant et derrière) engendrent la même imprécision. Par contre, les points d'intérêt extraits suivant les directions perpendiculaires à l'axe de déplacement deviendront rapidement différents lors du mouvement du robot. La prise de vue varie suffisamment rapidement pour sortir du domaine de robustesse des points d'intérêt. Le phénomène est expliqué pour les directions perpendiculaires à l'axe de déplacement mais il est généralisable à toutes les directions d'extraction permettant de sortir rapidement du domaine de robustesse des points d'intérêt. L'ensemble de directions discriminant pour la localisation dépend de la robustesse du détecteur utilisé. Le détecteur ASIFT [Yu et Morel, 2009] serait ici un très mauvais choix du fait de son excellente robustesse aux transformations affines. L'ensemble des points d'intérêt extraits à une position donnée est donc très caractéristique de cette position. Ainsi, il est possible d'obtenir une excellente localisation du robot dans l'environnement en utilisant la représentation sphérique. Cependant, il peut arriver des situations dégénérées où les points d'intérêt sont extraits uniquement dans l'axe de déplacement du robot. Ce cas est alors comparable à la prise de vue perspective : la localisation est imprécise. Ce type de situation peut arriver, par exemple, dans un couloir droit peu texturé. Les points d'intérêt vont alors être localisés à l'entrée et à la sortie du couloir mais pas sur les murs, sols et plafonds. La figure 3.5 représente une schématisation du principe de précision de la localisation du robot.



(a) Sphère bien conditionnée (b) Sphère mal conditionnée

FIG. 3.5 – Répartitions des points d'intérêt autour de la sphère (vue de dessus). Le cas (a) est le cas le plus courant où les points d'intérêt sont répartis tout autour de la sphère. Cette situation permet une localisation précise de la sphère dans l'environnement. Le cas (b) est un cas relativement rare. Il s'agit d'un cas dégénéré où les points d'intérêt sont concentrés dans certains endroits de l'environnement. Dans ce cas, la sphère est mal localisée dans l'environnement. C'est la cas, par exemple, dans un couloir peu texturé.

Bien qu'apportant une nette amélioration conceptuelle, simplement considérer la représentation sphérique dans le processus comme un ajout de nouveaux mots visuels ne permet pas de réaliser une bonne détection de fermeture de boucle. Considérer l'image comme uniquement un ensemble plus important de mots visuels engendre une imprécision supplémentaire. En effet, dans le cas de la vue perspective, les mots visuels sont caractéristiques de l'environnement et de l'orientation du robot. Dans le cas de la vue sphérique, les mots visuels sont uniquement caractéristiques de l'environnement. Ils sont alors moins discriminants dans le processus de détection de fermeture de boucle. De plus, la position du mot visuel sur la sphère n'étant pas prise en compte, une similitude peut exister entre deux mots visuels ayant des descripteurs semblables mais étant localisés à des positions différentes dans l'environnement.

En résumé, la détection de fermeture de boucle est améliorée dans la mesure où celle-ci devient indépendante de l'orientation du robot. La conséquence est une augmentation du nombre de fausses fermetures de boucle. Une solution consiste à conserver le système de vérification de consistance géométrique : les transformations géométriques invalides composant les fausses fermetures de boucles permettent de rejeter l'hypothèse de fermeture de boucle. Cependant, nous bénéficions d'une structure sphérique. Il s'agit d'une structure particulière présentant de nombreux avantages mais elle n'est utilisée que pour l'information supplémentaire qu'elle permet d'obtenir. Notre objectif est alors d'exploiter cette structure plutôt que de faire une vérification de consistance géométrique. Nous créons pour cela un nouveau descripteur global sphérique qui permet d'augmenter notre modèle de représentation en incluant la structure sphérique. Ce descripteur est directement inclus dans le processus d'évaluation des hypothèses de fermeture de boucle. L'étape de vérification géométrique après décision de l'algorithme de détection est alors supprimée. Un avantage supplémentaire de cette approche est que le système final est purement qualitatif. Aucun calcul géométrique n'est effectué évitant ainsi les erreurs d'estimation métrique.

3.2.2 Descripteur global

Notre descripteur sphérique global repose sur deux propriétés importantes :

- L'utilisation de la structure sphérique en elle-même.
- L'introduction d'une notion de localisation des mots visuels sans pour autant impliquer des calculs géométriques. Cette notion doit permettre d'améliorer la pertinence des mots visuels et ainsi augmenter la signification des fermetures de boucle.

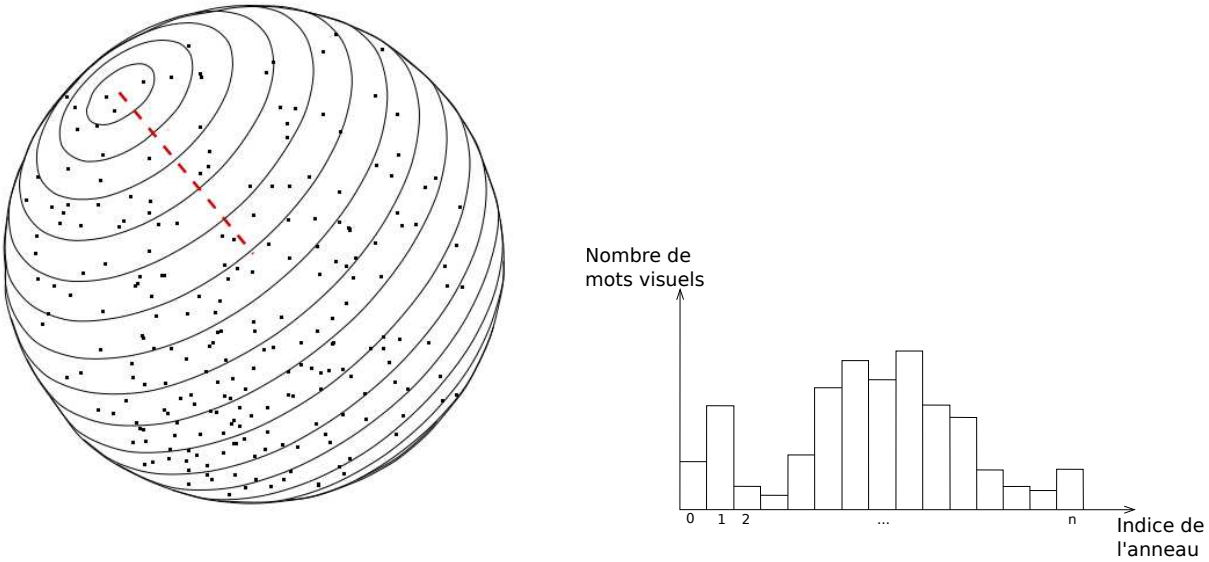
L'ensemble des mots visuels extraits de l'image sphérique se situe sur la surface de la sphère. Sans considérer leurs descripteurs qui permettent de les différencier, ils représentent simplement un nuage de points équidistants du centre de la sphère. Ce nuage de points est alors considéré comme une distribution de probabilité surfacique sphérique des mots visuels. Ce modèle permet de bénéficier des avantages d'une approche probabiliste. Les mots visuels ne sont pas contraints par une position métrique fixe mais par une distribution de probabilité permettant une meilleur tolérance à l'imprécision. D'autre part, la répartition des mots visuels sur la sphère correspond bien à une description partielle de la structure de l'environnement. Cette description n'est pas précise car il s'agit de la distribution de probabilité d'un nuage de points indiscernables (les descripteurs locaux ne sont pas conservés pour le

descripteur global). Pour qu'elle soit précise il aurait fallu qu'elle retranscrive exactement la position de chaque élément structurel de l'environnement. Toutefois, cette information structurelle présente une grande robustesse face aux positions et au nombre de mots visuels. Elle est de plus suffisamment discriminante pour éliminer les fausses fermetures de boucle.

Afin d'établir le descripteur global, il est nécessaire d'estimer la distribution de probabilité du nuage de points. La méthode choisie consiste simplement à discrétiser la sphère en anneaux parallèles à l'équateur. Le nombre de mots visuels contenus dans chaque anneau est alors déterminé. Des méthodes plus complexes d'estimation de densité de probabilité (basées sur des fenêtres de Parzen [Parzen, 1962] ou plus généralement sur des noyaux [Silverman, 1986]) existent mais nous conservons cette méthode simple largement suffisante dans notre approche. Le résultat de l'estimation de densité de probabilité est donc un simple histogramme contenant dans chaque case le nombre de mots visuels de l'anneau correspondant. Pour obtenir une distribution de probabilité, il suffit de normaliser par le nombre total de mots visuels de l'image. L'histogramme résultant est le descripteur global sphérique. Une schématisation du processus de discrétisation de la sphère et de calcul du descripteur est présentée dans la figure 3.6.

Cette méthode présente les avantages d'être très simple et rapide à calculer. Cependant, certaines précautions sont à prendre lors de l'utilisation de ce type de méthode. La première précaution à prendre concerne le pas de quantification, c'est-à-dire le nombre d'anneaux défini sur la sphère. Comme tout système d'estimation par histogramme, si le pas de quantification est trop faible, il devient impossible d'avoir une estimation correcte. Chaque case de l'histogramme contiendrait peu ou pas de mots visuels. La précision n'est pas accrue, l'information est simplement noyée dans du bruit de quantification n'ayant aucune signification. Il s'agit du problème du sur-échantillonnage. Inversement, si le pas de quantification est trop élevé, il est aussi impossible d'estimer de manière correcte une densité de probabilité. L'estimation devient alors très imprécise : le nombre de cases est trop faible pour décrire correctement les variations de la distribution de probabilité. Il s'agit cette fois du problème du sous-échantillonnage. Il est donc nécessaire d'ajuster judicieusement le pas de quantification pour obtenir un bon compromis entre une densité de probabilité bien estimée et un histogramme ne contenant que du bruit.

La deuxième précaution à prendre concerne la méthode de calcul. En effet, la quantification se fait



(a) Représentation du descripteur sphérique global. Les points noirs représentent les positions des mots visuels

(b) L'histogramme constituant la valeur du descripteur.

FIG. 3.6 – Détermination du descripteur sphérique. La sphère est découpée en anneaux concentriques relativement à l'axe joignant le mot visuel concerné et le centre de la sphère. Dans chaque anneau, nous cumulons le nombre de mots visuels présents. Le descripteur final est simplement l'histogramme contenant le nombre de mots visuels dans chaque anneau.

en terme d'angle d'azimut. Chaque anneau possède donc une largeur correspondant à un angle solide d'azimut constant. L'angle d'azimut ϕ étant défini tel que $\phi \in [0, \pi]$, la valeur de l'angle solide définissant la largeur d'un anneau est $\phi_q = \pi/n$ avec n le nombre d'anneaux. La distribution de probabilité décrit donc le nombre de mots visuels en fonction de l'angle d'azimut. Si la surface de chaque anneau avait été considérée comme base de la distribution de probabilité, la surface non constante des anneaux aurait engendré un problème. L'anneau d'équateur est forcément l'anneau avec la plus grande surface tandis que les anneaux des pôles possèdent les plus petites surfaces. Il aurait d'abord fallu normaliser le nombre de mots visuels par rapport à la surface de chaque anneau et ensuite normaliser l'ensemble pour obtenir une distribution de probabilité. Ces deux méthodes de calcul sont équivalentes dans la mesure où elle fournissent toutes deux une distribution de probabilité. Par contre, elles n'ont pas la même signification. La première méthode d'estimation de la distribution de probabilité en fonction de l'angle d'azimut permet d'obtenir une distribution de probabilité sur la surface de la sphère. Cette méthode possède la particularité d'avoir forcément une probabilité plus importante d'apparition de mots visuels à l'équateur qu'aux pôles. La deuxième méthode, en normalisant le nombre de mots

visuels par rapport à la surface de l'anneau, est équivalente à faire l'estimation sur des anneaux distribués sur un cylindre (et donc de surfaces égales). Cette deuxième méthode entraîne un changement de l'objet géométrique sur lequel est estimée la densité de probabilité. De ce fait, la spécificité de la sphère est perdue. Notre méthode utilise donc la première approche afin de conserver au mieux les propriétés géométriques de la sphère.

3.2.3 Invariance à la rotation

La méthode précédemment décrite discrétise la sphère en anneaux suivant l'axe joignant les pôles, ou encore en anneaux parallèles à l'équateur. Le descripteur global sphérique ainsi obtenu décrit effectivement la distribution des mots visuels sur la surface de la sphère. Le système actuel possède un défaut au niveau de la comparaison des descripteurs, comparaison qui permet de déterminer si les distributions sont équivalentes. Le descripteur est invariant seulement à une rotation autour de l'axe des pôles (modification de l'angle de longitude du robot). Si le robot se déplace sur un plan, le seul changement d'orientation possible est dû à une rotation autour de l'axe des pôles. Dans ce cas restreint, le descripteur est bien invariant à l'orientation du robot. La méthode ne conviendrait pas au cas où le robot se déplacerait dans des environnements présentant des dénivelés. La méthode ne conviendrait pas non plus dans le cas d'un drone dont l'inclinaison varie mais pas la position. Le descripteur élaboré n'est pas invariant à une changement d'orientation engendré par une rotation autour d'un axe équatorial (modification de l'angle d'azimut du robot). Ces cas entraîneraient la génération de descripteurs différents pour les angles d'azimut différents, ce qui poserait problème. L'objectif étant d'élaborer une méthode générique applicable aux différents types de robots, il est nécessaire de considérer l'invariance à toutes les orientations possibles.

Déterminer l'orientation relative entre deux sphères pour corriger les descripteurs serait une solution au problème. Cela revient cependant à faire de la vérification de consistance géométrique. Or, ce mécanisme devait être supprimé. Afin d'obtenir l'invariance à l'ensemble des orientations, la solution adoptée est de définir le descripteur global relativement à un mot visuel. Le descripteur global n'est plus défini de manière absolue. La discrétisation en anneaux est alors effectuée suivant l'axe joignant un mot visuel et le centre de la sphère. Étant donné qu'il n'est pas possible de prendre un mot visuel aléatoirement en tant que référence pour l'estimation de la distribution de probabilité des mots visuels, la distribution de probabilité est calculée relativement à chacun des mots visuels contenus dans l'image

sphérique. Il en résulte autant d'histogrammes que de mots visuels contenus dans l'image. La figure 3.7 présente une schématisation du mécanisme de calcul.

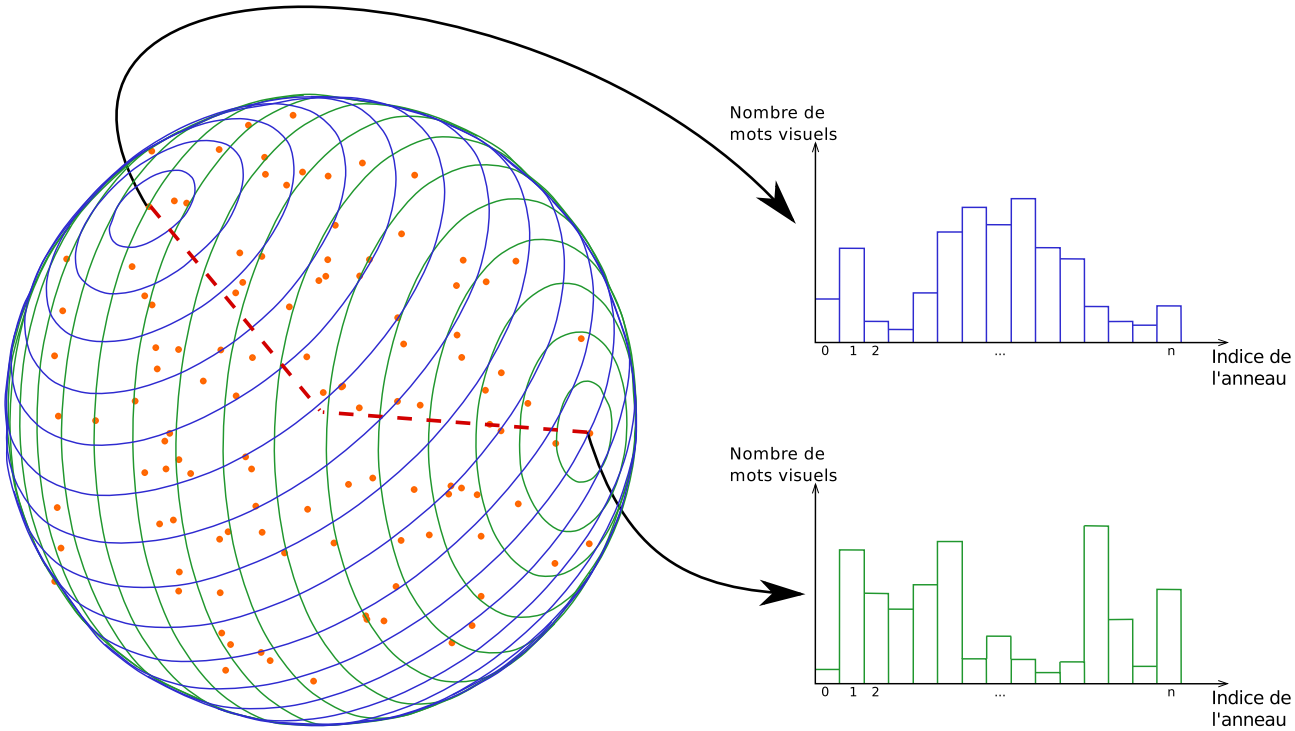


FIG. 3.7 – Schématisation du mécanisme d'estimation de la distribution de probabilité des mots visuels. Ce mécanisme permet d'obtenir une invariance à l'ensemble des orientations. La sphère est discrétisée en anneaux suivant les axes liant chaque mot visuel au centre de la sphère. Les points orange représentent les positions des mots visuels sur la sphère. Deux discrétisations de la sphère sont illustrées sur la figure. Pour chaque discrétisation, une distribution de probabilité associée au mot visuel est estimée. L'histogramme bleu, respectivement vert, correspond à l'estimation de la distribution de probabilité suivant la discrétisation représentée par des anneaux bleus, respectivement verts.

L'inconvénient majeur de cette méthode est que le descripteur global est défini par un ensemble de descripteurs décrivant la distribution de probabilité relativement à chaque mot visuel. Le descripteur global est alors de taille variable. De plus, il est nécessaire d'associer pour chaque estimation de la distribution le mot visuel correspondant afin de ne pas perdre la référence de calcul de la distribution. Toutefois, les distributions étant définies relativement à des mots visuels connus, il suffira de comparer les distributions associées aux mots visuels semblables pour assurer l'invariance à l'orientation. Ainsi, chacun des mots visuels est enrichi de l'information globale de distribution des autres mots visuels dans une image donnée. En considérant l'ensemble des descripteurs constituant le descripteur global,

l'estimation de la distribution du nuage de mots visuels sur la surface de la sphère est encore plus précise. En effet, les mots visuels étant répartis sur toute la surface, la sphère est discrétisée en anneaux suivant différents axes. L'ensemble des estimations de la même distribution suivant différentes discrétisations permet alors de décrire plus finement les particularités. Cette propriété est intéressante car il est possible d'estimer assez finement les variations de la distribution de probabilité sans diminuer le pas de quantification. Le mécanisme d'estimation de la distribution de probabilité permet d'obtenir un système avec un fort pouvoir discriminant.

3.2.4 Modification du système de vote

Afin d'introduire le descripteur global sphérique, il est nécessaire d'opérer un changement architectural de l'algorithme ainsi qu'une modification du calcul dans le système de vote. Sans considérer les modifications apportées par notre approche, une image est caractérisée par les mots visuels qui la constituent. Ces mots sont enregistrés dans un dictionnaire. Associé au dictionnaire, un index inversé permet de savoir dans quelles images chacun des mots visuels a été extrait. Le système de score sert à déterminer le nombre de mots visuels communs entre deux images pour déterminer la similitude. À partir des scores de similarité, la fonction de vraisemblance permet de mettre à jour les hypothèses de fermeture de boucle. La figure 3.2 illustre ce mécanisme.

Les mots visuels n'étant pas dépendants de la structure globale de l'environnement mais simplement d'éléments locaux, il est possible de les enregistrer indépendamment de l'image à partir de laquelle ils ont été extraits. Ils sont alors stockés dans le dictionnaire. Pour ajouter le mécanisme du descripteur global sphérique, il est nécessaire de déterminer où l'information doit être stockée. Étant donné que les descripteurs sont associés à un mot visuel, une solution consisterait à enregistrer l'information dans le dictionnaire. Toutefois, cela n'est pas possible car le descripteur global est dépendant de l'environnement et donc de l'image dont il a été extrait. Si le descripteur global était enregistré dans le dictionnaire, cela reviendrait à enregistrer le mot visuel local enrichi du descripteur global. Tous les descripteurs seraient alors différents et l'avantage de la similitude des descripteurs locaux serait perdu. La solution retenue est donc de ne pas modifier l'information contenue dans le dictionnaire. Par contre, l'index inversé enregistre déjà de l'information relative au lieu d'extraction de l'information. Il enregistre pour chaque mot du dictionnaire les lieux où ceux-ci ont été observés. Le descripteur global est donc ajouté dans l'index inversé. L'index inversé modifié, pour un mot visuel donné, enregistre

l'ensemble des lieux où il a été observé et pour chacun de ces lieux le descripteur global associé est stocké. La figure 3.8 illustre l'architecture modifiée de l'algorithme.

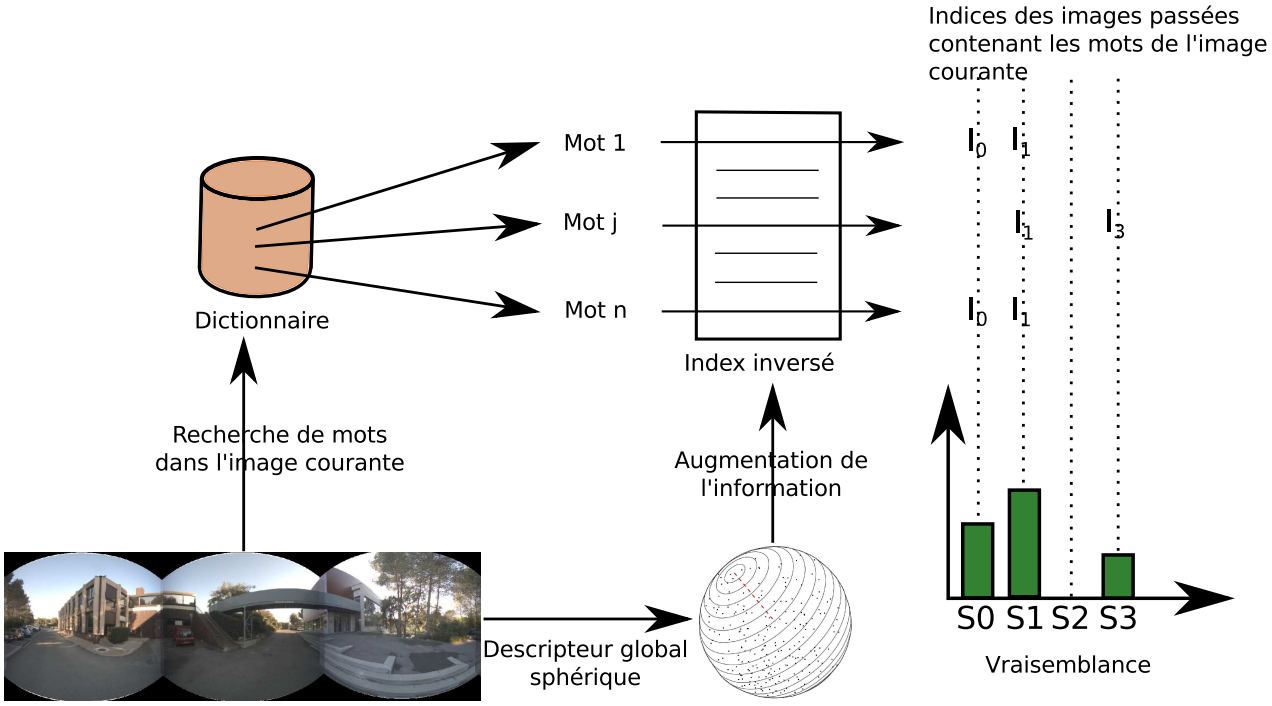


FIG. 3.8 – Modification du système de vote pour prendre en compte l'information du descripteur global sphérique.

La modification du système de vote repose sur une modification du calcul du score. Le score précédent était calculé sur le principe du *tf-idf*. Une modification du système de pondération du score permet de prendre en compte les descripteurs globaux. Étant donné que le descripteur global sphérique enregistre une information de structure de l'environnement, le nouveau système de calcul du score est nommé *structure consistency - inverse document frequency (i.e. sc-idf)*. Le terme *idf* est conservé car, même si les distributions des mots visuels sont différentes, les mots vus moins fréquemment sont plus significatifs que les mots souvent observés. Le terme *sc* remplace le terme *tf*. *sc* est une mesure de similitude de l'environnement (descripteur global) lors de la comparaison des mots visuels. Étant donné qu'il s'agit, à un facteur de normalisation près, d'une distribution de probabilité, il n'est pas possible de simplement calculer la norme L_2 pour déterminer si deux distributions sont identiques. La distance de Tanimoto [Tanimoto, 1957] est une mesure de similarité adaptée à la comparaison entre

deux distributions de probabilité quelconques. Le terme sc est directement le résultat du calcul de la distance de Tanimoto :

$$sc_{wi} = \frac{< hist_{wc}.hist_{wi} >}{||hist_{wc}||^2 + ||hist_{wi}||^2 - < hist_{wc}.hist_{wi} >} \tag{3.14}$$

$< A.B >$ est le produit scalaire entre les vecteurs A et B. $hist_{wc}$ est le descripteur global représentant la distribution des mots visuels dans l'image courante et associé au mot w. $hist_{wi}$ est le descripteur global associé au même mot w mais représentant la distribution des mots visuels dans une image i avec laquelle il y a une hypothétique fermeture de boucle. La distance de Tanimoto est bornée à l'intervalle $[0, 1]$ avec 0 dénotant une dissimilarité complète et 1 dénotant une parfaite correspondance. Le calcul complet du score pour un mot w appartenant à l'image courante et à une image i devient :

$$sc\text{-}idf_{wi} = sc_{wi} \log \frac{N}{n_w} \tag{3.15}$$

Le calcul du score total de similitude entre deux images est alors :

$$sc\text{-}idf_i = \sum_{w \in S_w} sc\text{-}idf_{wi} \tag{3.16}$$

$$= \sum_{w \in S_w} sc_{wi} \log \frac{N}{n_w} \tag{3.17}$$

avec S_w l'ensemble des mots visuels appartenant à la fois à l'image courante et à l'image comparée i.

3.3 Structures de données pour le dictionnaire

Le dictionnaire, contenant l'ensemble des mots visuels, présente un rôle essentiel au sein de l'algorithme. Il constitue la base de connaissance de l'ensemble des lieux visités. Pour chaque nouveau mot visuel extrait de l'image courante, il est nécessaire de chercher dans le dictionnaire quel est le mot le plus proche (s'il existe). La structure du dictionnaire est donc très importante car elle influe considérablement sur le temps de calcul de l'algorithme. Afin d'obtenir un algorithme temps-réel, le dictionnaire doit avoir une structure de données efficace en terme de temps de recherche. L'index inversé est important du fait de son contenu mais sa structure n'est pas critique pour le temps de recherche. La recherche s'effectue dans le dictionnaire et chacun des mots du dictionnaire est directement lié au contenu associé dans l'index inversé. Toute l'information contenue pour une entrée dans

l'index inversé est ensuite utilisée pour le calcul du score, il n'y a donc pas de tri de données ou de sélection particulière à effectuer. Seules les spécificités du dictionnaire en termes de recherche de mots visuels et de mise à jour sont abordées.

La structure de données la plus simple pour enregistrer l'information est de créer un dictionnaire linéaire (*i.e.* un vecteur) contenant tous les mots visuels les uns à la suite des autres. La mise en place de ce type de dictionnaire est très simple. La recherche du mot visuel le plus proche s'effectue simplement par un parcours linéaire du dictionnaire en comparant le mot concerné avec chacun des mots. Bien que simple de représentation, cette approche présente un inconvénient majeur en terme de temps de calcul. Si N est la taille du dictionnaire et W le nombre de mots visuels dans l'image courante, la recherche du plus proche voisin dans le dictionnaire pour l'ensemble des mots de l'image s'effectuera avec une complexité en $\mathcal{O}(NW)$. Même avec W relativement faible, il peut quand même être de l'ordre du millier de mots visuels dans une image. En ce qui concerne la taille du dictionnaire, étant donné qu'il s'agit de l'ensemble des mots visuels connus, elle peut être très élevée : de l'ordre de plusieurs centaines de milliers de mots ($\sim$ 100000 à $\sim$ 300000 mots). Le temps de recherche dans un dictionnaire linéaire est donc rédhibitoire pour des applications temps-réel.

La solution adoptée est d'utiliser une structure de dictionnaire de type KD-tree [Bentley, 1975]. Il s'agit d'une structure en arbre multidimensionnel dont le principe est de répartir les données de part et d'autre d'hyperplans. La dimension du KD-tree est la dimension des vecteurs de données à stocker. À chaque pallier de profondeur k de l'arbre, l'hyperplan est déterminé comme passant par la valeur de la coordonnée k de la donnée stockée et orthogonal à toutes les autres dimensions. Les données suivantes seront stockées de part et d'autre de cet hyperplan d'indice k en fonction de leurs valeurs de coordonnée k. Si la donnée suivante y possède une valeur $y(k)$ supérieure, respectivement inférieure, à la coordonnée de l'hyperplan h_k elle sera alors stockée comme élément « à droite », respectivement « à gauche », de l'hyperplan. En terme de structure de données en arbre, cela signifie que la donnée sera stockée soit dans la feuille droite soit dans la feuille gauche. La nouvelle donnée permet alors de définir l'hyperplan de profondeur $k+1$ centré sur la valeur de la coordonnée $y(k+1)$ de la donnée. L'avantage de cette architecture est qu'elle divise par deux les nombres d'éléments de comparaison lors de la recherche à chaque fois que nous descendons d'un cran en profondeur dans l'arbre. Toutefois, si une donnée se trouve trop près d'un hyperplan, en fonction du seuil de similitude choisi (rayon de la

sphère dans l'espace des descripteurs), il peut être nécessaire de faire du backtracking. Cela consiste en parcourir aussi la branche voisine augmentant le nombre d'éléments de comparaisons sans pour autant avoir à parcourir tous les éléments de l'arbre. Dans le pire des cas, le backtracking entraîne un parcours complet de l'arbre et la recherche devient aussi inefficace que dans le cas du dictionnaire linéaire. Le backtracking est implémenté dans le mécanisme de recherche du mot visuel le plus proche. Il est donc effectué automatiquement si une donnée se trouve trop près d'un hyperplan de séparation. C'est-à-dire si la sphère localisée sur la primitive visuelle intersecte un hyperplan.

En terme de complexité, en moyenne, le temps de recherche d'un mot dans le dictionnaire KD-tree est en $\mathcal{O}(log(N))$. Dans le pire des cas, la recherche est en $\mathcal{O}(N)$. Pour une recherche de W mots visuels dans le dictionnaire, le temps moyen est alors $\mathcal{O}(W.log(N))$. Dans le pire cas, la complexité est en $\mathcal{O}(NW)$. Une technique classique pour maximiser l'apparition du cas moyen est d'équilibrer l'arbre. Cette méthode est possible lorsque les données sont présentées en masse lors de la création de l'arbre. La méthode consiste alors à prendre la donnée médiane suivant la coordonnée correspondant à la profondeur d'insertion et de l'ajouter créant ainsi un hyperplan sur cette donnée. Étant donné qu'il s'agit de la donnée médiane, la moitié des données doit se trouver à droite de l'hyperplan et l'autre moitié à gauche. Cela permet d'éviter d'avoir un arbre dégénéré, c'est-à-dire un arbre ne présentant pas d'équilibre en terme de nombre de feuilles entre ses différentes branches. L'arbre le plus dégénéré qui soit donne naissance à un dictionnaire linéaire. Dans le cadre de notre algorithme, les données sont présentées en ligne, donc une par une ; il n'y a alors pas de système d'équilibrage de l'arbre. Toutefois, nos expériences ont montré que nous sommes majoritairement dans le cas moyen en temps de recherche. En effet, le gain de temps pour la recherche entre le dictionnaire linéaire et le dictionnaire KD-tree correspond bien au facteur apparaissant dans les complexités. Notre arbre est donc construit de manière équilibrée et peu de backtracking est nécessaire. La figure 3.9 présente le concept d'un KD-tree équilibré et le principe des hyperplans séparateurs. Dans cet exemple, nous avons un arbre à trois dimensions où des vecteurs à trois dimensions (x, y, z) sont stockés.

L'aspect dynamique ou statique du dictionnaire est à prendre en considération. Dans les deux cas, le dictionnaire utilise une structure de type KD-tree. Un dictionnaire statique est un dictionnaire dont la taille ne varie pas au cours de l'expérimentation. Il est prédéterminé et est une connaissance a priori non modifiable. Pour construire un dictionnaire statique il existe deux possibilités. La première est

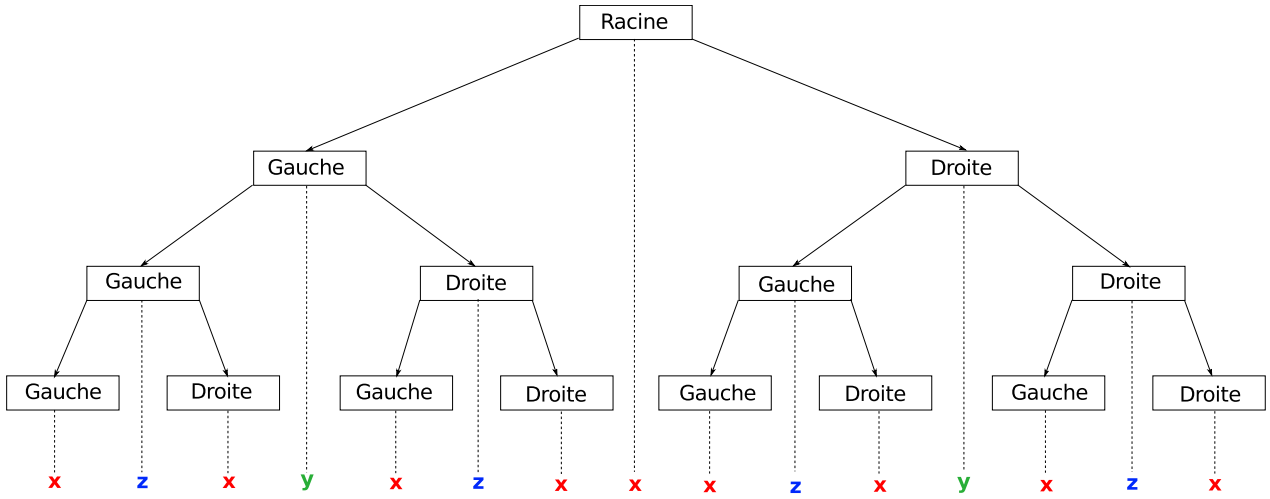


FIG. 3.9 – Exemple de KD-tree à trois dimensions. Les lettres x, y et z représentent l'axe auquel est parallèle l'hyperplan de séparation des données.

de faire une première expérimentation afin de collecter un ensemble de mots visuels qui constitueront le dictionnaire pour les expérimentations futures. L'inconvénient de cette construction est qu'elle ne contient que des mots visuels relatifs à l'expérimentation d'apprentissage. Par conséquent le dictionnaire ne sera pas utilisable pour n'importe quel environnement. L'autre possibilité est d'utiliser des dictionnaires déjà construits et contenant des mots visuels extraits de multiples bases de données afin de couvrir un maximum d'information. Cette méthode permet de couvrir un ensemble plus important d'environnements (ceux dont les mots visuels ont été extraits). L'inconvénient est cette fois d'obtenir une base de connaissance suffisamment large ; l'opération est fastidieuse. De plus, cela ne garantit pas forcément que le dictionnaire est utilisable dans toutes les situations. Par contre, l'avantage des dictionnaires statiques est qu'ils sont prédéterminés, éventuellement épurés de l'information peu significative et optimisés (équilibrage) pour une recherche efficace. La taille fixe permet d'éviter un accroissement démesuré de la taille du dictionnaire lors de l'expérimentation. Le temps de recherche de mots visuels connus est forcément limité.

Un dictionnaire dynamique possède un contenu qui varie au fur et à mesure de l'expérimentation. Il est même possible de débuter l'expérimentation avec un dictionnaire vide, une phase d'apprentissage préalable n'est pas nécessaire. L'avantage principal de ce type d'approche est qu'elle permet de s'adapter au mieux à l'environnement étudié. Elle est par définition utilisable dans n'importe quel type d'environnement. Cette approche est donc beaucoup plus souple vis-à-vis de l'environnement

d'étude. L'inconvénient qui en découle est qu'un dictionnaire construit dynamiquement sera forcément moins optimisé qu'un dictionnaire statique construit et optimisé hors ligne. Il sera alors plus difficile d'avoir un arbre bien équilibré (dégénérescence limitée). D'autre part, il n'est pas forcément possible de déterminer quelle est l'information peu significative. Il en résulte un arbre de plus grande taille. De même, la taille de l'arbre n'est pas contrainte. Il est possible d'avoir un arbre qui grandit démesurément, allongeant ainsi le temps de recherche dans le dictionnaire. En ce qui concerne le coût d'ajout d'un élément, il est le même que celui de recherche (avec backtracking), à savoir $log(N)$ en moyenne. Lors de la recherche d'un mot dans le dictionnaire, soit le mot existe et le dictionnaire n'est pas modifié, soit le mot n'existe pas mais il sera forcément une feuille droite ou gauche du mot le plus proche trouvé. La détermination de l'existence du mot dans le dictionnaire repose sur la comparaison de la distance entre le mot cherché et le mot le plus proche par rapport à un seuil de similarité prédéfini (et déterminé expérimentalement). Le mécanisme d'insertion est basé sur le mécanisme de recherche et engendre un surcout négligeable. La construction en ligne du dictionnaire n'est donc pas pénalisante en terme de temps de calcul pour l'algorithme. Dans notre algorithme, nous utilisons un dictionnaire dynamique pour sa grande adaptabilité à l'environnement. Les expérimentations ont permis de valider une telle approche avec un dictionnaire qui, bien que de grande taille, reste de taille raisonnable et relativement équilibré. Le temps de recherche est significativement réduit par rapport au dictionnaire linéaire.

Chapitre 4

Résultats expérimentaux

Sommaire

4.1 Présentation des expériences

Afin de tester la validité et la robustesse de l'algorithme développé, il a été nécessaire d'effectuer des expérimentations spécifiques. Tout d'abord, il fallait des images sphériques pour pouvoir valider le modèle de représentation sphérique. À défaut d'images sphériques, il était au moins nécessaire d'avoir des images projetables sur une sphère. La condition sine qua non est une image qui approxime suffisamment bien la sphère, contenant donc de l'information visuelle suivant un maximum d'orientations. D'autre part, la trajectoire suivie par le robot devait contenir des fermetures de boucle assez variées afin de tester l'indépendance à l'orientation de l'algorithme. Étant donné qu'il n'existe pas de base de données disponible regroupant toutes ces conditions, nous avons créé une base de données suffisamment riche pour tester l'algorithme dans différentes situations.

L'acquisition de la base de données a été réalisée à l'aide du robot Cycab. Il s'agit d'un véhicule électrique automatisé équipé de différents équipements de mesure et d'acquisition. La figure 4.1 présente le véhicule d'expérimentations.

FIG. 4.1 – Véhicule électrique Cycab.

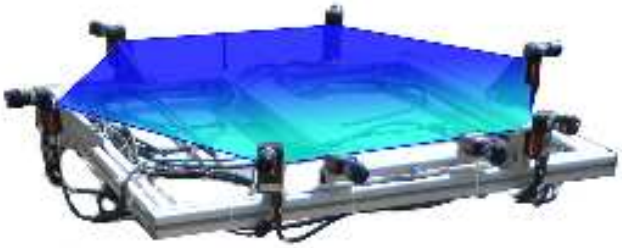


FIG. 4.2 – Anneau de caméras pour l'acquisition d'images sphériques.

Sur le toit du véhicule est monté un système multi-caméras illustré par la figure 4.2. Il s'agit d'un anneau composé de six caméras grand angle dont l'intersection des axes optiques se situe en un point

unique au centre de l'anneau. Ce système permet de générer des images sphériques en accord avec le modèle de représentation sphérique utilisé par l'algorithme. Étant donné la configuration du système de caméras, l'image sphérique résultante ne contient pas d'information aux pôles de la sphère et ne fournit qu'un panorama sphérique incomplet. L'algorithme travaille avec les images projetées sur un plan. La vue sphérique et le panorama résultant de la projection sur un plan de la vue sphérique sont respectivement illustrés par les figures 4.3 et 4.4. Toutefois, cette information est suffisante dans le cadre de cette expérience où le robot se déplace dans le plan, à l'exception de forts dénivelés que nous considérons comme des portions localement planes.

FIG. 4.3 – Vue sphérique.

FIG. 4.4 – Panorama résultant de la projection de la vue sphérique.

Le lieu de l'expérimentation est le site de l'INRIA Sophia Antipolis. Une vue satellite du campus est montrée sur la figure 4.5. L'avantage du site est qu'il offre des environnements très variés qui permettent de tester efficacement la robustesse de l'algorithme. L'INRIA de Sophia Antipolis est constituée de nombreux bâtiments offrant un environnement structuré contenant des mots visuels très caractéristiques facilement identifiables et comparables. Il est assez aisé pour ce type d'algorithme de faire la différence entre les différents bâtiments, en admettant que ceux-ci présentent suffisamment de dissimilitude. Tous ces bâtiments sont construits au milieu de la nature. Il en résulte que la végétation est abondante et apparaît très fréquemment dans la vision du robot. Contrairement aux bâtiments, la végétation est très compliquée à différencier. Elle génère en général de nombreux mots visuels bruités et non significatifs. Quant à ceux qui sont les plus significatifs, ils sont souvent très communs et apparaissent dans de nombreux environnements de végétation relativement différents. Ces mots visuels

sont générés souvent par le feuillage lui-même ou le feuillage contrasté par l'environnement qui se situe derrière. Ces mots visuels sont de manière générale assez néfastes pour ce type d'algorithme. En dernier lieu, le site contient de nombreux parkings possédant un degré de similitude assez élevé. Ceci permet de tester la fiabilité de la détection de fermeture de boucle vis-à-vis de l'aliasing perceptuel. Lors de l'expérimentation, les images ont été acquises sur une trajectoire d'environ 1.5 km. Le nombre d'environnements différents est alors suffisant et la détection de fermetures de boucle est effectuée sur une trajectoire significative.

FIG. 4.5 – Plan du site de l'expérimentation : le campus de l'INRIA Sophia Antipolis.

La trajectoire suivie par le robot est constituée de 1473 images acquises à la fréquence de 1 Hz.

La taille des images panoramiques traitées par l'algorithme (sphères projetées) est de 866x260. La vérité terrain est constituée de 670 fermetures de boucle. Ces fermetures de boucle sont présentes dans différentes conditions d'observation :

- Revisite en sens opposé à la première visite. Ce cas constitue la majorité des fermetures de boucle présentes lors de la trajectoire.
- Revisite avec une prise de vue à $\sim 90°$ par rapport à la première visite. Ce cas se situe à l'entrée du site de l'INRIA, le croisement de routes est assimilable à un carrefour.
- Présence d'objets dynamiques : piétons, voitures.

4.2 Résultats et analyses

La section résultats est organisée en quatre sous-sections. La première contient différents exemples de fermetures de boucle afin de montrer la validité de l'algorithme dans diverses situations, *i.e.* divers environnements et plusieurs orientations. La seconde se focalise sur une analyse des résultats en terme de robustesse de l'algorithme de détection de fermeture de boucle vis-à-vis de l'environnement. Une analyse d'impact des paramètres importants de l'algorithme sur les résultats est aussi effectuée. La troisième se focalise sur des tests des dictionnaires utilisés et les timings de l'algorithme. La dernière met en avant une application de la fermeture de boucle à la correction de dérive d'estimation de trajectoire.

4.2.1 Tests de fermetures de boucle

Les tests de fermetures de boucle sont des exemples d'images correspondant à des fermetures de boucle détectées par l'algorithme. Ils permettent de démontrer la validité de l'algorithme par sa capacité à détecter des fermetures de boucle valides dans les différentes configurations envisagées. Ils permettent aussi de mettre en avant certaines limitations en terme de fausses fermetures de boucle détectées. Ces résultats sont une vue d'ensemble des capacités de l'algorithme de détection mais ne constituent pas une analyse véritable des performances.

Les exemples de la figure 4.6 sont de véritables fermetures de boucle détectées. Les trois premières lignes correspondent à des fermetures de boucle lorsque le robot revient dans un lieu déjà visité en sens inverse. Il existe donc un angle de 180° entre les deux vues sphériques.

- Le premier cas démontre la détection de fermeture de boucle dans un cas relativement complexe

puisqu'il s'agit d'un parking présentant de nombreuses similarités. Ces parkings sont plutôt nombreux sur le site de l'INRIA Sophia Antipolis. L'algorithme développé permet de trouver de manière robuste la bonne fermeture de boucle dans ce type d'environnement.

– Le deuxième cas est une fermeture de boucle dans la situation la plus avantageuse, c'est-à-dire dans un environnement de type urbain. Bien qu'il ne s'agisse pas vraiment d'une ville, le nombre de bâtiments et la structure environnante sont suffisants pour le qualifier d'urbain.
– Le troisième cas montre une détection de fermeture de boucle dans un environnement très riche en végétation. L'algorithme est capable de détecter des fermetures de boucle dans des environnements similaires comme le dénotait déjà la fermeture de boucle dans le parking mais aussi dans des environnements assez peu informatifs contenant beaucoup de végétation peu discernable.

La quatrième ligne est une détection de fermeture de boucle en présence d'objets dynamiques, il s'agit ici d'une voiture qui arrive en sens inverse par rapport au robot. Étant donné l'approche par mots visuels, donc par extraction d'information locale, cette robustesse aux objets dynamiques est en adéquation. Toutefois, elle est à prendre avec précaution puisque la robustesse dépend de la place qu'occupe l'objet dynamique dans l'image et le nombre de mots visuels qui lui sont associés. L'avantage de la vue sphérique est qu'elle limite forcément la quantité d'information associée à l'objet dynamique. La fermeture de boucle est donc possible en utilisant la forte information restante provenant des autres orientations (orientation opposée à l'objet dynamique par exemple). La cinquième ligne, et dernier cas, est une fermeture de boucle dans un cas où le robot revient perpendiculairement à sa première trajectoire. L'angle entre les deux vues est alors d'environ 90°. Ce cas est d'un intérêt tout particulier puisqu'il renforce la validité de l'algorithme quant à l'invariance à l'orientation du robot. Ce cas avait déjà été souligné par les détections de fermetures de boucle en sens opposé mais il s'agit d'un cas supplémentaire dans une configuration particulière.

Les exemples de la figure 4.7 sont des fausses fermetures de boucle détectées par l'algorithme. La première ligne est un cas assez complexe à expliquer puisqu'il n'y a pas de similitude entre les deux images. D'après l'analyse de la séquence d'images, l'erreur proviendrait du fait que peu de mots visuels sont extraits du bâtiment laissant ainsi une prédominance aux mots visuels décrivant la végétation. Le deuxième cas est intéressant car les images, bien qu'appartenant à deux lieux bien distincts, sont très similaires que ce soit au niveau de la végétation ou encore de la présence de bâtiments derrière les arbres pareillement localisés. La différence principale réside dans la présence de voitures dans l'image

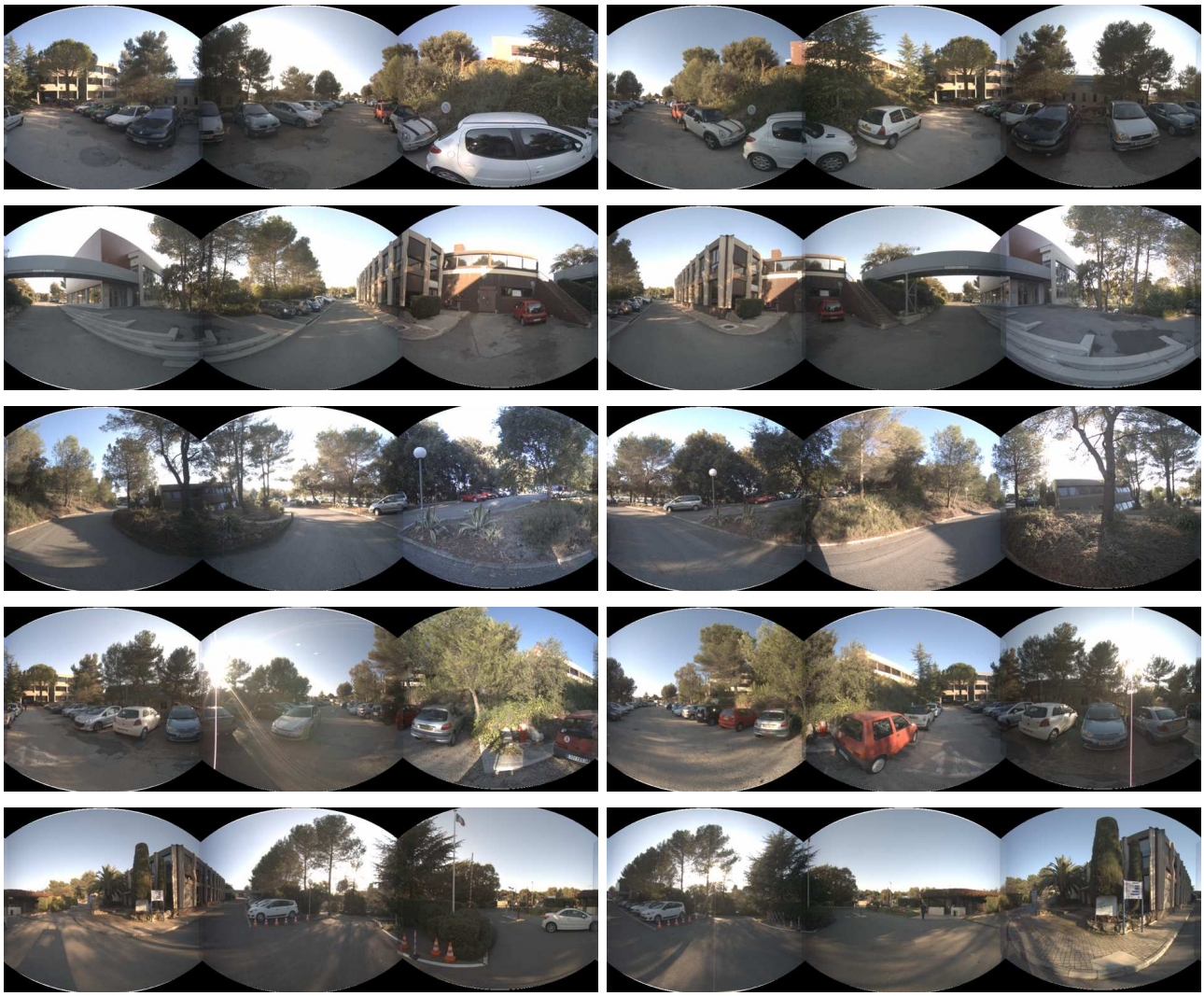


FIG. 4.6 – Exemples de fermetures de boucle obtenues grâce à l'algorithme. L'image de gauche correspond à l'image courante et l'image de droite est l'image avec laquelle l'algorithme ferme la boucle.

de fermeture de boucle.

Avant de passer à une analyse des résultats, cette première approche montrant simplement des images de tests de fermetures de boucle prouve l'avantage indéniable de la représentation sphérique. Cette dernière associée à l'algorithme développé permet de détecter parfaitement les fermetures de boucle dans des situations très différentes et des orientations très diverses. Les cas les plus complexes mettent en exergue la robustesse de l'algorithme. Il reste toutefois quelques fausses fermetures de boucle mais, bien que critiques, elles sont quantité négligeable vis-à-vis de la trajectoire et des types

FIG. 4.7 – Quelques fausses fermetures de boucle détectées par l'algorithme. L'image de gauche est l'image courante tandis que l'image de droite est la fausse fermeture de boucle détectée.

d'environnements étudiés. Sur la trajectoire complète, avec les paramètres optimaux de l'algorithme, seulement onze fausses fermetures de boucle sont détectées. Cependant, une version améliorée ne contenant aucune fausse fermeture de boucle est nécessaire. Pour pouvoir inclure notre algorithme dans un système plus complet de création de carte topologique, il est indispensable de n'avoir aucune fausse fermeture de boucle.

4.2.2 Analyse de robustesse de l'algorithme

En ce qui concerne l'analyse des résultats, nous commençons par faire une comparaison entre une approche standard utilisant une caméra monoculaire produisant une image perspective et notre approche optimisée pour la représentation sphérique. Pour cela, la base de notre algorithme travaillant avec des images perspectives est utilisée, identique à la méthode développée par [Angeli *et al.*, 2008c]. La comparaison est donc effectuée entre cet algorithme et celui développé dans cette thèse. L'image perspective est obtenue à partir d'une des caméras située à l'avant du Cycab. La comparaison des deux algorithmes est ainsi effectuée sur le même jeu de données. Cela évite toute conclusion erronée basée sur une différence qui proviendrait de l'environnement de test et non de l'algorithme. En ce qui concerne l'algorithme standard, aucune vérification de consistance géométrique n'est effectuée. Les fausses fermetures de boucle détectées ne sont alors pas supprimées. Il ne s'agit pas d'un inconvénient majeur car l'objectif de cette comparaison est de montrer la capacité de détection de fermeture de boucle de l'algorithme développé par rapport à l'approche standard. La robustesse à l'aliasing perceptuel n'est pas analysée dans cette comparaison. Les résultats sont présentés sur la figure 4.8. Il s'agit de courbes ROC, *Receiver Operating Characteristic*, qui permettent d'obtenir la sensibilité (*se*)

en fonction de « un moins la spécificité » $(1 - sp)$ ou encore le taux de vrais positifs en fonction du taux de faux positifs. Idéalement, une courbe doit avoir un taux de vrais positifs égal à un, indiquant que toutes les fermetures de boucles sont détectées (pas de faux négatifs), et un taux de faux positifs égal à zéro, indiquant qu'il n'y a pas de fausse fermeture de boucle (pas de faux positifs). Les courbes se situant dans la partie supérieure gauche sont des courbes d'un algorithme efficace. Celles suivant l'axe $f(x) = x$ sont celles d'un algorithme moyen. Quant à celles qui se situent dans la partie inférieure droite, elles correspondent à des algorithmes inadéquats. Les courbes de la figure 4.8 montrent que l'algorithme développé présente de très bonnes performances. L'algorithme standard, quant à lui, présente de mauvaises performances. Sa sensibilité est proche de zéro indiquant qu'il est incapable de détecter les fermetures de boucle. La majorité des fermetures de boucle de la séquence de test étant des fermetures de boucle où l'orientation du deuxième passage du robot est opposée par rapport à l'orientation du premier passage, il est normal que l'algorithme standard ait une très faible sensibilité. Comme expliqué précédemment, et comme le confirme les résultats, un algorithme standard est incapable de détecter ce type de fermetures de boucle. Ces deux courbes montrent donc l'avantage de la représentation sphérique et de l'algorithme développé pour une détection fiable des fermetures de boucle indépendamment de l'orientation du robot.

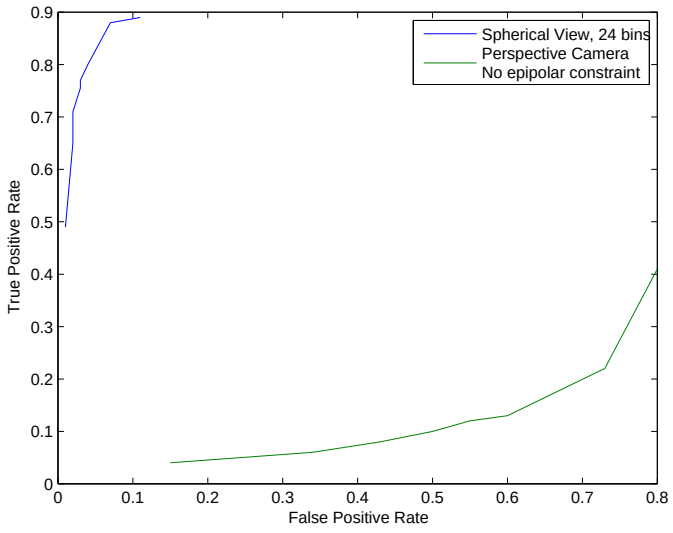


FIG. 4.8 – Graphique de comparaison montrant les courbes ROC d'une approche standard avec caméra monoculaire (Perspective Camera, No Epipolar constraint) et de l'approche utilisant la représentation sphérique (Spherical View, 24 bins).

Il est intéressant d'étudier l'intérêt du descripteur global sphérique afin de déterminer s'il apporte une amélioration des résultats ou si la représentation sphérique à elle seule peut suffire. Pour cela, des courbes ROC sont aussi utilisées pour comparer les deux approches. La première est l'algorithme sans le score modifié, c'est-à-dire qu'il s'agit simplement d'enregistrer tous les mots visuels sur la surface de la sphère et d'utiliser le système de score classique de $tf-idf$. La deuxième est l'approche complète utilisant la discrétisation optimale de la sphère. Ces deux courbes sont affichées sur la figure 4.9. La courbe de l'approche sans le descripteur global sphérique se nomme « SIFT Only (No density descriptor) » et la courbe optimale de l'approche complète est celle nommée « 24 bins ». La lecture des courbes ROC indique que l'approche utilisant le descripteur sphérique global montre de meilleures performances que l'approche utilisant uniquement la représentation sphérique. À taux de vrais positifs égaux, le taux de faux positifs est plus élevé pour l'algorithme n'utilisant que la représentation sphérique. Cela signifie que pour une performance égale en nombre de vraies fermetures de boucle détectées, le nombre de fausses fermetures de boucle est plus élevé. L'algorithme sans descripteur global est capable d'avoir un taux de vrais positifs plus élevé mais au prix d'un taux de faux positifs très élevé. Les faux positifs sont critiques et il est nécessaire de les supprimer, ce qui est fait par la vérification de contrainte épipolaire dans l'approche standard utilisant une vue perspective. L'algorithme complet développé dans cette thèse, en utilisant le descripteur global sphérique, est capable de réduire drastiquement le nombre de faux positifs et de s'absoudre du calcul de contrainte épipolaire. L'algorithme obtenu correspond donc aux objectifs établis : algorithme purement visuel et qualitatif présentant de bonnes propriétés.

L'influence du paramètre de discrétisation de la sphère pour le calcul du descripteur global sphérique est à prendre en compte du fait de son impact important tel que décrit dans la partie théorique 3.2.2. L'impact de la discrétisation est résumé par des courbes ROC sur la figure 4.9. « n bins » indique que la sphère est discrétisée en n anneaux. Pour une discrétisation trop faible ou trop élevée, les performances de l'algorithme sont dégradées. L'optimum de performance est obtenu pour une discrétisation en 24 anneaux. Dans le cas de la trop forte discrétisation, l'algorithme présente des résultats pires que ceux obtenus sans le descripteur global sphérique. Cette dégradation s'explique par la sur-discrétisation qui ajoute beaucoup de bruit dans le système et le rendant incapable de détecter correctement les fermetures de boucle. Elles sont supprimées par non consistance de distribution des mots visuels alors qu'il ne s'agit que de bruit. Les effets de la discrétisation décrits dans la partie théorique sont vérifiés. Il en résulte effectivement un optimum de performances pour une discrétisation particulière. Globalement, et notamment dans le cas optimum, les performances de l'algorithme complet sont largement

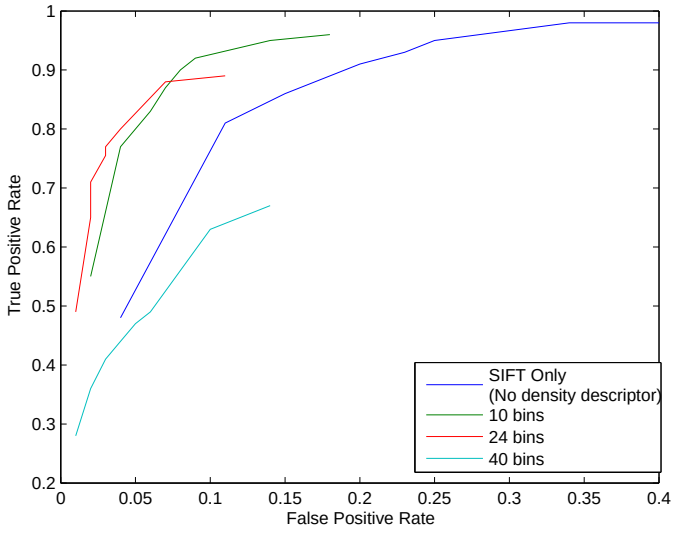


FIG. 4.9 – Graphique de comparaison montrant l'influence du paramètre de discrétisation de la sphère sur les performances de l'algorithme. La courbe SIFT Only correspond à l'utilisation de l'algorithme sans le descripteur global sphérique.

supérieures à l'algorithme utilisant les images perspectives et à l'algorithme utilisant uniquement la représentation sphérique.

4.2.3 Temps de calcul et analyse des performances du dictionnaire

Étant donné que l'approche doit être temps-réel pour pouvoir traiter les données en ligne lors du déplacement du robot, le temps de traitement pour une image est un facteur de performance important. L'algorithme fonctionne à 1 Hz, ce qui est suffisant pour l'approche considérée et par rapport aux expérimentations. Étant donné la faible vitesse de déplacement du Cycab, 1 Hz permet un bon échantillonnage de l'environnement sans pour autant avoir des images trop proches les unes des autres en terme de distance entre les lieux d'acquisition. Toutefois, les fréquences d'acquisition et de traitement de l'algorithme dépendent de la vitesse de déplacement du véhicule. Si le système doit être monté sur un véhicule se déplaçant plus vite (une voiture à 50 km/h en zone urbaine par exemple), il serait nécessaire d'améliorer l'algorithme afin d'obtenir une fréquence de traitement plus élevée. L'algorithme est donc temps-réel dans le cadre des expérimentations effectuées mais il n'est pas exportable à n'importe quelle expérimentation. Du point de vue implémentation, il s'agit d'un code C++ qui a été testé sur un processeur Xeon E5540 avec 16 Gb de mémoire vive. Le programme ne profite pas de l'avantage de l'architecture multiprocesseurs, il est exécuté seulement sur un cœur. Le temps de

calcul moyen pour une image est de 413 ms et le maximum observé est de 656 ms. En ce qui concerne une amélioration éventuelle du temps de calcul, différents éléments sont parallélisables tels que la recherche dans le dictionnaire, le calcul du descripteur global sphérique et le calcul du score. Paralléliser les instructions serait bénéfique dans le cas d'une architecture multiprocesseur.

Comme précisé dans la partie théorique, la structure du dictionnaire possède une influence importante sur le temps de calcul de l'algorithme. Deux paramètres sont alors à considérer :

– La taille du dictionnaire.

– Le temps de recherche moyen d'un élément dans le dictionnaire.

L'évolution de la taille du dictionnaire au cours de l'expérimentation est représentée sur la figure 4.10. Il est intéressant de remarquer que la taille du dictionnaire croît linéairement tout au long de l'expérience. À la fin de l'expérience, il contient presque 350000 mots. Il s'agit d'un dictionnaire de grande taille allongeant le temps de recherche des mots lors de l'évaluation de la fonction de vraisemblance. Malgré la vue sphérique, lors du passage dans les endroits déjà visités, le dictionnaire continue de croître. Cela provient du fait qu'il s'agit d'un dictionnaire dynamique dont les mots peu significatifs ne sont pas exclus. Or, de nombreux mots visuels sont détectés dans le feuillage contrasté par le ciel. Ils sont peu significatifs et peu reproductibles mais sont présents dans le dictionnaire. Toutefois, le dictionnaire contenant 350000 mots permet de décrire un environnement sur une trajectoire de 1.5 km. La description est, de plus, suffisamment fiable pour détecter de manière robuste les fermetures de boucle.

Lié à la taille du dictionnaire, le temps d'accès moyen à un élément permet d'étudier les performances de la structure utilisée pour l'enregistrement du dictionnaire. L'évolution du temps d'accès moyen à un mot visuel dans le dictionnaire au cours de l'expérience, et par conséquent en fonction de la taille du dictionnaire, est représentée sur la figure 4.11. En ordonnée est indiqué le temps moyen de recherche en millisecondes. La courbe suit bien une allure logarithmique comme attendu du fait de l'implémentation de type KD-Tree du dictionnaire. Il en résulte que pour un dictionnaire d'une taille d'environ 350000 mots, le temps d'accès moyen est seulement de 0.16 ms. L'allure logarithmique permet de déduire que l'arbre est équilibré. La courbe $log(N)$ du temps d'accès moyen théorique à un élément de l'arbre est représentée afin de pouvoir comparer la courbe expérimentale à la courbe théorique. La courbe théorique est mise à l'échelle de la courbe expérimentale pour permettre une comparaison cohérente étant donné que la courbe expérimentale informe sur le temps d'accès moyen

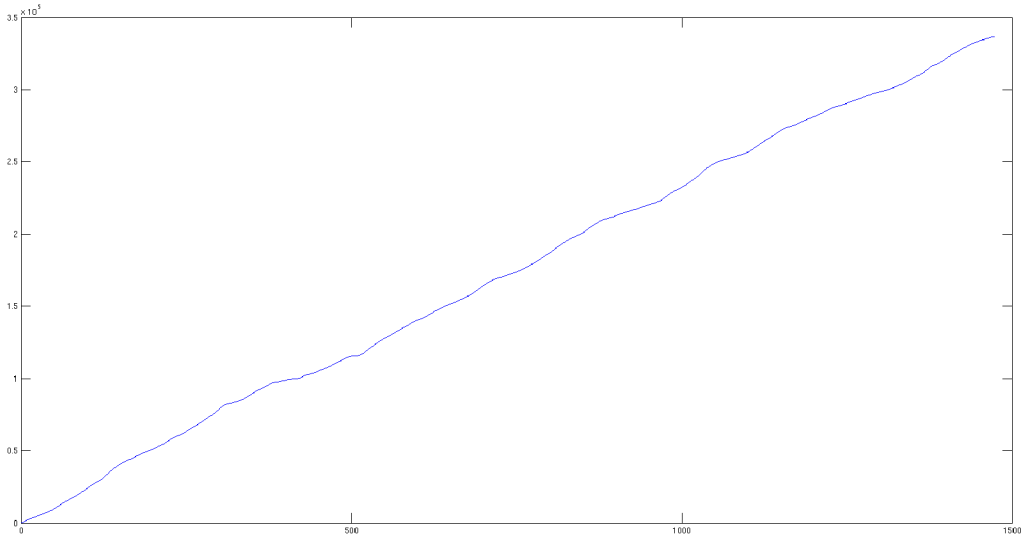


FIG. 4.10 – Évolution de la taille du dictionnaire au cours de l'expérimentation

en millisecondes. L'allure globale de la courbe expérimentale suit une loi logarithmique mais des pics sont répartis de part et d'autre de cette allure. Les pics indiquant un temps d'accès faible par rapport à l'allure signifient que, en moyenne, les mots sont trouvés rapidement dans le dictionnaire. C'est-à-dire qu'il n'y a pas eu backtracking et que les mots ne se situaient pas très profondément dans l'arbre. Les pics indiquant un temps d'accès élevé par rapport à l'allure signifient que, en moyenne, l'algorithme de recherche a eu recours au backtracking à plusieurs reprises. Ces pics étant peu nombreux lors de l'expérimentation, le backtracking est relativement peu utilisé indiquant que l'arbre est construit sur une répartition assez éparse des mots visuels dans l'espace de représentation.

4.2.4 Détection de fermeture de boucle appliquée à la réduction de dérive d'estimation

L'estimation de trajectoire métrique est un problème ardu car l'estimation, basée sur l'intégration des déplacements, contient toujours de la dérive. Les différentes approches accumulent plus ou moins de dérive en fonction de la précision des capteurs utilisés et de la robustesse de l'algorithme. Des méthodes très précises d'odométrie visuelle ont été développées par [Meilland *et al.*, 2011]. La trajectoire est estimée à partir d'information purement visuelle et le résultat contient peu de dérive. La trajectoire estimée par cette méthode sur le jeu de données utilisé dans cette expérimentation est affichée sur la première image de la figure 4.12. Normalement, du fait de la conception de l'expérimentation et donc de la vérité terrain, les points de départ et d'arrivée doivent correspondre. De plus, le robot passe

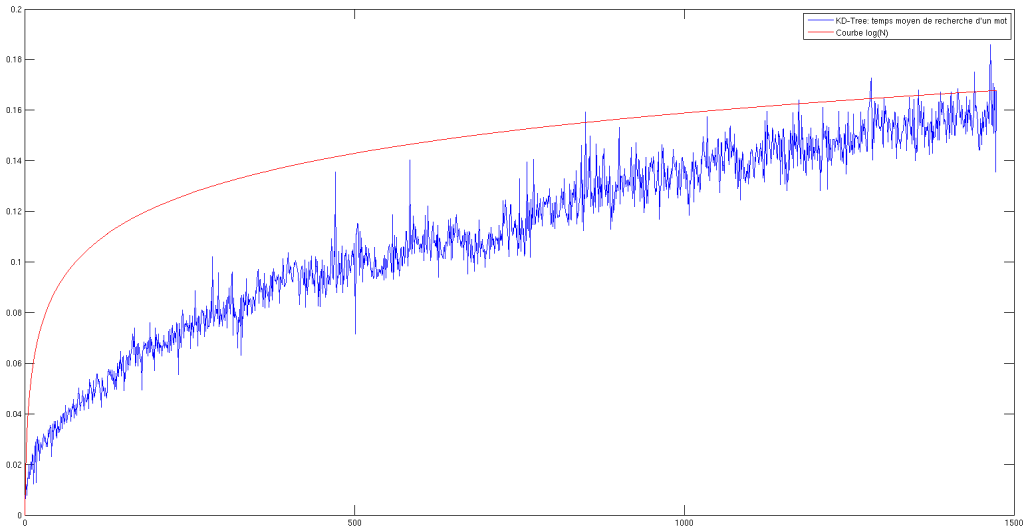


FIG. 4.11 – Temps de recherche moyen d'un mot visuel dans le dictionnaire adoptant une structure en arbre. La courbe du temps de recherche est comparée avec la courbe de complexité $\mathcal{O}(log(N))$, cas moyen du temps d'accès à un élément d'un KD-Tree.

souvent par la même route dans un sens puis dans l'autre. Or, les points de départ et d'arrivée ne coïncident pas et certaines portions de trajectoire qui devraient être confondues sont dissociées. Bien que grandement réduite par l'approche utilisée, la dérive subsiste et notamment à cause de la difficulté de l'environnement. La contrainte de fermeture de boucle a alors été ajoutée à cette estimation de trajectoire. Les fermetures de boucle sont fournies par algorithme développé dans cette thèse. Le système d'ajout de contrainte de fermeture de boucle est le système TORO [Grisetti *et al.*, 2007]. Ce système permet de réduire la dérive en imposant les contraintes de fermetures de boucle à la trajectoire. Les erreurs sont ré-estimées et sont corrigées par propagation le long de la trajectoire. Le graphe de poses est ainsi optimisé. La nouvelle trajectoire est affichée en deuxième image sur la figure 4.12. Sont aussi affichées sur cette image toutes les fermetures de boucle, les onze erreurs exclues. Ces dernières ont été enlevées manuellement avant d'estimer la nouvelle trajectoire à l'aide de TORO. Les points bleus sont les lieux courants lors de l'estimation par l'algorithme de détection de fermeture de boucle. Les points rouges sont les lieux courants lorsqu'il y a fermeture de boucle et les points verts sont les lieux correspondants. À chaque point rouge est associé un point vert, ils forment la paire des lieux fermant la boucle. Un point rouge est forcément un lieu plus récent qu'un point vert. Les traits rouges affichés montrent les différentes paires afin d'expliciter la cohérence des fermetures de boucle : deux lieux constituant une fermeture de boucle doivent être relativement proches. La nouvelle trajectoire correspond mieux à la vérité terrain. Les éléments problématiques notés précédemment sur la première

estimation ont été supprimés.

La dernière image de la figure 4.12 est un zoom sur la zone de la trajectoire où les fermetures de boucles s'effectuent à 90° entre l'image courante et l'image précédemment enregistrée. Même dans une situation perpendiculaire avec des trajectoires un peu écartées, l'algorithme est capable de détecter les fermetures de boucle. Ceci montre sa grande robustesse à l'orientation du robot. Étant donné que l'approche est de type topologique et purement visuelle, les fermetures de boucle malgré l'écart de trajectoire sont cohérentes. Les lieux sont alors considérés comme identiques.

4.3 Discussion

L'algorithme que nous avons développé dans le cadre de cette thèse permet d'apporter une solution efficace au problème de la détection visuelle de fermeture de boucle. Ses atouts majeurs sont :

- Détection visuelle des fermetures de boucle indépendamment de l'orientation du robot.
- Utilisation de la représentation sphérique.
- Exploitation efficace des propriétés de la sphère au travers du descripteur sphérique global.
- Algorithme purement qualitatif : la vérification de contrainte épipolaire est supprimée au profit d'une contrainte de similitude à base de distribution de probabilités. Cette dernière présente une grande robustesse aux erreurs d'estimation.
- Le résultat du processus de décision de fermeture de boucle ne nécessite pas de filtrage supplémentaire, son résultat est fiable.
- Algorithme incrémental et temps-réel (héritage des travaux de [Angeli, 2008]).
- Robustesse à l'aliasing perceptuel (partiellement héritée des travaux de [Angeli, 2008] et renforcée par le descripteur global sphérique).

Malgré ses avantages sur les méthodes existantes, notre algorithme présente l'inconvénient de détecter quelques fausses fermetures de boucle. Bien qu'elles soient très peu nombreuses vis-à-vis du nombre d'images acquises lors de l'expérimentation, elles restent une limitation. Pour pouvoir utiliser l'algorithme dans d'autres processus, il est nécessaire d'augmenter la fiabilité afin de supprimer toutes les fausses fermetures de boucle sans pénaliser la détection des véritables fermetures de boucle. Augmenter simplement le seuil de confiance permet de supprimer toutes les fausses fermetures de

boucle mais beaucoup de vraies fermetures de boucle ne sont alors plus détectées. Une orientation pour fiabiliser l'algorithme serait d'améliorer le système de comparaison des distributions des mots visuels. Bien que donnant des résultats satisfaisants, la distance de Tanimoto n'est pas nécessairement la méthode de comparaison la plus adéquate. D'autre part, il est aussi envisageable de modifier le système de calcul du score afin de le rendre plus robuste et ainsi éviter qu'il augmente la probabilité d'une hypothèse correspondant à une fausse fermeture de boucle. Ajouter le terme tf au score modifié serait une solution.

L'objectif est d'inclure l'algorithme dans un environnement complet de SLAM topologique. Il remplirait les tâches de localisation globale du robot dans le graphe et de construction consistante de la carte topologique. L'avantage d'un algorithme fiable de détection de fermeture de boucle indépendant de l'orientation du robot est qu'il est possible de construire des cartes topologiques fiables indépendamment de l'environnement et de la trajectoire suivie par le robot. Les expérimentations ont été effectuées sur une plateforme robotique mobile de type véhicule. L'algorithme est aussi exploitable pour des applications de cartographie faites à l'aide de drones.
Le temps de calcul reste une limitation d'ordre pratique. L'algorithme est pour le moment restreint à des robots évoluant à faible vitesse. Il sera nécessaire d'améliorer son implémentation pour pouvoir l'exporter sur des plateformes se déplaçant à vitesse plus élevée.

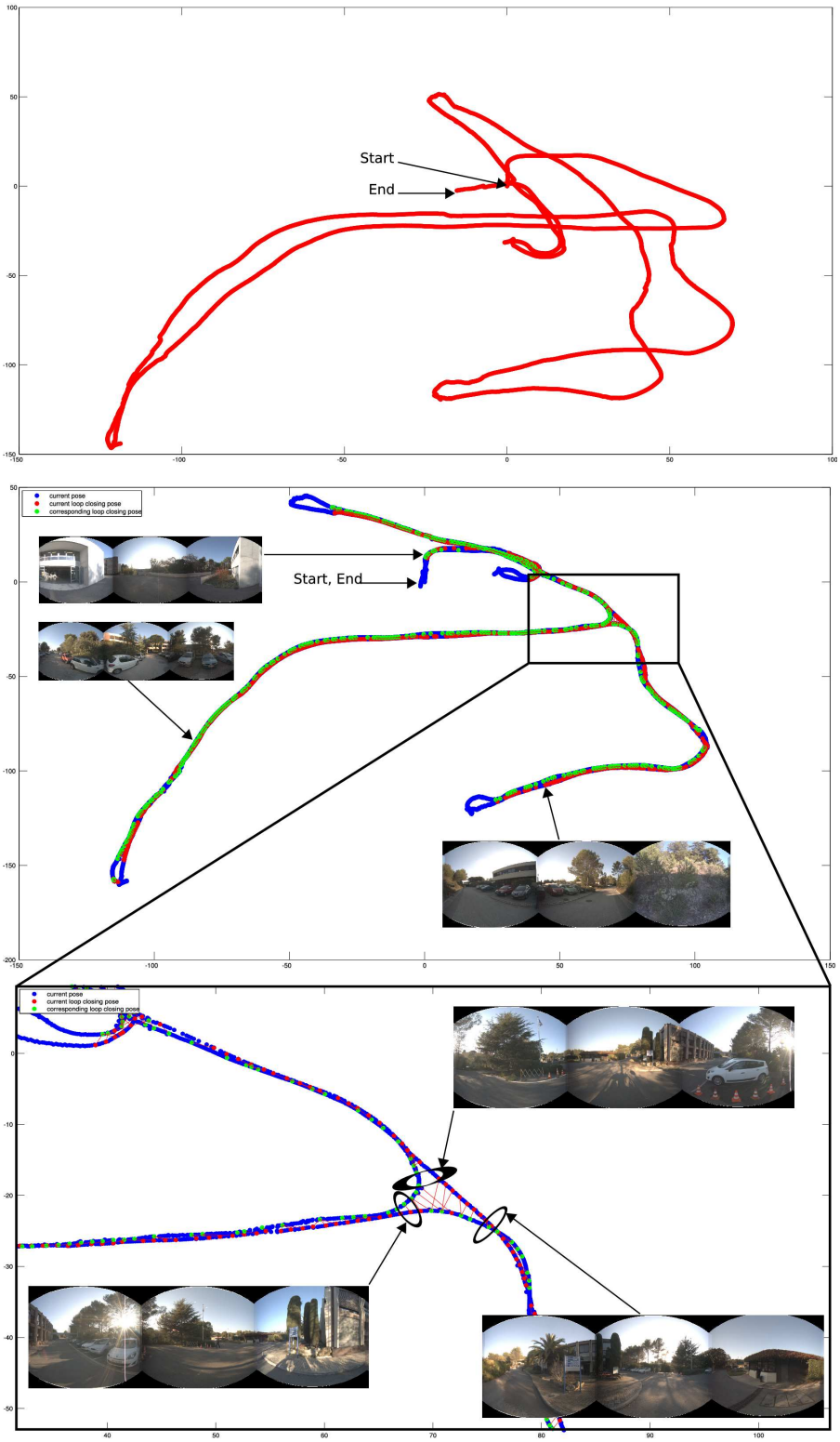


FIG. 4.12 – Application de la contrainte de fermeture de boucle à l'estimation de trajectoire. L'image du haut est l'estimation de la trajectoire sans la contrainte de fermeture de boucle (la vérité terrain donne la même localisation pour le point de départ et celui d'arrivée). L'image centrale est la trajectoire réestimée avec la contrainte de fermeture de boucle. Toutes les fermetures de boucle sont représentées par les points rouges et verts. La dernière image est un zoom sur le lieu où les fermetures de boucle sont détectées dans des situations à 90°.

Deuxième partie

Segmentation de l'environnement pour la cartographie topologique

Chapitre 5

État de l'art

Sommaire

5.1 Introduction

La notion de Localisation et Cartographie Simultanées, *i.e.* SLAM, implique des concepts majeurs qui sont la représentation de la carte et la manière de se localiser dans celle-ci. Depuis le début de l'étude du SLAM il y a une trentaine d'années, plusieurs approches ont été élaborées. Le SLAM métrique fonctionne à partir d'information métrique extraite de l'environnement et du robot. L'environnement est souvent représenté par un ensemble de points 2D ou 3D. Le référentiel de la carte est défini au point de départ du robot. L'information acquise lors du déplacement est alors représentée relativement à ce référentiel. De fait, la localisation du robot est représentée par une pose dans la carte métrique. L'avantage principal de cette représentation est sa précision au niveau local permettant des tâches de navigation précise. En contrepartie, cette représentation est sujette à la dérive de l'estimation (*cf.* l'état de l'art sur la détection de fermeture de boucle, section 1) et les constructions de cartes sur de longues trajectoires perdent en précision. De plus, cette représentation nécessite beaucoup de mémoire pour enregistrer la carte. L'approche devient donc difficile à gérer.

Une façon efficace de pallier ces problèmes est d'utiliser le SLAM topologique. Les cartes sont alors représentées par des graphes dont les nœuds représentent les lieux topologiques et les arêtes représentent l'accessibilité entre les différents lieux. La localisation est alors une tâche plus facile puisqu'il est simplement nécessaire de connaître le nœud courant. Le SLAM topologique présente plusieurs propriétés intéressantes. Tout d'abord, il apporte un bon niveau d'abstraction pour la représentation de l'environnement. Du fait de l'utilisation des graphes, il est aisé de réaliser des tâches de navigation telles que le homing, l'exploration, la planification de trajectoire. Il présente par contre l'inconvénient de ne pas contenir d'information métrique rendant les tâches précises impossibles à réaliser. L'autre inconvénient est d'ordre conceptuel : la représentation utilise des lieux topologiques mais aucune définition précise n'existe. Il n'y a donc pas de méthode de détermination algorithmique bien définie du lieu topologique.

Le SLAM hybride permet de tirer profit des avantages des deux approches en représentant l'environnement via un graphe et en ajoutant l'information métrique dans chacun des nœuds. Ainsi, l'information topologique permet la navigation à grande échelle et l'exécution de tâches de navigation tandis que l'information métrique permet d'effectuer des tâches locales précises.

La construction automatique de représentations topologiques est la problématique abordée dans cette deuxième partie. La détermination des différents lieux topologiques sera obtenue à l'issue d'une segmentation automatique de l'environnement.

5.2 Méthodes de cartographie topologique

5.2.1 Méthodes basées sur les HMM et POMDP

Les méthodes basées sur les HMM, *i.e.* Hidden Markov Model, et POMDP, *i.e.* Partially Observable Markov Decision Process, utilisent une représentation Markovienne de l'environnement. Elles sont historiquement les premières méthodes utilisées pour l'estimation de cartes topologiques. Les méthodes HMM se basent sur une modélisation de l'environnement par des états inconnus du point de vue du robot. Les méthodes POMDP sont quant à elles une extension des méthodes HMM. Elles ajoutent pour chaque état les possibilités d'action du robot en fonction de l'environnement et des capacités du robot. Cette approche définit les concepts d'observation et de décision comme des actions possibles, enrichissant ainsi l'estimation du modèle de Markov. C'est-à-dire que les probabilités de transitions entre les états peuvent être évaluées plus précisément. Que ce soit les méthodes HMM ou POMDP, les deux méthodes utilisent un modèle de Markov avec des états inconnus et des probabilités de transition non définies. L'objectif est alors d'estimer l'ensemble de ces paramètres. Les auteurs dans [Koenig et Simmons, 1996] présentent de bons résultats dans un environnement simple et simulé. L'approche repose sur l'utilisation du modèle POMDP. L'environnement est constitué d'un ensemble de couloirs et d'intersections à angles droits. Le modèle d'action ajouté au processus de décision est la direction que peut prendre le robot à chaque estimation de son environnement proche. Ainsi, dans un couloir, la seule solution est d'avancer ou de reculer. Par contre, à une intersection, il sera aussi possible de tourner à gauche ou à droite (dans le cas d'une intersection de type carrefour). Les résultats sont illustrés par la figure 5.1. Quant aux auteurs dans [Shatkay et Kaelbling, 1997], ils utilisent l'information odométrique comme perception de l'environnement. Dans ce cas, l'expérimentation est menée dans un environnement réel. Malgré les cartes topologiques consistantes produites, ces algorithmes présentent d'importantes limitations dues à un calcul hors-ligne de la carte. De plus, le temps de calcul croît rapidement avec la quantité de données. En effet, l'algorithme de résolution des HMM et POMDP est l'algorithme de Baum-Welch [Baum *et al.*, 1970], algorithme qui devient rapidement inutilisable du fait de la quantité de données nécessaire pour obtenir une bonne estimation des probabilités du modèle.

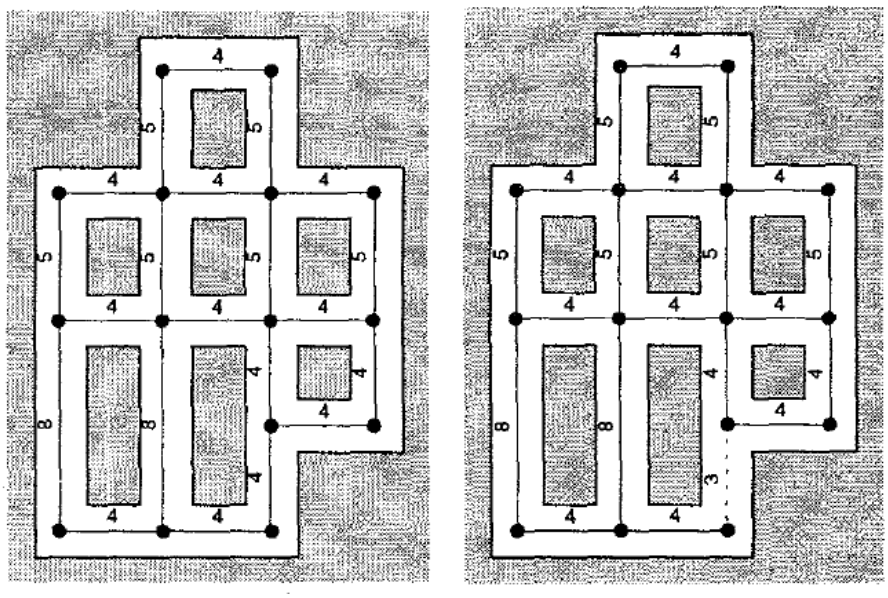


FIG. 5.1 – Résultat de l'estimation de la carte de l'environnement à l'aide d'une approche POMDP. La figure de gauche représente la vérité terrain. La figure de droite représente la carte obtenue à partir de l'algorithme. Source : [Koenig et Simmons, 1996]

Une solution plus intéressante est présentée par les auteurs dans [Tapus et Siegwart, 2005]. Le POMDP est utilisé simplement pour la mise à jour de la carte. La comparaison entre les lieux est faite en utilisant un système d'empreinte efficace. L'empreinte d'un lieu est réalisée en créant un vecteur de caractères représentant l'information extraite de l'environnement à partir de différents capteurs : une caméra omnidirectionnelle et deux lasers possédant un champ de vision de 180°. Le POMDP est alors combiné avec un calcul d'entropie afin de construire la carte topologique. Si l'entropie est faible, ce qui implique une position certaine du robot, l'algorithme met à jour la carte (nœud courant) avec l'empreinte courante. Si elle est trop élevée, le robot continue son exploration jusqu'à ce que l'entropie diminue. Durant la phase d'exploration, des nœuds sont ajoutés à la carte topologique. Cette méthode présente de bons résultats et a l'avantage d'être calculable en ligne. Un exemple de carte topologique extraite de leur article est affiché sur la figure 5.2.

5.2.2 Méthodes des Graph-cuts

Le principe des méthodes Graph-cuts consiste à créer un graphe basique suivant des heuristiques plus ou moins complexes, les plus utilisées étant l'ajout de nœuds à intervalle de temps ou de distance constant. À partir de ce graphe initial, l'algorithme de partitionnement spectral (spectral clustering) est appliqué afin de découper le graphe initial suivant les arêtes de moindre énergie. La différence

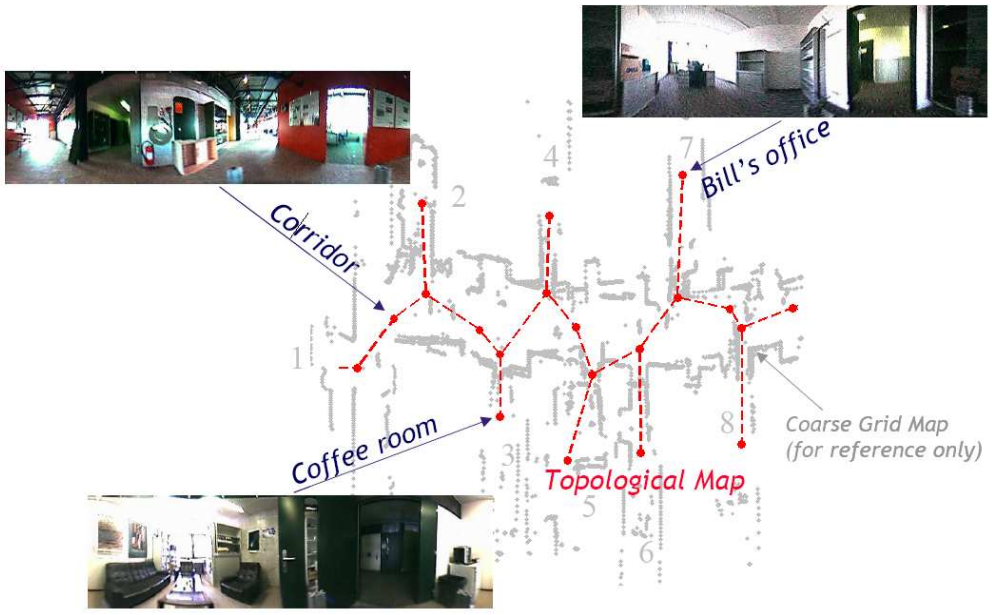


FIG. 5.2 – Exemple de carte topologique obtenue avec l'approche développée par [Tapus et Siegwart, 2005]

majeure entre les différents algorithmes est la méthode de détermination de l'énergie des arêtes. Les nœuds communs, liés par de fortes énergies, sont groupés dans un lieu topologique et liés aux autres lieux topologiques par de nouvelles arêtes. La carte topologique de l'environnement est alors obtenue.

Le concept de la création de carte topologique en utilisant la méthode des graph-cuts a notamment été introduit dans les travaux de [Zivkovic *et al.*, 2007]. Le graphe initial est créé en ajoutant des nœuds à intervalle de temps régulier réalisant un échantillonnage non-uniforme de l'environnement. Le principe repose ensuite sur la découpe du graphe initial en fonction du nombre d'arêtes sur chaque nœud. Étant donné que la découpe du graphe dépend du nombre d'arêtes, la non-uniformité de l'échantillonnage est problématique. Pour pallier ce problème, les auteurs effectuent un ré-échantillonnage uniforme des nœuds. L'algorithme du partitionnement spectral est alors appliqué sur le graphe ré-échantillonné et normalisé. La normalisation sert à éviter d'avoir des groupes ne contenant qu'un seul nœud. Le résultat est alors un graphe de lieux topologiques correspondant principalement à des pièces (*cf.* figure 5.3).

Cette méthode convient bien pour des environnements d'intérieur où les lieux topologiques correspondent à des pièces. Les coupures étant effectuées lorsqu'il y a peu d'arêtes entre les nœuds, elles

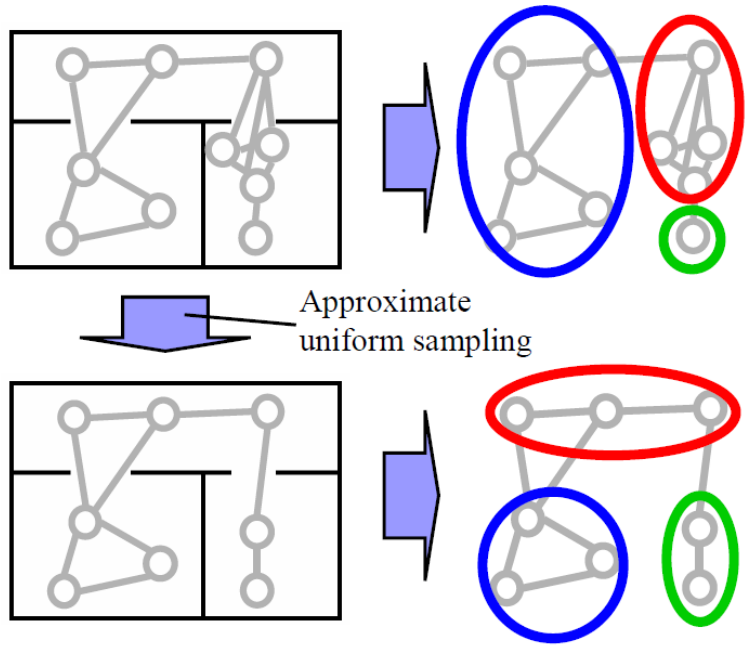


FIG. 5.3 – Méthode de segmentation de l'environnement basée sur la méthode des graph-cuts. L'énergie est ici calculée en fonction du nombre d'arêtes sur chaque nœud. Le ré-échantillonnage uniforme de l'environnement permet d'obtenir la segmentation illustrée dans l'image du bas. Dans l'image du haut, les nœuds ne sont pas ré-échantillonnés. Source : [Zivkovic *et al.*, 2007]

correspondent principalement aux séparations des pièces faites par des portes. De fait, cette méthode ne conviendrait pas pour des environnements d'extérieur où les différents lieux ne sont pas forcément séparés par des chemins étroits. Une limitation de cet algorithme est aussi le fait qu'il se calcule hors-ligne après une phase d'apprentissage de l'environnement.

Les auteurs dans [Blanco *et al.*, 2009] introduisent un nouveau concept de calcul d'énergie pour le partitionnement : le chevauchement d'espace perçu (sensed space overlap). Pendant la navigation du robot dans l'environnement, l'espace est perçu par l'ensemble de ses capteurs et des nœuds sont ajoutés au graphe initial avec un intervalle de temps ou de distance constant. Les capteurs peuvent être de n'importe quel type : caméras, laser. Le chevauchement d'espace perçu correspond aux caractéristiques partagées par plusieurs nœuds. Plus les nœuds partageront de caractéristiques communes et plus l'énergie les liant sera élevée. Le partitionnement est donc fait en regroupant les nœuds liés par de fortes énergies. Ou, vu différemment, les coupures sont effectuées sur les arêtes possédant de faibles énergies et donc entre les nœuds partageant peu de caractéristiques communes. Après chaque

acquisition d'un nouveau nœud dans le graphe, la matrice d'énergie contenant les énergies liant chacun des nœuds du graphe est mise à jour avec le calcul du chevauchement d'espace perçu. L'algorithme de partitionnement spectral est alors appliqué pour déterminer si un partitionnement est nécessaire mettant ainsi à jour la carte topologique. La figure 5.4 montre un exemple de carte construite avec cet algorithme en utilisant un système de stéréo-vision.

Cette approche donne de très bons résultats et présente l'avantage de prendre en compte la visibilité du robot. Le chevauchement d'espace perçu permet d'obtenir des cartes topologiques dont la taille des lieux dépend du champ de vision des capteurs et donc de la capacité du robot à percevoir son environnement. Toutefois, les lieux topologiques résultants ne seront pas forcément très intuitifs et ne correspondront pas nécessairement aux lieux recherchés par un être humain.

Un avantage de cet algorithme est qu'il est capable de gérer des environnements d'intérieur et d'extérieur. Le chevauchement d'espace perçu dépend du champ de vision des capteurs du robot mais aussi du degré d'ouverture de l'environnement. La méthode n'est pas restrictive aux chemins étroits puisque la fonction d'énergie dépend de la similarité entre les différents nœuds. Dans un environnement très ouvert, les lieux topologiques peuvent être très différents car liés par de faibles énergies mais être liés par de nombreuses arêtes. La précédente approche ne permettait pas de gérer ce type d'environnement du fait d'un partitionnement basé sur le nombre d'arêtes entre les nœuds.

L'algorithme est calculable en ligne du fait d'une matrice d'énergie construite séquentiellement pouvant être mise à jour. Le partitionnement peut alors être effectué à chaque incrément. Ainsi, une carte topologique est disponible pour la navigation et la localisation à chaque instant du déplacement du robot.

Les auteurs dans [Martín *et al.*, 2012] utilisent une approche très similaire basée sur le CovGraph. Il s'agit d'un graphe de covisibilité contenant une mesure de similarité entre les différents nœuds. L'algorithme de partitionnement spectral est appliqué sur ce graphe initial en utilisant comme énergie le CovGraph. La méthode donne aussi de bons résultats, similaires à ceux obtenus avec la méthode du chevauchement d'espace perçu.

Dans l'approche développée dans [Zivkovic *et al.*, 2005], l'énergie des arêtes est déterminée par une mesure de similarité entre les points d'intérêt extraits des images (points d'intérêt SIFT [Lowe, 2004]). La mesure de similarité dépend aussi de la contrainte de transformation géométrique entre les images. La transformation doit correspondre à une transformation de corps rigide sinon les images sont considérées comme différentes. Comme les méthodes précédentes, la solution des graph-cuts offre

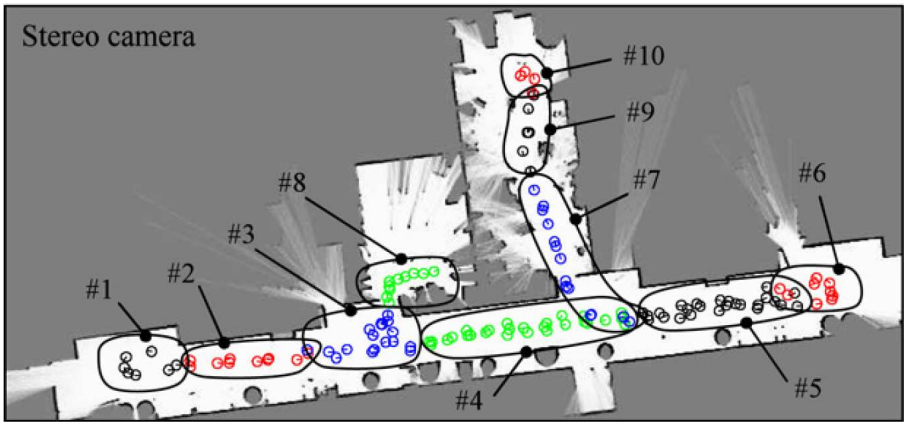


FIG. 5.4 – Exemple de carte topologique obtenue en utilisant les graph-cuts et le concept du chevauchement d'espace perçu. Le capteur utilisé est une paire stéréo-vision. La carte est extraite de l'article [Blanco *et al.*, 2009].

de bons résultats.

5.2.3 Méthodes de distance visuelle

Présentée dans les travaux de [Goedemé *et al.*, 2007], cette méthode de partitionnement de l'environnement en lieux topologiques repose sur la théorie de Dempster-Shafer [Dempster, 1967], [Shafer, 1976]. Contrairement aux approches classiques probabilistes, l'avantage de cette théorie est qu'elle permet de modeler l'ignorance. La possible occurrence d'un évènement est modélisée par une valeur (appartenant à l'intervalle $[0, 1]$) mais l'ignorance à propos de cet évènement est aussi modélisée par une valeur (appartenant aussi à l'intervalle $[0, 1]$). Une ignorance élevée signifie qu'il n'y a aucune connaissance de l'évènement et qu'il est ainsi possible de réfuter une hypothèse d'occurrence. Afin de partitionner l'environnement en lieux topologiques, ils associent à la théorie de Dempster-Shafer la notion de distance visuelle. Cette dernière n'est en fait qu'une simple mesure de similarité entre les différentes images acquises à partir d'une caméra omnidirectionnelle. La mesure de similarité est calculée en utilisant les points d'intérêt extraits de l'image et consiste en un ratio entre les points d'intérêt communs et ceux qui sont différents entre les images comparées. La carte topologique est alors calculée à partir de chaque nouvelle image acquise à intervalle de temps constant. Les nouvelles images acquises sont comparées et éventuellement groupées avec les précédentes. Le groupement se fait en fonction des hypothèses de groupement déterminées à partir de la théorie de Dempster-Shafer appliquée aux mesures de similarité. Le processus étant réalisé en ligne, un détecteur de fermeture de boucle est ajouté, basé sur le même principe de modélisation de l'ignorance, afin de créer des cartes

topologiques cohérentes. Un exemple de carte topologique obtenue est présenté sur la figure 5.5.
Bien que la théorie soit différente, elle n'est pas sans rappeler le principe du chevauchement d'espace perçu présenté dans [Blanco *et al.*, 2009]. L'algorithme présente alors les mêmes avantages de prendre en compte la visibilité du robot et de créer des cartes adaptées aux capteurs. Par rapport aux graph-cuts, il est nécessaire de considérer le processus de détection de fermeture de boucle pour permettre la construction de cartes topologiques cohérentes. Les méthodes utilisant les graph-cuts reposent sur un partitionnement effectué à partir des énergies liant chacun des nœuds, le notion de fermeture de boucle est donc intrinsèque au processus.

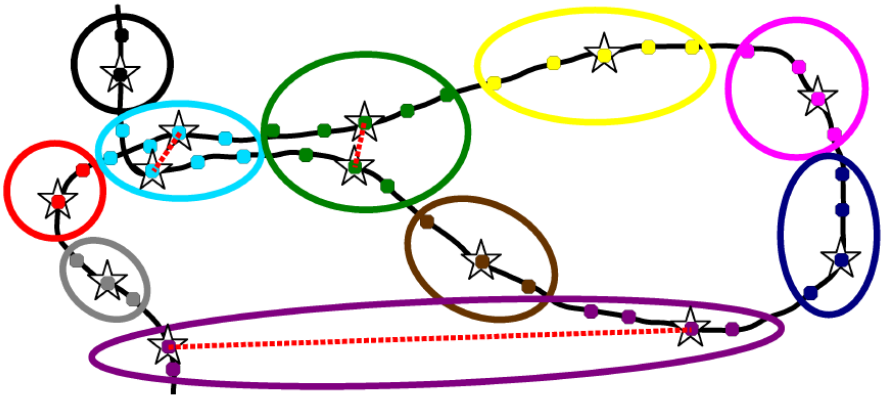


FIG. 5.5 – Exemple de carte topologique obtenue avec la méthode de la distance visuelle combinée à la théorie de Dempster-Shafer. Source : [Goedemé *et al.*, 2007].

5.2.4 Méthodes d'inférence bayésienne

Une méthode originale de création de carte topologique est présentée par les auteurs dans [Ranganathan *et al.*, 2006]. La méthode repose sur un système d'inférence bayésienne dans l'espace des cartes topologiques. Les hypothèses sont alors différentes cartes topologiques établies à partir de l'information extraite de l'environnement et de repères sélectionnés manuellement. La probabilité pour chaque hypothèse est mise à jour à l'aide d'une fonction de vraisemblance établissant la correspondance entre l'observation faite de l'environnement et les différentes hypothèses. Les observations sont la combinaison de l'odométrie bruitée du robot et d'un vecteur contenant une signature de la transformée de Fourier extraite d'une image omnidirectionnelle. Du fait de l'utilisation de la transformée de Fourier sur une image omnidirectionnelle, l'algorithme est indépendant à une rotation du robot dans le plan. Le résultat de l'algorithme est un ensemble de probabilités associées aux différentes hypothèses. L'al-

gorithme garantit d'avoir la bonne carte topologique dans l'ensemble des hypothèses mais ne garantit pas qu'elle obtienne la probabilité la plus élevée. Il s'agit d'une limitation du point de vue création de carte topologique mais l'objectif de l'approche n'est pas d'obtenir la carte exacte mais un ensemble de solutions possibles dans l'espace des cartes topologiques. L'algorithme est testé dans un environnement d'intérieur et l'ensemble des solutions proposées par l'algorithme est cohérent avec la vérité terrain. Toutes les solutions proposées ne sont pas exactes mais elles sont toujours proches de la meilleure solution.

Une autre approche développée par les auteurs [Ranganathan et Dellaert, 2009] utilise le principe de l'inférence bayésienne. Nous la développons dans la partie concernant les méthodes de détection de rupture de modèle. De par sa nature globale, l'algorithme correspond à un système de détection de changement de modèle basé sur un calcul d'inférence bayésienne.

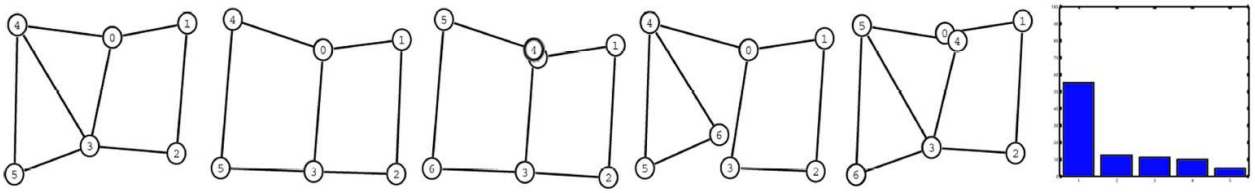


FIG. 5.6 – Ensemble de cartes topologiques proposées de l'environnement associé à la probabilité de vraisemblance de chacune d'entre elles. La méthode repose sur un mécanisme d'inférence bayésienne mettant à jour les hypothèses de chacune des cartes à partir des observations de l'environnement. Dans ce cas, la carte topologique la plus probable est la première carte. Source : [Ranganathan *et al.*, 2006].

5.2.5 Méthodes de détection de rupture de modèle

La détection de changement de modèle, ou détection de rupture, est une méthode très utilisée dans la segmentation vidéo pour la détection d'évènements remarquables. Cette technique repose sur différentes théories comme le CUSUM dans [Tsechpenakis *et al.*, 2006] ou la théorie de l'information de Shannon dans [Itti et Baldi, 2005]. Bien qu'utilisée depuis longtemps dans le domaine de la segmentation vidéo, le méthode n'a été que récemment appliquée au problème du partitionnement de l'environnement en lieux topologiques. Les premiers travaux ont été réalisés par les auteurs dans [Ranganathan et Dellaert, 2009] et [Ranganathan, 2010]. Le principe de la détection de changement de modèle est de détecter des évènements remarquables dans un cadre probabiliste. Conceptuellement, l'environnement est modélisé par des paramètres de probabilité ou des distributions. Il s'agit alors de détecter les variations importantes, appelées évènements remarquables, dans ces paramètres ou distri-

butions. Le problème est alors de modéliser les probabilités de l'environnement. La solution adoptée dans ces articles est de modéliser les paramètres de l'environnement avec un modèle de distribution de Polya multivariée. Le modèle est inséré dans un système d'inférence bayésienne couplé à une analyse statistique de survie pour la détection de changement de modèle.

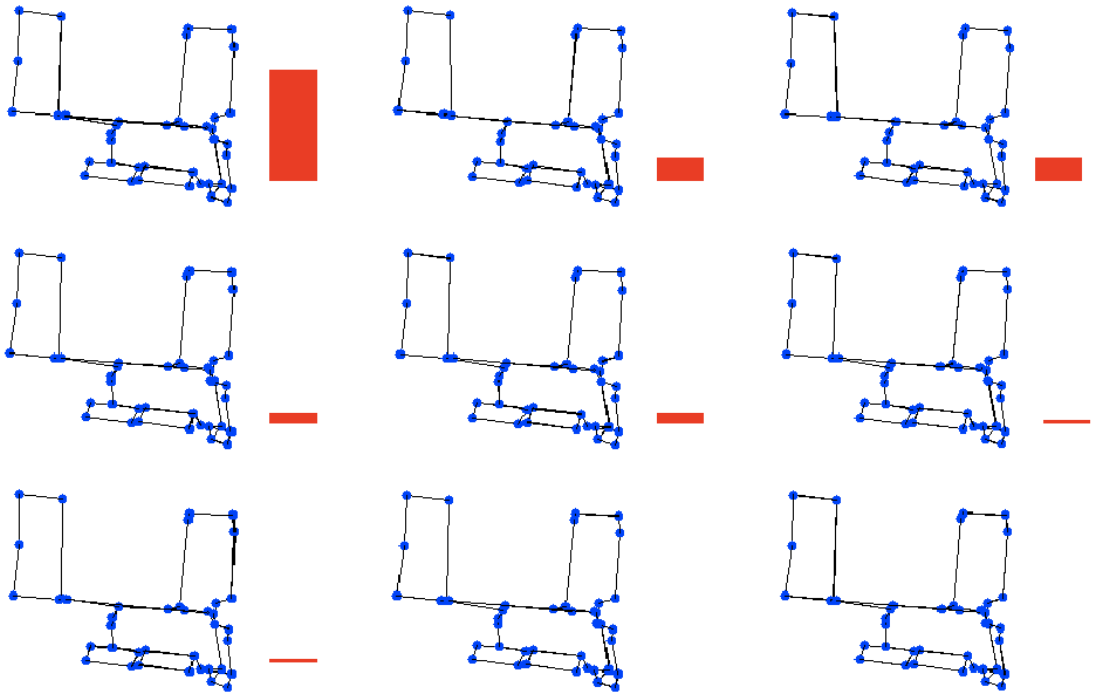


FIG. 5.7 – Ensemble de cartes topologiques proposées par l'algorithme de détection de changement de lieu couplé à une mécanisme d'inférence bayésienne. La carte en haut à gauche correspond à la vérité terrain. L'ensemble des solutions retenues dans l'espace des cartes topologiques correspond assez fidèlement à la vérité terrain. Source : [Ranganathan et Dellaert, 2009].

L'algorithme proposé donne de bons résultats et fonctionne en ligne (*cf.* figure 5.7). Les calculs se font en temps constant en utilisant quelques astuces avec un filtre à particules permettant une réduction de l'information enregistrée. Le calcul est un calcul approximatif car le calcul exact prendrait beaucoup trop de temps. Cette méthode ne requiert pas de phase d'apprentissage mais il est possible d'en ajouter une facilement si nécessaire. Comme la méthode développée par [Blanco *et al.*, 2009] et présentée dans le partie 5.2.2, la détection de changement de lieu dépend du champ de vision des capteurs du fait que les paramètres probabilistes sont extraits de l'information visuelle dans le cas de l'utilisation de cameras. Toutefois, l'objectif de l'approche développée dans [Ranganathan, 2010] n'est

pas la construction de cartes topologiques mais la reconnaissance de lieu, qui peut être considérée comme un système de détection de fermeture de boucle sans étiquetage. L'algorithme peut aussi être utilisé pour la création de cartes topologiques mais il présente en l'état un inconvénient important. Entre les différents lieux, l'algorithme crée une zone dite de transition signifiant que l'algorithme est incapable d'établir la position actuelle du robot. Il est alors incapable d'inférer un modèle probabiliste du lieu. L'algorithme est utilisé pour créer des cartes topologiques dans [Ranganathan et Dellaert, 2009] en apportant les modifications nécessaires pour obtenir une carte topologique cohérente.

5.3 Capteurs utilisés

Au delà de l'algorithme utilisé pour la construction de la carte topologique, la façon dont l'environnement est perçu et représenté est d'importance capitale. De multiples données obtenues de différents capteurs sont utilisées comme l'odométrie dans [Shatkay et Kaelbling, 1997], des coupes laser et la vision dans [Tapus et Siegwart, 2005] et [Ranganathan et Dellaert, 2009] ou encore de la vision pure comme dans [Zivkovic *et al.*, 2005]. La vision présente l'avantage d'être le plus informatif de tous ces capteurs. La cartographie topologique basée sur les images permet un partitionnement en lieux topologiques en prenant en compte une quantité importante d'information extraite de l'environnement. La représentation sphérique telle que présentée dans la partie 2.2 est encore plus riche que la représentation perspective et permet d'obtenir une information indépendamment de l'orientation du robot.

5.4 Définitions des lieux

Cette section est transverse aux précédentes. Elle permet de regrouper les concepts utilisés pour la création des lieux topologiques indépendamment de l'algorithme utilisé. La définition la plus commune trouvée dans l'état de l'art est qu'un lieu topologique correspond à une pièce telle un bureau, une chambre, ...*etc.* En conséquence chaque lieu est séparé par un passage étroit qui est la porte. Cette définition, bien que très intéressante, présente toutefois une limitation. Elle convient bien pour un environnement d'intérieur mais est assez difficile à transposer dans un environnement d'extérieur. Cette approche est adoptée notamment par les auteurs dans [Zivkovic *et al.*, 2007].

D'autres méthodes se basent sur la similitude de caractéristiques extraites de l'environnement en plusieurs endroits. Lorsque les caractéristiques sont suffisamment similaires, elles définissent un lieu

topologique. Les caractéristiques extraites peuvent être diverses et de n'importe quel type (droites dans des coupes laser, points d'intérêt dans une image, ...*etc*). Par contre, ces caractéristiques nécessitent une covisibilité afin de pouvoir établir la mesure de similarité. C'est-à-dire qu'un lieu topologique ne peut être créé qu'à partir de caractéristiques qui se « voient » mutuellement. Le lieu topologique est donc défini sur la base de la covisibilté. Cette approche est abordée de manière différente par les auteurs :

- [Blanco *et al.*, 2009] avec le concept du sensed space overlap
- [Martín *et al.*, 2012] avec le concept du covgraph, très similaire au précédent
- [Zivkovic *et al.*, 2005] avec un mécanisme d'appariement de points SIFT et de vérification de la consistance de la transformation géométrique
- [Goedemé *et al.*, 2007] avec la notion de distance visuelle

Les auteurs [Tapus et Siegwart, 2005] avec un mécanisme de calcul d'entropie ou encore les auteurs [Ranganathan et Dellaert, 2009] avec un mécanisme de détection d'événement surprenant induisent une approche du lieu topologique différente. Le critère de covisibilité n'est plus nécessaire. La création de lieux topologiques s'effectue sur la base de détection de changement de l'information perçue de l'environnement. L'information est constituée par l'empreinte construite dans le premier cas et par la modélisation par une distribution de Polya multivariée de l'environnement dans le second cas.

5.5 Conclusion et motivations

Tous les algorithmes présentés précédemment ont été élaborés pour des environnements d'intérieur. En ce qui concerne ceux qui n'imposent pas cette restriction, ils n'ont toutefois pas été testés en environnement extérieur. D'autre part, un lieu topologique est communément défini comme étant une pièce ou un couloir. Cette définition est la plus intuitive et la plus compréhensible mais elle présente l'inconvénient de ne pas s'appliquer à des environnements d'extérieur. Pourtant, il est possible de définir des lieux topologiques et de créer des cartes topologiques en environnements extérieurs. Une définition plus générique du lieu topologique est donc nécessaire pour pouvoir convenir à la fois pour les environnements d'intérieur et d'extérieur.

Au sein du contexte du SLAM topologique, notre objectif se focalise sur la détermination précise

du lieu topologique. Tout d'abord, nous tenterons de donner une définition du lieu topologique valide pour n'importe quel type d'environnement. Une explication plus approfondie des limitations menant à notre définition du lieu topologique est donnée dans la partie 6.1. Une fois cette nouvelle définition élaborée, le système développé permet d'extraire les paramètres correspondant au lieu topologique à partir de la vision uniquement et en utilisant la représentation sphérique. Les paramètres seront ensuite utilisés dans un algorithme permettant de distinguer les paramètres des différents lieux topologiques rencontrés permettant de créer ainsi une carte topologique cohérente. Parmi les objectifs, l'algorithme doit être utilisable en-ligne et donc fonctionner en temps réel.

Chapitre 6

Descripteur de structure de l'environnement

Sommaire

6.1 Définition d'un lieu topologique

Avant de faire de la segmentation en lieux topologiques, il est nécessaire de définir ce qu'est un lieu topologique. Topologie provient du grec « *topos* » et de « *logos* » qui signifient respectivement *lieu* et *étude*. La topologie est donc l'étude des lieux. Le terme lieu topologique semble donc avoir une définition assez obscure puisqu'il s'agirait d'un lieu caractérisé par son étude. En fait, le terme topologique est utilisé par opposition au terme métrique, qui correspond à la pose du robot dans le repère de référence. La topologie a donc pour objectif d'exprimer les caractéristiques de l'environnement indépendantes d'une métrique. Il s'agit donc d'extraire de l'information de l'environnement pour décrire la topologie du lieu. Quoiqu'il en soit, s'il est possible d'obtenir une définition propre du lieu topologique dans le sens littéral, sa transposition n'est pas forcément évidente d'un point de vue segmentation et cartographie de l'environnement.

Étant donnée la section 5.4 de l'état de l'art, notre objectif est alors de donner une définition générale du lieu topologique valide à la fois pour les environnements d'intérieur et d'extérieur. Les approches les plus générales utilisent la notion de covisibilité de caractéristiques de l'environnement (le sensed space overlap [Blanco *et al.*, 2009], le covgraph [Martín *et al.*, 2012], appariement de points SIFT [Zivkovic *et al.*, 2005], distance visuelle [Goedemé *et al.*, 2007]). Toutefois, aucune définition claire du lieu topologique n'est établie. De plus, la notion de covisibilité induit que les caractéristiques doivent être vues par les différentes observations de l'environnement. Ceci limite donc à la création de lieux topologiques dont l'ensemble des caractéristiques sont visibles depuis n'importe quelle position dans le lieu créé. Un exemple de cette limitation est le cas d'un couloir en « L ». Il s'agit du même couloir mais les caractéristiques présentes dans chaque partie du « L » ne sont pas visibles dans l'autre partie du « L ». Il en résulte alors au moins deux lieux topologiques.

Afin de conserver une définition générique du lieu topologique et pallier les limitations du principe de la covisibilité, nous caractérisons un lieu topologique comme étant un endroit dont les paramètres structurels varient peu quelque soit la position d'observation dans le lieu. Avec cette définition, il est possible de créer des lieux topologiques sur la base de caractéristiques non covisibles mais similaires. Le cas du couloir en « L » engendrerait alors la création d'un seul lieu topologique. La transition entre deux lieux est caractérisée par une variation rapide des paramètres structurels. Ce sera le cas par exemple lors du passage d'un bureau vers un couloir dans un environnement d'intérieur ou alors du passage d'un carrefour à une ligne droite dans un environnement d'extérieur. Cette définition se

rapproche des méthodes élaborées par les auteurs [Tapus et Siegwart, 2005] et [Ranganathan et Dellaert, 2009]. Toutefois, la définition que nous avons adoptée permet de formaliser le concept du lieu topologique contrairement aux deux approches précitées. Ce formalisme permet de définir le type de caractéristiques de l'environnement exploitables ainsi que les conditions d'observation et de groupement de celles-ci. Il reste à préciser ce que sont ces paramètres structurels et comment ils sont extraits à partir des images sphériques. Ces précisions sont détaillées dans la section 6.2.3.

Il est important de remarquer que, étant donnée la définition du lieu topologique retenue, la segmentation de l'environnement dépendra de la façon dont le robot perçoit son environnement. Il est compréhensible que des caméras avec des champs de vue (CDV) différents ne permettront pas la même extraction de paramètres structurels. Les faibles CDV engendreront de plus petits lieux topologiques du fait que les paramètres seront estimés sur une petit partie de l'environnement. Les CDV les plus larges couvrent plus d'information spatiale et permettront de générer des paramètres plus globaux. La segmentation finale obtenue sera contrainte par les caractéristiques physiques des lieux, les conditions d'observation et la nature des capteurs utilisés.

Dans la suite de cette partie, les sections 6.2 et 6.3 sont consacrées à l'explication des paramètres structurels : ce qu'ils sont, ce qu'ils représentent, la façon de les extraire. Cela repose sur la présentation du descripteur GIST et le concept de l'enveloppe spatiale. Ensuite, un descripteur basé sur le concept de l'enveloppe spatiale et calculé à partir d'harmoniques sphériques est élaboré pour permettre une meilleure adaptation à la représentation sphérique. Le chapitre 7 présente l'algorithme de détection en-ligne des variations des paramètres structurels. Le chapitre 8 présente les résultats obtenus.

6.2 Le descripteur GIST

Le descripteur GIST, en tant que descripteur de structure de l'environnement, tient une place centrale dans notre première approche de segmentation de l'environnement en places topologiques. Avant de présenter ce descripteur, un rappel des formalismes mathématiques indispensables à sa compréhension est effectué. Une description détaillée des modifications, nécessaires pour l'adaptation à notre modèle de représentation sphérique, est ensuite fournie.

6.2.1 Transformée de Fourier

Étant donné que les sujets de la transformée de Fourier et du filtrage sont des sujets très vastes, seulement l'essentiel de la théorie et des méthodes utiles à l'explication des algorithmes développés sera abordé. Les problèmes d'échantillonnage ne seront pas traités.

6.2.1.1 Transformée unidimensionnelle

La transformée de Fourier unidimensionnelle permet d'associer un spectre de fréquence à un signal unidimensionnel, c'est-à-dire une fonction d'une seule variable. Le plus souvent, la variable considérée est le temps t, il s'agit alors d'un signal temporel. Soit le signal temporel $s(t)$, le spectre de fréquences associé $S(f)$ s'obtient par l'équation suivante :

$$S(f) = \int_{-\infty}^{+\infty} s(t) e^{-j2\pi ft} dt \tag{6.1}$$

où t représente le temps, f la fréquence et j la variable complexe telle que $j^2 = -1$. Pour retrouver le signal à partir de son spectre, il suffit d'appliquer la transformée de Fourier inverse d'équation :

$$s(t) = \int_{-\infty}^{+\infty} S(f) e^{j2\pi ft} df \tag{6.2}$$

$S(f)$ est une valeur complexe s'écrivant sous la forme :

$$S(f) = |S(f)| e^{j.arg(S(f))} \tag{6.3}$$

Cette écriture permet de faire apparaître le module $|.|$ et la phase $arg(.)$ du spectre complexe du signal. Le module correspond à l'amplitude de chacune des fréquences qui composent le signal tandis que la phase correspond à la position de chacune de ces fréquences. Formellement, la transformée de Fourier permet d'effectuer un changement de base passant d'une base temporelle où chaque élément de la base est un instant t à une base fréquentielle où chaque élément est une sinusoïde de fréquence f. Le module exprime alors les contributions de chacune des sinusoïdes et la phase représente le décalage par rapport à l'origine de chacune d'entre elles. Une interprétation un peu plus intuitive de cette explication sera donnée dans le cas de la transformée de Fourier bidimensionnelle.

La première limitation de cette définition formelle est l'intervalle sur lequel est calculée l'intégrale $[-\infty, +\infty]$. En effet, le signal est :

– supposé périodique, car composé d'une somme de signaux périodiques.
– de durée infinie. Les signaux périodiques sont à base temporelle infinie : si le motif sur la période est connu, il est possible de trouver la valeur du signal à n'importe quel instant t.

En pratique, l'analyse du spectre est effectuée sur des signaux qui ne sont ni périodiques ni à base temporelle infinie. Si les signaux périodiques sont possibles en pratique, ce n'est pas le cas de la base temporelle infinie. L'intégration est alors effectuée sur un intervalle défini $[0, T]$:

$$S(f) = \int_0^T s(t)e^{-j2\pi ft}dt \tag{6.4}$$

La partie 6.2.1.3 traite des effets indésirables générés et des moyens d'y remédier.

La deuxième limitation provient du fait que nos algorithmes traitent des signaux numériques et non analogiques. L'expression de la transformée de Fourier telle quelle n'est valable que pour des signaux continus. Dans le cas des signaux discrets ou numériques, le signal ne possède une valeur qu'à certains instants précis correspondant à l'échantillonnage du signal. Le signal s'écrit sous la forme $s = [s_0, s_1, ..., s_i, ..., s_n]$. Nous parlerons alors de transformée de Fourier discrète (TFD). Le spectre du signal s'obtient par la formule suivante :

$$S_k = \sum_{i=0}^{n} s_i e^{-j2\pi \frac{k}{n} i} \tag{6.5}$$

où i est l'indice temporel et k l'indice fréquentiel. L'expression de la transformée de Fourier discrète prend en compte l'intervalle de définition du signal en n'effectuant la somme que sur les indices $i \in [\![0, n]\!]$. Formellement, elle est aussi définie pour les indices $i \in [\![-\infty, +\infty]\!]$ mais la remarque précédente est déjà incorporée. De même que pour la transformée de Fourier continue, il existe la transformée de Fourier discrète inverse (TFDI) dont l'expression est :

$$s_i = \frac{1}{n} \sum_{k=0}^{n} S_k e^{j2\pi \frac{k}{n} i} \tag{6.6}$$

6.2.1.2 Passage à deux dimensions

La transformée de Fourier bidimensionnelle n'est qu'une extension de la transformée de Fourier unidimensionnelle. Comme précédemment, elle permet d'associer un spectre de fréquence à un signal bidimensionnel, c'est-à-dire une fonction de deux variables. Le cas le plus souvent considéré est la

transformée de Fourier d'une image où les deux variables sont les indices des pixels suivant la hauteur et la largeur. Étant donné que les grandeurs des deux variables sont des distances et que la transformée de Fourier est calculée sur une surface, il s'agit d'un signal spatial. L'image est un signal discret mais avant d'être définie dans le cas discret, la transformée de Fourier bidimensionnelle est définie dans le cas continu. Soit le signal $s(x, y)$ dont le spectre de fréquences associé $S(u, v)$ s'obtient par l'équation :

$$S(u,v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x,y) e^{-j2\pi(ux+vy)} dx.dy \tag{6.7}$$

u, respectivement v, représente les fréquences suivant l'axe de la variable x, respectivement y. Le signal d'origine s'obtient par transformée de Fourier inverse par l'équation :

$$s(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} S(u,v) e^{j2\pi(ux+vy)} du.dv \tag{6.8}$$

La transformée de Fourier discrète bidimensionnelle (TFD 2D) est obtenue en considérant les diverses remarques déjà faites pour le cas unidimensionnel (limitation de l'intervalle d'intégration et passage à l'équation discrète). Étant donné que la TFD 2D s'applique à des images, le signal $s_{i,j}$ correspond à une image ne comportant qu'un seul canal. C'est-à-dire $s_{i,j} \in [\![0, 255]\!]$ avec $(i, j) \in [\![0, m]\!] \times [\![0, n]\!]$ avec $m + 1$ la largeur de l'image en pixels et $n + 1$ la hauteur de l'image en pixels. L'équation de transformation est la suivante :

$$S_{k,l} = \sum_{i=0}^{m} \sum_{j=0}^{n} s_{i,j} e^{-j2\pi\left(\frac{k}{m}i + \frac{l}{n}j\right)} \tag{6.9}$$

Pour retrouver l'image à partir de son spectre, il suffit d'appliquer la transformée de Fourier discrète bidimensionnelle inverse (ITFD 2D) donnée par l'équation suivante :

$$s_{i,j} = \sum_{k=0}^{m} \sum_{l=0}^{n} S_{k,l} e^{j2\pi\left(\frac{k}{m}i + \frac{l}{n}j\right)} \tag{6.10}$$

L'application de la TFD 2D à l'image permet d'illustrer l'interprétation faite sur le module et la phase. Les TFD 2D respectives des images du guépard et du zèbre de la figure 6.1 sont calculées. En mélangeant ensuite les amplitudes et phases des spectres des deux images, il résulte un spectre contenant la phase du guépard et le module du zèbre et un autre spectre contenant la phase du zèbre et le module du guépard. La figure 6.2 permet de visualiser les images obtenues par ITFD 2D de ces deux spectres. Il est aisé de reconnaître le guépard, respectivement le zèbre, dans l'image contenant la phase du guépard, respectivement du zèbre. Ceci permet d'illustrer le fait que l'information de

position est contenue dans la phase et non dans le module. Ce dernier ne contient que l'information de contribution de chacune des sinusoïdes constituant l'image.

FIG. 6.1 – Images extraites des slides de D. A. Forsyth dans le cours Advances in Computer Vision du MIT, images du livre Computer Vision - A Modern Approach [Forsyth et Ponce, 2002]. (a) Guépard et (b) zèbre.

6.2.1.3 Fenêtrage et zero-padding

Comme évoqué dans 6.2.1.1, calculer la transformée de Fourier -continue ou discrète, en unidimensionnel ou bidimensionnel- sur l'intervalle de définition du signal plutôt que sur un intervalle infini engendre des erreurs dans le spectre résultant. Afin de simplifier la démonstration, l'illustration est faite avec un signal sinusoïdal unidimensionnel mais les explications restent valables pour le cas bidimensionnel.

Les différentes étapes sont illustrées sur la figure 6.3. Le signal de référence est la sinusoïde en haut à gauche. Le module de sa transformée de Fourier idéale (résultat mathématique sur un signal infini ou sur une période dans le cas de signaux périodiques) correspond à deux Diracs en $-f$ et f, f étant la fréquence de la sinusoïde. Ce résultat est illustré en haut à droite. Réduire l'intervalle d'intégration correspond mathématiquement à multiplier le signal par une fenêtre carrée de largeur de l'intervalle d'intégration. Le signal étudié, connu sur l'intervalle $[-\infty, +\infty]$, est considéré réduit par multiplication

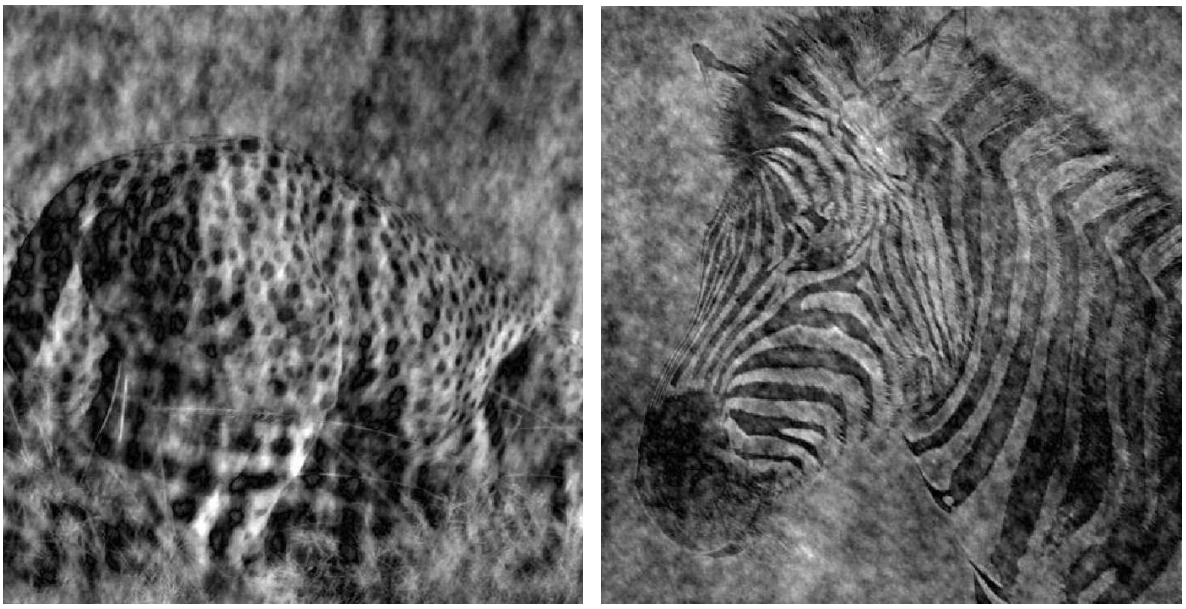


FIG. 6.2 – Images extraites des slides de D. A. Forsyth dans le cours Advances in Computer Vision du MIT, images du livre Computer Vision - A Modern Approach [Forsyth et Ponce, 2002]. (a) Image contenant la phase du guépard et le module du zèbre et (b) image contenant la phase du zèbre et le module du guépard.

de la fenêtre carrée. Cette dernière correspond à l'intervalle exact d'étude du signal (signal mesuré). Elle est illustrée sur la figure au milieu à gauche. L'inconvénient est que cette fenêtre possède aussi un spectre de fréquences qui va interférer avec celui de la sinusoïde. Le module du spectre de la fenêtre est représenté au milieu à droite. Le signal d'étude résultant est illustré en bas à gauche. L'interférence entre le spectre de la sinusoïde et celui de la fenêtre est représentée en bas à droite. Les deux Diracs de la sinusoïde sont toujours présents mais additionnés d'un bruit fréquentiel ajouté par la fenêtre. Les lobes non centraux sont indésirables car ils correspondent à des fréquences supplémentaires qui viennent perturber le signal d'origine. Pour remédier à ce problème, il existe de nombreuses fenêtres qui sont appliquées sur le signal avant de calculer la transformée de Fourier. Elles ont pour avantage d'avoir des lobes non centraux bien moins importants que ceux de la fenêtre carrée. Le spectre ainsi obtenu est plus épuré en terme de fréquences parasites. Ces fenêtres sont les fenêtres de Hanning, Hamming, Tukey, Bartlett, Lanczos, ...*etc.*

Les différents types de fenêtres ne sont pas plus développés car il ne s'agit pas de l'objet d'étude. Ce qu'il est important d'observer est que l'information de signal périodique est toujours présente mais additionnée de fréquences dans ce cas dites parasites. Malgré cette réduction de l'intervalle d'intégration,

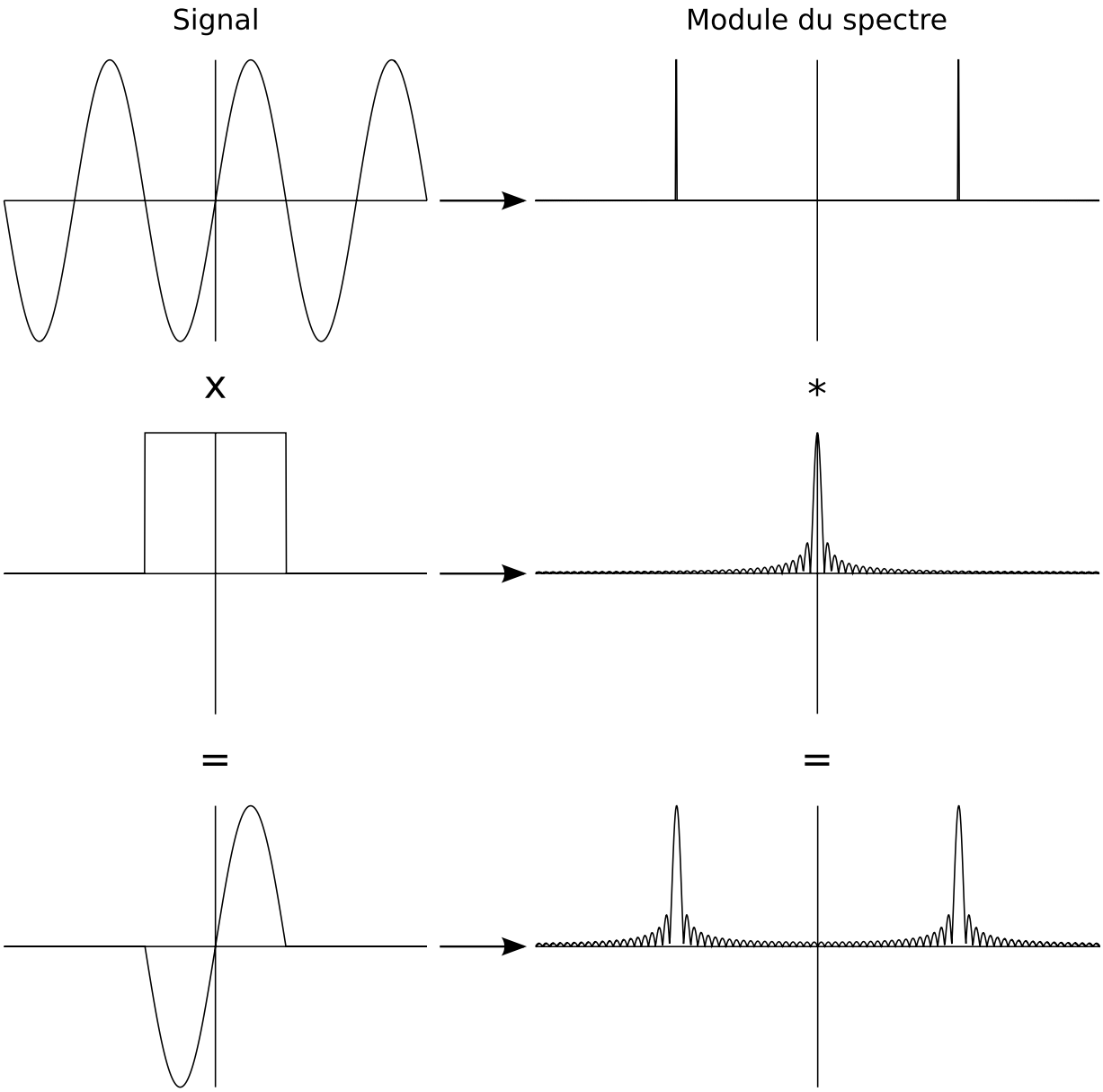


FIG. 6.3 – Les signaux étudiés se trouvent dans la colonne de gauche et le module de leurs spectres respectifs se trouvent dans la colonne de droite. La première ligne est la sinusoïde idéale. La seconde ligne correspond à la fenêtre carrée qui limite l'intervalle d'intégration lors de la transformée de Fourier. La dernière ligne montre l'influence de cette fenêtre sur le signal et sur le spectre de la sinusoïde.

la transformée de Fourier considère le signal étudié comme périodique ; ce qui est logique puisqu'il s'agit d'un changement d'espace de représentation et, dans le domaine fréquentiel, les éléments de base de l'espace sont des sinusoïdes. Toutefois, tous les signaux étudiés ne sont pas forcément périodiques et il est nécessaire de pouvoir le prendre en compte. Dans le cas de signaux non périodiques, les fréquences jugées parasites précédemment sont cette fois nécessaires, elles représentent les fréquences

supplémentaires du signal. L'étalement fréquentiel est caractéristique de signaux non périodiques car ils sont composés de nombreuses fréquences afin de traduire ou compenser toutes les variations du signal. L'exemple le plus simple à interpréter est le Dirac temporel qui est un signal non périodique mais dont le spectre est constitué de toutes les fréquences. Un autre exemple est la fenêtre carrée qui est aussi un signal non périodique. Afin de traduire les angles de la fenêtre, de hautes fréquences sont nécessaires. Si elles sont absentes, des sinusoïdes apparaissent aux angles de la fenêtre (effet Gibbs).

Outre son utilisation pour le ré-échantillonnage dans le domaine audio ou son utilité pour augmenter la précision dans le spectre discret, le zero-padding permet de supprimer la considération périodique du signal étudié. Le zero-padding consiste, dans ce cas, à ajouter des zéros à la fin du signal avant de calculer sa transformée de Fourier. Ceci crée une discontinuité permettant de prendre en compte les fréquences constitutives du signal nul. Il en résulte un étalement fréquentiel.
Le mécanisme est équivalent à multiplier le signal par une fenêtre carrée réduisant l'intervalle d'intégration. Le signal ainsi obtenu est ensuite re-multiplié par une fenêtre carrée plus large (voir figure 6.4 pour une visualisation de la fenêtre ajoutant le zero-padding). La transformée de Fourier d'une fenêtre carrée est un sinus cardinal ($sinc(x) = \frac{sin(x)}{x}$, avec $sinc(0) = 1$) dont la largeur des lobes est inversement proportionnelle à la largeur de la fenêtre. Le spectre de la fenêtre carrée est celui de la figure 6.3 en bas à droite. Dans le cas du zero-padding, le signal est alors convolué une première fois par le sinus cardinal de la fenêtre de réduction de l'intervalle. Il est convolué une deuxième fois par le sinus cardinal de la fenêtre ajoutant le zero-padding. Cette deuxième convolution engendre une hausse de l'amplitude des fréquences des lobes non centraux. Il y a bien un étalement fréquentiel traduisant la non périodicité du signal étudié, à savoir un sinus périodique complété d'un signal nul.

L'explication du zero-padding reste valable pour des signaux bien plus complexes et pour les cas multidimensionnels. Elle sera notamment utile pour la considération de la périodicité spatiale des images sphériques.

6.2.1.4 Filtrage

Le filtrage est une opération constamment utilisée lors du traitement de signaux, que ce soit pour modifier les caractéristiques d'un signal ou pour modéliser les effets d'un canal sur celui-ci. Le concept du filtrage est très simple et se représente par le schéma de la figure 6.5. Il consiste à faire entrer un signal $e(t)$ dans un filtre possédant ses propres caractéristiques $h(t)$, il en ressort un signal modifié $s(t)$.

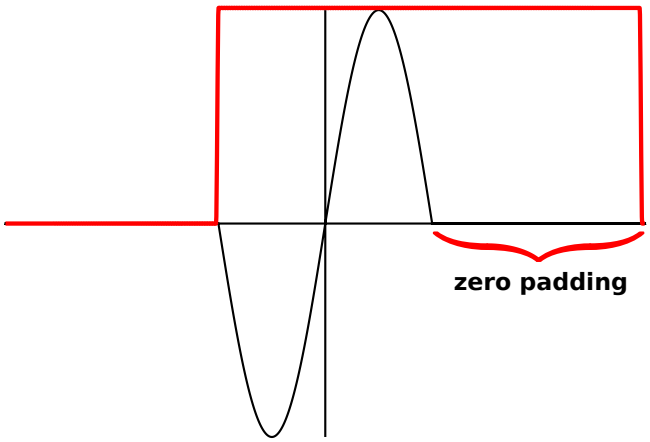


FIG. 6.4 – Interprétation du zero-padding par multiplication de fenêtres carrées

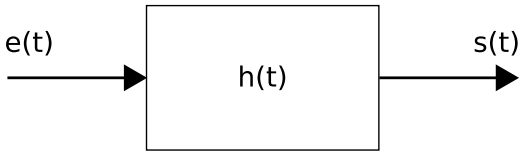


FIG. 6.5 – Modèle de filtre

La modélisation mathématique du filtrage est la suivante : pour obtenir le signal de sortie $s(t)$, il faut effectuer la convolution du signal d'entrée $e(t)$ avec le signal du filtre $h(t)$. L'équation de filtrage s'écrit :

$$s(t) = (e * h)(t) = \int_{-\infty}^{+\infty} e(t-\tau)h(\tau)d\tau \tag{6.11}$$

Cette écriture n'est ni très pratique ni très intuitive. Pour étudier les problèmes de filtrage, la transformée de Fourier est utilisée pour passer les signaux dans le domaine fréquentiel. L'avantage de travailler dans le domaine fréquentiel est que le produit de convolution devient une multiplication comme le décrit l'équation 6.12.

$$S(f) = H(f)E(f) \tag{6.12}$$

L'écriture complexe du spectre, obtenue à l'équation 6.3, permet de réécrire l'équation précédente :

$$S(f) = |H(f)|e^{j.arg(H(f))}|E(f)|e^{j.arg(E(f))} \tag{6.13}$$

$$= |H(f)||E(f)|e^{j(arg(H(f))+arg(E(f)))} \tag{6.14}$$

Pour analyser le résultat d'un filtrage, il suffit alors de multiplier les modules des spectres et d'additionner les phases du signal d'entrée et du signal du filtre. L'écriture de l'équation de filtrage dans le domaine fréquentiel permet de mieux comprendre l'influence d'un filtre sur un signal. Voir quelles seront les fréquences atténuées/amplifiées en faisant le produit des modules des spectres est bien plus aisé que par l'étude du produit de convolution. Par ailleurs, un filtre n'est jamais défini dans le domaine temporel mais toujours dans le domaine fréquentiel. La phase du filtre indique le retard qu'ajoute le filtre au signal. Elle a son importance mais le gabarit du filtre repose sur le module. Le filtrage consiste essentiellement à enlever une partie des fréquences constituantes du signal d'entrée.

$$|S(f)| = |H(f)||E(f)| \tag{6.15}$$

$$arg(S(f)) = arg(H(f)) + arg(E(f)) \tag{6.16}$$

La théorie du filtrage exposée ici, incomplète mais suffisante pour les besoins de cette thèse, repose sur un signal temporel unidimensionnel. Elle reste valable pour les signaux bidimensionnels. Les équations sont alors les suivantes :

Convolution des signaux (domaine spatial) :

$$s(x,y) = (h * e)(x,y) \tag{6.17}$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e(x-\mu, y-\nu)h(\mu,\nu)d\mu d\nu \tag{6.18}$$

Multiplication des spectres complexes (domaine fréquentiel) :

$$S(u,v) = H(u,v)E(u,v) \tag{6.19}$$

$$= |H(u,v)||E(u,v)|e^{j(arg(H(u,v))+arg(E(u,v)))} \tag{6.20}$$

$$|S(u,v)| = |H(u,v)||E(u,v)| \tag{6.21}$$

$$arg(S(u,v)) = arg(H(u,v)) + arg(E(u,v)) \tag{6.22}$$

La théorie est aussi valable pour des signaux discrets et notamment des images. Il suffit d'adapter les équations précédentes en fonction de ce qui a été présenté sur la transformée de Fourier (6.2.1.1 et 6.2.1.2).

6.2.2 Filtre de Gabor

Le filtre de Gabor est un filtre, comme défini précédemment, permettant de conserver seulement les fréquences désirées suivant un gabarit particulier : il est défini dans les domaines temporel et spatial

comme étant une sinusoïde modulée par une gaussienne. Le filtre de Gabor existe en une dimension (temporel) mais les explications suivantes sont faites à partir du cas à deux dimensions (spatial). Ce cas est plus riche et correspond à la façon de l'utiliser dans les algorithmes présentés. Notamment, la fréquence centrale et l'orientation du filtre sont définies dans le domaine fréquentiel. Ces deux paramètres sont le fondement même de l'utilité de ce type de filtre dans le domaine des images.

Le filtre de Gabor est une fonction complexe définie dans le domaine spatial par :

$$g(x, y) = s(x, y)w(x, y) \tag{6.23}$$

où $s(x, y)$ est une sinusoïde complexe, nommée porteuse dans le cadre du filtrage et de la modulation de fréquence. $w(x, y)$ est une Gaussienne bidimensionnelle, nommée enveloppe. La sinusoïde complexe est définie par l'équation :

$$s(x, y) = e^{j(2\pi(u_0 x + v_0 y) + \phi)} \tag{6.24}$$

avec (u_0, v_0) la fréquence spatiale et ϕ la phase. Dès à présent $\phi = 0$ car il s'agit d'une composante de phase additive. Comme vu précédemment, la phase correspond à une notion de position. ϕ correspond donc à un décalage par rapport à l'origine du repère dans l'espace des fréquences. Ce décalage ne présentant aucun intérêt dans notre approche, la composante de phase additive est annulée. L'équation de sinusoïde complexe devient simplement :

$$s(x, y) = e^{j2\pi(u_0 x + v_0 y)} \tag{6.25}$$

Étant donné qu'il est plus intéressant de travailler avec les paramètres de fréquence centrale et d'orientation du filtre, il est utile d'exprimer la sinusoïde en coordonnées polaires. La fréquence centrale F_0 et l'orientation θ_0 s'obtiennent par les équations suivantes :

$$F_0 = \sqrt{u_0^2 + v_0^2} \tag{6.26}$$

$$\theta_0 = arctan\left(\frac{v_0}{u_0}\right) \tag{6.27}$$

La transformation inverse s'écrit :

$$u_0 = F_0 cos(\theta_0) \tag{6.28}$$

$$v_0 = F_0 sin(\theta_0) \tag{6.29}$$

L'expression de la sinusoïde complexe, en intégrant le passage en coordonnées polaires, devient :

$$s(x,y) = e^{j2\pi F_0(xcos(\theta_0)+ysin(\theta_0))} \tag{6.30}$$

La Gaussienne quant à elle s'exprime de la façon suivante :

$$w(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{1}{\sigma_x^2}(x-x_0)_r^2+\frac{1}{\sigma_y^2}(y-y_0)_r^2\right)} \tag{6.31}$$

L'indice $_r$ correspond à une rotation d'angle θ dans le sens horaire du vecteur $[x,y]$ par rapport au centre de la Gaussienne $[x_0,y_0]$ telle que :

$$R = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \tag{6.32}$$

$$\begin{bmatrix} (x-x_0)_r \\ (y-y_0)_r \end{bmatrix} = R \begin{bmatrix} (x-x_0) \\ (y-y_0) \end{bmatrix} \tag{6.33}$$

$$\begin{bmatrix} (x-x_0)_r \\ (y-y_0)_r \end{bmatrix} = \begin{bmatrix} (x-x_0)cos(\theta)+(y-y_0)sin(\theta) \\ -(x-x_0)sin(\theta)+(y-y_0)cos(\theta) \end{bmatrix} \tag{6.34}$$

σ_x et σ_y sont les écart-types de la Gaussienne respectivement suivant les variables x et y. L'expression complète du filtre de Gabor dans le domaine spatial, en coordonnées polaires, est alors :

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{1}{\sigma_x^2}(x-x_0)_r^2+\frac{1}{\sigma_y^2}(y-y_0)_r^2\right)} e^{j2\pi F_0(xcos(\theta_0)+ysin(\theta_0))} \tag{6.35}$$

Comme pour tous les filtres, son influence dans le domaine fréquentiel est analysée au travers de sa transformée de Fourier. La démonstration du calcul de la transformée de Fourier du filtre de Gabor est fournie en annexe A. Étant donné que la transformée de Fourier d'une sinusoïde est un Dirac et que celle d'une Gaussienne est une Gaussienne, la transformée de Fourier du filtre de Gabor, résultant de la convolution de ces deux spectres, est simplement une Gaussienne centrée sur le Dirac correspondant à la fréquence de la sinusoïde (d'où les noms de porteuse et d'enveloppe). L'expression du spectre du filtre de Gabor est alors la suivante :

$$G(u,v) = e^{-\frac{1}{2}\left(\frac{1}{\sigma_u^2}(u-u_0)_r^2+\frac{1}{\sigma_v^2}(v-v_0)_r^2\right)} e^{-j2\pi((u-u_0)x_0+(v-v_0)y_0)} \tag{6.36}$$

avec $\sigma_u = \frac{1}{2\pi\sigma_x}$ et $\sigma_v = \frac{1}{2\pi\sigma_y}$. En analysant l'équation, il s'agit bien d'une Gaussienne centrée autour de (u_0, v_0), soit (F_0, θ_0) en coordonnées polaires. θ, présent dans la matrice de rotation indiquée par l'indice $_r$, est le paramètre de réglage de l'orientation du filtre. Le deuxième terme étant une exponentielle complexe, il n'agit que sur la phase du filtre. Il ne présente alors pas d'intérêt dans le cas étudié ici où seul le module est utilisé. Nous noterons simplement que ce terme dépend de la position (x_0, y_0) de la Gaussienne dans le domaine spatial et vérifie alors le lien entre la position et la phase. L'expression du module du filtre de Gabor est donc :

$$|G(u,v)| = e^{-\frac{1}{2}\left(\frac{1}{\sigma_u^2}(u-u_0)_r^2 + \frac{1}{\sigma_v^2}(v-v_0)_r^2\right)} \tag{6.37}$$

Les paramètres de réglages du filtre de Gabor sont :

- La fréquence centrale (u_0, v_0) en coordonnées cartésiennes (ou (F_0, θ_0) en coordonnées polaires).
- Les écart-types de la Gaussienne (σ_u, σ_v) qui paramètrent son étalement.
- L'orientation du filtre θ

Ces paramètres et leurs influences sont résumés dans la figure 6.6.

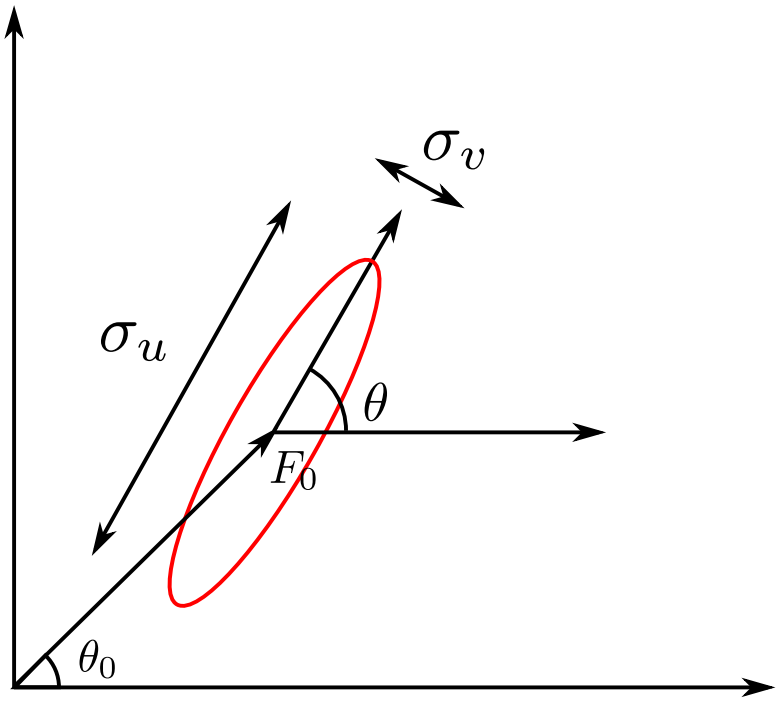


FIG. 6.6 – Filtre de Gabor et paramètres de réglages $(F_0, \theta_0, \sigma_u, \sigma_v, \theta)$

6.2.3 Présentation du GIST

6.2.3.1 Introduction

Initialement proposé par A. Oliva et A. Torralba dans [Oliva et Torralba, 2001] et [Oliva et Torralba, 2006], le but du descripteur GIST était à l'origine de permettre la classification automatique d'images. Partant du constat que l'être humain est capable de classifier une image en très peu de temps et quelque soit sa complexité, l'idée était de recréer un système capable d'effectuer la même opération. Il s'agissait alors de capturer l'information essentielle contenue dans l'image : son gist (*i.e.* son essence). Le GIST est un descripteur global permettant d'extraire l'enveloppe spatiale de l'image qui contient l'essentiel de l'information. L'enveloppe spatiale correspond aux différentes fréquences et orientations contenues dans l'image. En dégradant une image en ne conservant que les fréquences et orientations principales, il est toujours possible d'identifier les paramètres principaux de l'image et de la classer. Étant donné qu'il s'agit d'un descripteur global capturant l'essentiel de l'information, ces paramètres ne peuvent pas concerner les détails de l'image. Les paramètres de classification ont été définis dans les travaux sur le GIST comme étant la possibilité de distinguer un environnement naturel d'un environnement urbain, le degré d'ouverture de l'environnement (par exemple une plage ou une montagne), la rugosité, le caractère accidenté d'un environnement naturel et le degré d'expansion pour un environnement urbain. Le classifieur complet, basé sur une SVM, a été entraîné suivant ces paramètres sur une base de données d'entraînement. La classe d'appartenance d'une image s'obtient simplement en fournissant son GIST au classifieur.

Montrant de bonnes performances dans la classification automatique, le descripteur GIST a rapidement été utilisé pour d'autres tâches. Il a notamment été utilisé par [Douze *et al.*, 2009] pour évaluer ses performances dans le cadre de la recherche et de la comparaison d'images à l'échelle du web. Les auteurs [Murillo et Košecká, 2009] utilisent le GIST afin de faire de la reconnaissance de lieux. Cette dernière approche est étendue à la détection de fermeture de boucle dans [Singh et Košecká, 2010]. Mis à part le cas de la reconnaissance d'image à l'échelle du web où les résultats sont discutables sur l'utilisation d'un descripteur global d'une manière générale en fonction de la précision du résultat souhaité, les résultats obtenus dans les approches de reconnaissance de lieu et de détection de fermeture de boucle sont convaincants.

6.2.3.2 Principe de fonctionnement

Dans le cadre des travaux présentés ici, le GIST est utilisé en tant que descripteur de structure de l'environnement, dénommée enveloppe spatiale dans les travaux de A. Oliva et A. Torralba. Son principe de fonctionnement est présenté avant d'expliquer les modifications apportées dans le cadre des algorithmes développés. L'information essentielle de l'image, comme vu précédemment, est contenue dans les fréquences et orientations de celle-ci. Chercher les fréquences contenues dans une image implique le calcul de sa TFD 2D (*cf.* section 6.2.1.2). Les images visuellement proches possèdent des spectres similaires. La figure 6.7 montre des images d'environnements ainsi que leurs spectres et les modèles de spectres auxquels il se rapportent.

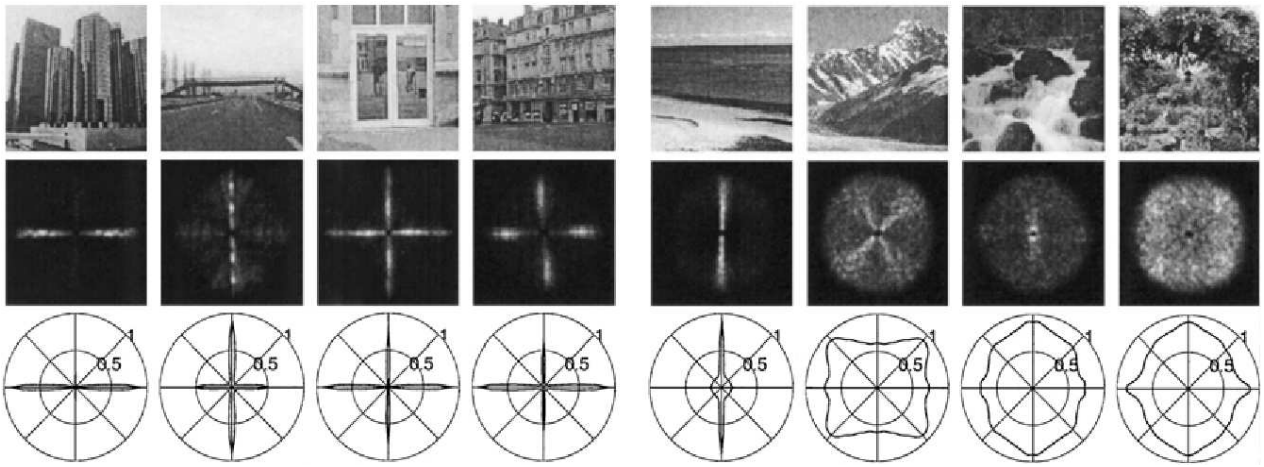


FIG. 6.7 – Image extraite des travaux de A. Oliva et A. Torralba [Oliva et Torralba, 2001]. La première ligne contient des images exemples. La seconde ligne correspond aux spectres des images. La troisième ligne représente les patterns types auxquels les spectres se rapportent.

Les fréquences et orientations les plus importantes sont facilement déterminables à partir du spectre des images. Le principe du GIST est alors de récupérer ces fréquences et orientations principales. Pour cela, plusieurs filtres de Gabor (*cf.* section 6.2.2) sont utilisés. La banque de filtres utilisée regroupe plusieurs filtres de Gabor chacun défini par une fréquence centrale et une orientation. Le nombre de filtres est paramétrable en fonction du nombre de fréquences centrales et d'orientations désirées. En pratique, le choix des filtres utilisés est déterminé de manière à ne pas avoir de recouvrement des spectres de chacun des filtres dans les zones de maximum d'amplitude ; il y a seulement recouvrement dans les zones de fortes atténuations. Les fréquences centrales F_0 sont alors automatiquement déterminées en fonction de l'étalement de la Gaussienne dans le domaine fréquentiel et du nombre

de filtres souhaités afin de limiter le recouvrement. De même, les orientations θ sont déterminées de manière à limiter le recouvrement et de façon à orienter le filtre suivant l'axe origine-centre du filtre, soit $\theta = \theta_0$ (*cf.* 6.2.2). Un exemple de superposition de filtres est donné dans le figure 6.9. Le nombre de filtres agit sur la granularité du résultat et donc sur sa précision. Plus il y a aura de filtres et plus il sera possible d'avoir une analyse fine du spectre de l'image.

Pour appliquer les filtres de Gabor à l'image, cela se fait simplement par multiplication du module du spectre du filtre de Gabor avec le module du spectre de l'image. Comme expliqué dans la section 6.2.1.4, le filtrage correspond à une multiplication dans le domaine fréquentiel. Le résultat obtenu est donc une image de la taille de l'image d'origine représentant le spectre filtré par le filtre de Gabor. Il en résulte alors autant de spectres que de filtres. Pour une image en noir et blanc, le résultat correspond au filtrage direct de l'image par la banque de filtres. Pour les images en couleurs, chaque canal est traité indépendamment ; le résultat correspond à la concaténation du filtrage de chacun des canaux. Le descripteur GIST final est la concaténation de tous ces résultats de filtrage dans un seul vecteur de taille :

$$S_d = N^2 N_f N_c \tag{6.38}$$

avec S_d la taille du vecteur, N la taille de l'image sachant que l'algorithme ne traite que des images carrées (d'où le terme N^2), N_f le nombre de filtres de Gabor et N_c le nombre de canaux de l'image. Une première réduction de la taille du descripteur est obtenue en prenant la moyenne des valeurs des spectres filtrés sur de petits blocs d'image carrés sans recouvrement. En considérant w comme étant la taille de la fenêtre dans laquelle la moyenne est calculée, la taille du descripteur devient :

$$S_d = \frac{N^2 N_f N_c}{w^2} \tag{6.39}$$

Le descripteur résultant est de grande dimension mais décrit globalement l'image, il s'agit de la concaténation de l'information essentielle de l'image permettant sa classification efficace suivant les paramètres définis précédemment. Ce descripteur décrit alors la structure de l'environnement.

Il est important de savoir que l'image est pré-traitée. En effet, le spectre n'est pas calculé sur l'image telle qu'elle. L'image est « zero-paddée » suivant les deux axes permettant d'obtenir une image

carrée à partir d'une image rectangulaire. L'intérêt principal réside toutefois dans l'influence que le zero-padding a sur le spectre de l'image (*cf.* section 6.2.1.3).

6.2.4 Modification du GIST

Le descripteur GIST tel que présenté précédemment possède quelques inconvénients vis-à-vis de l'approche développée. D'une part, il est originellement conçu pour être calculé sur des images carrées. Or les projections de sphères dans le plan sont des images rectangulaires (panorama sphérique incomplet). D'autre part, il n'est pas prévu pour prendre en compte la périodicité spatiale introduite par le modèle de représentation sphérique. Étant donné le calcul du GIST et la représentation sphérique utilisée, il ne sera pas possible de prendre complètement en compte la périodicité offerte par la sphère. L'axe horizontal du panorama sphérique correspond à l'angle $\theta \in [0, 2\pi]$ des coordonnées sphériques, soit un déplacement dans le plan longitudinal. L'axe vertical correspond à l'angle $\phi \in [\frac{-\pi}{2}, \frac{\pi}{2}]$, soit un déplacement en latitude sur la sphère. Nous obtenons une périodicité horizontale de l'image mais pas de périodicité verticale. C'est-à-dire que le bord droit de l'image peut être joint au bord gauche sans créer de discontinuité dans l'information visuelle de l'image. Il n'est par contre pas possible d'effectuer la même opération avec le bord haut de l'image et le bord bas. L'image 6.8 permet de visualiser ces considérations de périodicité. D'ailleurs, θ est défini sur un intervalle de largeur 2π représentant le tour complet d'un cercle tandis que ϕ n'est défini que sur un intervalle de largeur π.

FIG. 6.8 – Panorama résultant de la projection de la vue sphérique.

Afin d'introduire cette périodicité dans le calcul du GIST, il suffit simplement de supprimer le zero-padding suivant l'axe horizontal (*cf.* section 6.2.1.3). Ceci implique donc de supprimer le système qui permettait de convertir une image rectangulaire en image carrée pour le calcul du GIST. Pour remédier à cela, deux possibilités existent : effectuer un zero-padding suivant l'axe vertical de façon à obtenir

une image carrée (sachant que la hauteur des images traitées est toujours inférieure à la largeur) ou alors adapter le calcul du GIST au cas des images rectangulaires. Nous avons opté pour la deuxième solution car elle permet un calcul du GIST plus générique avec possibilité pour l'utilisateur de choisir son propre zero-padding si nécessaire. De plus, le ratio hauteur-largeur des images est d'environ 1/4. Cela engendrerait une image carrée constituée aux trois quarts de zéros. Bien qu'un zero-padding vertical soit nécessaire, un tel agrandissement de l'image entraîne un ralentissement dans le calcul du GIST avec une quantité de zéros bien plus grande que nécessaire. Enfin, adapter le calcul du GIST aux images rectangulaires n'est pas compliqué puisqu'il suffit de prendre en compte les dimensions de l'image dans le calcul de la TFD 2D 6.2.1.2 et celui du filtre de Gabor 6.2.2. Si une des dimensions de l'image est plus grande que l'autre, cela doit entraîner un élargissement du filtre de Gabor suivant cet axe dans le domaine fréquentiel car la résolution fréquentielle sera plus importante (plus de pixels pour représenter le même intervalle de fréquences). Pour rappel, l'équation de la TFD 2D est :

$$S_{k,l} = \sum_{i=0}^{m} \sum_{j=0}^{n} s_{i,j} e^{-j2\pi\left(\frac{k}{m}i + \frac{l}{n}j\right)} \tag{6.40}$$

En utilisant l'équation de la TFD 2D et l'équation du filtre de Gabor, l'équation discrète du module du spectre du filtre de Gabor s'écrit :

$$|G_{k,l}| = e^{-\frac{1}{2}\left(\frac{1}{\sigma_u^2}\left(\frac{k}{m} - \frac{k_0}{m}\right)_r^2 + \frac{1}{\sigma_v^2}\left(\frac{l}{n} - \frac{l_0}{n}\right)_r^2\right)} \tag{6.41}$$

avec dans les deux équations précédentes $k \in [\![0, m]\!]$, $k_0 \in [\![0, m]\!]$, $l \in [\![0, n]\!]$ et $l_0 \in [\![0, n]\!]$. m et n sont respectivement la largeur et la hauteur de l'image sur laquelle sont appliqués la TFD 2D et le filtre de Gabor. Dans le calcul du GIST original $m = n$. Cela traduit le fait que l'image traitée soit carrée et permet de simplifier les équations. Dans l'approche développée, m et n sont réintégrées avec la possibilité d'avoir $m \neq n$. La fréquence centrale du filtre devient alors :

$$F_0 = \sqrt{\left(\frac{k_0}{m}\right)^2 + \left(\frac{l_0}{n}\right)^2} \tag{6.42}$$

Au travers des équations 6.41 et 6.42, il est à noter que la dimension de l'image influe sur l'étalement du filtre de Gabor. L'étalement est proportionnel sur chaque axe à la dimension respective de l'image. Il s'agit au final d'une simple mise à l'échelle avec respect des proportions de l'image.

La figure 6.9 représente la banque de filtres de Gabor utilisée dans notre approche. Les filtres sont générés automatiquement avec pour consigne le nombre de fréquences, le nombre d'orientations et les dimensions de l'image.

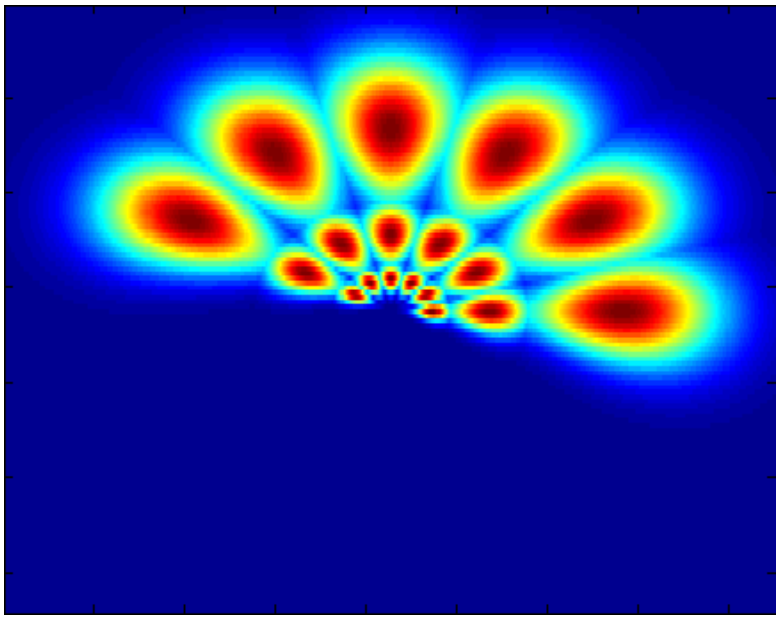


FIG. 6.9 – Modules des 18 filtres de Gabor superposés. La banque de filtres est suivant 3 fréquences et 6 orientations. Le rouge correspond aux valeurs élevées tandis que le bleu correspond aux faibles valeurs.

6.2.5 Réduction de la dimension du descripteur

Le descripteur GIST obtenu précédemment est de très grande dimension. En général, le descripteur est de 320 dimensions par canal. Il en résulte un descripteur de 320 dimensions pour une image en noir et blanc et de 960 dimensions pour une image en couleurs. Comme il s'agit d'un descripteur global, l'information essentielle est concaténée dans cet unique descripteur. Dans une application ne nécessitant que l'analyse de la structure de l'environnement, un descripteur très dégradé contenant un résumé de l'information est suffisant. Ceci dépend toutefois de l'objectif visé ; dans le cadre de la classification dans les travaux de A. Oliva et A. Torralba, il est important de conserver une quantité d'information suffisante pour la distinction des différents types de lieux. Dans le cas de l'analyse de la structure de l'environnement pour la segmentation, l'objectif est simplement d'observer les variations structurelles au fur et à mesure du déplacement du robot. Un descripteur de très faible dimension est alors suffisant comme les résultats (*cf.* section 8) le prouvent. Une réduction drastique de la dimension

du descripteur est alors effectuée dans le cadre de nos travaux.

Une méthode classique de réduction de dimension est la *Principal Component Analysis* (PCA) [Jolliffe, 2002]. Cette méthode permet d'extraire les vecteurs suivant lesquels la variabilité est maximale. Ils encodent donc un maximum d'information. Le descripteur devient alors une combinaison linéaire de ces vecteurs. Pour obtenir une réduction, il suffit de conserver les coefficients des vecteurs principaux en choisissant le pourcentage d'information à conserver. Un inconvénient majeur de cette approche est la détermination des vecteurs principaux (*i.e.* les composantes principales). Pour déterminer des composantes principales significatives, il est nécessaire d'avoir une base de données de descripteurs d'entraînement. Les méthodes utilisant la PCA nécessitent alors une connaissance a priori empêchant la réalisation de processus d'apprentissage en-ligne. D'autre part, il est nécessaire d'avoir des descripteurs suffisamment variés pour représenter tous les environnements possibles. Si seulement des environnements de types plages et urbains sont pris en compte pour l'élaboration des composantes principales, il sera difficile d'obtenir une projection correcte pour un environnement de montagne. Les coefficients alors obtenus ne seront pas significatifs. La base d'entraînement doit alors être suffisamment large pour permettre une élaboration correcte des composantes principales. La méthode PCA est donc tributaire du problème classique de la base d'apprentissage et de sa pertinence.

Une autre approche est la transformation de Karhunen-Loeve (KLT) qui présente l'avantage d'avoir une projection suivant des composantes principales décorrelées. Les difficultés de la technique PCA subsistent. Ces méthodes sont très efficaces mais pas toujours simples à mettre en œuvre. Il est nécessaire d'avoir suffisamment d'échantillons représentatifs pour déterminer les composantes principales suivant lesquelles les vecteurs pourront être projetés.

Afin de réduire la taille du descripteur, la solution choisie est d'étendre l'idée déjà utilisée dans le calcul du GIST. Comme décrit dans la section 6.2.3.2, la taille du descripteur est $S_d = \frac{N^2 N_f N_c}{w^2}$ grâce à un moyennage sur une fenêtre carrée de largeur w. En prenant une fenêtre non carrée de la taille de l'image, le résultat du filtrage est résumé par la moyenne des réponses du filtre de Gabor suivant chacune des fréquences. L'interprétation de ce résultat est qu'il s'agit de la contribution totale des fréquences et orientations de l'image pondérées par le filtre de Gabor choisi. Plutôt que d'avoir un résultat précis de filtrage pour un couple de fréquences (u, v), le résultat prend en compte un lot de

fréquences autour de la fréquence centrale du filtre et suivant son orientation. Il s'agit de la réponse globale du filtrage par le filtre de Gabor. Il en résulte une valeur par filtre de Gabor et par canal de l'image. La dimension du descripteur final est alors :

$$S_d = N_f N_c \tag{6.43}$$

Les conventions sont les mêmes que précédemment. Par ailleurs, l'information de structure apparaît nettement dans une image en noir et blanc. Il n'est donc pas nécessaire d'utiliser une image couleur. Nous obtenons alors un descripteur dont la taille correspond directement au nombre de filtres de Gabor utilisés :

$$\begin{aligned} S_d &= N_f & (6.44)\\ &= N_{freq} N_{or} & (6.45) \end{aligned}$$

avec N_{freq} le nombre de fréquences et N_{or} le nombre d'orientations choisies. Pour nos expérimentations, nous avons choisi 3 fréquences et 6 orientations, soit un total de 18 filtres de Gabor. Le descripteur final est, dans ce cas, de seulement 18 dimensions. Il s'agit d'un descripteur global de très faible dimension comportant l'essentiel de l'information structurelle de l'image. Les 18 filtres superposés sont représentés sur la figure 6.9.

6.3 Harmoniques sphériques

Bien que donnant déjà des résultats satisfaisants, le GIST modifié, conserve un inconvénient majeur. Il repose sur le calcul de la transformée de Fourier de l'image plane et ne prend donc pas parfaitement en compte la structure sphérique de la représentation. Pour pouvoir transformer une image sphérique dans le domaine des fréquences, il faut s'orienter vers le calcul des harmoniques sphériques. Elles sont l'équivalent de la transformée de Fourier mais sur la surface de la sphère. Toutefois, il est à noter qu'il s'agit d'un équivalent et pas d'une adaptation. Les changements de base des deux transformations sont très différents.

6.3.1 Théorie

Tel que décrit dans [MacRobert, 1948], l'étude des harmoniques sphériques débute à la fin du 18ième siècle avec les travaux de Laplace sur la mécanique céleste. L'objectif était alors d'établir le

potentiel gravitationnel d'un point à partir de masses localisées à différents points de l'espace. Peu de temps avant, Legendre étudie une extension du potentiel newtonien et élabore les polynômes de Legendre. Ces polynômes sont un cas particulier des harmoniques sphériques. Ils ont été utilisés, en coordonnées sphériques, par Laplace dans son traité de *Mécanique Céleste* afin de représenter l'angle solide entre le point d'étude et les localisations des masses. Au milieu du 19ième siècle, Lord Kelvin et Peter Guthrie Tait introduisent les harmoniques sphériques solides comme solutions homogènes à l'équation de Laplace. Le terme « harmoniques sphériques » est introduit pour la première fois dans leur traité *Treatise on Natural Philosophy*. Parallèlement, durant le 19ième siècle, l'introduction des séries de Fourier a permis de résoudre de nombreux problèmes physiques dans le cadre rectangulaire. De nombreux aspects de la théorie des séries de Fourier pouvaient être généralisés par extension des harmoniques sphériques. Ceci représentait alors un avantage important pour les problèmes présentant une symétrie sphérique telle que la mécanique céleste. Les harmoniques sphériques étaient déjà alors très importantes pour la résolution des problèmes de physique. La mécanique quantique apparue au 20ième siècle a contribué à accroître encore cette importance. Elles sont utilisées dans ce domaine pour représenter les différentes configurations quantifiées des orbites atomiques.

La limite à l'utilisation de la transformée de Fourier est qu'elle n'est pas calculable sur la surface de la sphère. Il n'est pas possible de prendre en compte les fréquences constitutives de l'image parallèlement à la périodicité intrinsèque spatiale offerte par la sphère. Le principe de la transformée de Fourier est de projeter selon une base constituée de sinusoïdes de différentes fréquences. En ce qui concerne les harmoniques sphériques, il s'agit de projeter suivant les fonctions propres (*i.e.* eigenfunctions) de l'opérateur Laplacien sphérique. Ces fonctions sont les harmoniques sphériques $Y_l^m : S^2 \to \mathbb{C}$. La théorie sur les harmoniques sphériques exposée dans cette section est issue des articles [Bülow et Daniilidis, 2001], [Bülow, 2002], [Green, 2003] et [Schönefeld, 2005].

La sphère unitaire S^2 incluse dans $\mathbb{R}^3$ est paramétrée en utilisant les coordonnées sphériques. Un élément η de S^2 s'écrit :

$$\eta = \begin{bmatrix} cos(\theta)sin(\phi) \\ sin(\theta)sin(\phi) \\ cos(\phi) \end{bmatrix} \tag{6.46}$$

avec $\theta \in [0, 2\pi]$ l'angle de gisement et $\phi \in [0, \pi]$ l'angle d'élévation. Les harmoniques sphériques

constituent un système orthonormal complet de l'espace des fonctions quadratiques intégrables sur la sphère $L^2(S^2)$ telles que :

$$\int_{S^2} Y_l^m(\eta)\overline{Y_{l'}^{m'}(\eta)}d\eta = \delta_{mm'}\delta_{ll'} \tag{6.47}$$

$d\eta$ est une notation simplifiée remplaçant $d\eta = sin(\phi)d\phi d\theta$. $\overline{Y_{l'}^{m'}(\eta)}$ est le conjugué de $Y_{l'}^{m'}(\eta)$ et $\delta_{kk'}$ est le symbole de Kronecker tel que :

$$\delta_{kk'} = \begin{cases} 1 \text{ si } k = k' \\ 0 \text{ sinon} \end{cases} \tag{6.48}$$

Les harmoniques sphériques sont définies par :

$$Y_l^m(\eta) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^{|m|}\left(cos\left(\phi\right)\right)e^{jm\theta} \tag{6.49}$$

avec $l \in \mathbb{N}$ et $|m| \leq l$ où l est le numéro de bande correspondant à une fréquence et m est un paramètre d'orientation. P_l^m correspond aux polynômes de Legendre associés avec $x \in [-1, 1]$ et d'équation :

$$P_l^m(x) = \frac{(-1)^m(1-x^2)^{m/2}}{2^l l!}\frac{d^{l+m}}{dx^{l+m}}(x^2-1)^l \tag{6.50}$$

Les polynômes de Legendre associés sont définis pour $l \in \mathbb{N}$ et $m \in [\![0, l]\!]$. Les six premiers polynômes sont représentés dans la figure 6.10.

Toute fonction définie sur la sphère peut être décomposée en somme d'harmoniques sphériques. Il en résulte la combinaison linéaire suivante :

$$f = \sum_{l\in\mathbb{N}}\sum_{|m|\leq l} f_l^m Y_l^m \tag{6.51}$$

Les coefficients f_l^m sont obtenus à partir d'une fonction f comme suit :

$$f_l^m = \int_{\eta\in S^2} f(\eta)\overline{Y_l^m(\eta)}d\eta \tag{6.52}$$

Si $f_l^m = 0$ pour tout $l > L$, f est dite à bande limitée avec une largeur de bande L. L'ensemble des coefficients f_l^m est appelé la transformée de Fourier sphérique ou le spectre de f. Les cinq premières bandes d'harmoniques sphériques sont représentées dans la figure 6.11.

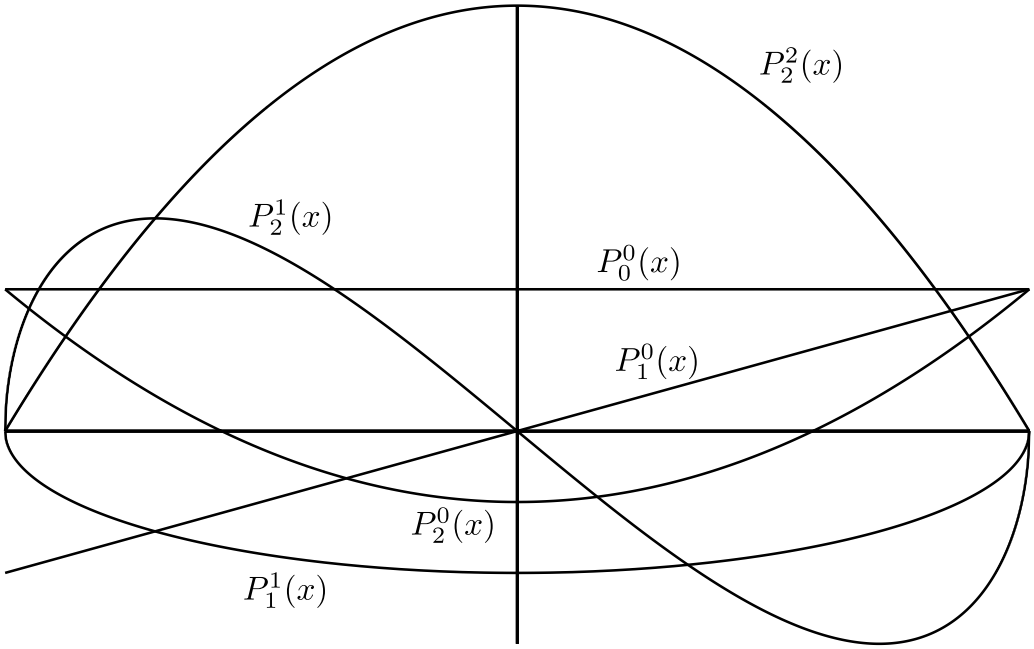


FIG. 6.10 – Les six premiers polynômes de Legendre associés

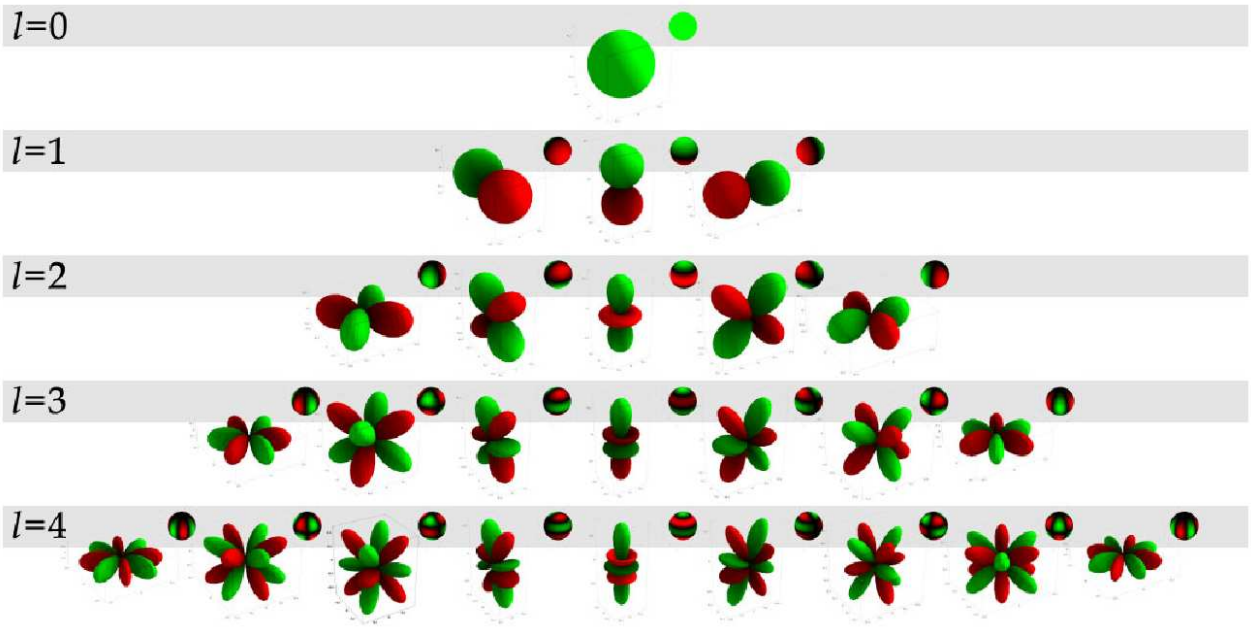


FIG. 6.11 – Les cinq premières bandes d'harmoniques sphériques sont présentées comme fonctions sphériques non signées à partir de l'origine et par couleur sur la sphère unité. Le vert correspond aux valeurs positives et le rouge aux valeurs négatives. Image extraite du tutoriel Spherical Harmonic Lighting : The Gritty Details de Robin Green [Green, 2003]

Étant donnée l'équation du calcul des harmoniques sphériques données en 6.49, le module du spectre est invariant à une rotation d'angle θ, soit une rotation autour de l'axe z. En effet, l'angle θ n'apparaît que dans l'exponentielle complexe dont le module vaut $|e^{jm\theta}| = 1$. Dans notre approche, bien qu'elle se veuille générale, nous n'avons testé que des cas de déplacements dans le plan d'un robot mobile. La seule rotation du robot utilisée est donc celle autour de l'axe z. Or, les harmoniques sphériques ont un module qui en est indépendant. Toutefois, cela ne reste valable que pour un déplacement parfaitement plan du robot. Si l'inclinaison du plan change, cela induit une modification de l'angle d'élévation. Si la variation d'angle d'élévation devient trop importante, il sera nécessaire d'ajouter une invariance du spectre à cette rotation.

Comme énoncé dans la section sur le GIST en 6.2.3.2, l'information de structure de l'environnement est contenue dans les fréquences de l'image acquise. Les harmoniques sphériques étant une description fréquentielle de l'image sphérique, le spectre est alors utilisé comme descripteur de structure de l'environnement. Ce dernier présente l'avantage d'être adapté à la représentation sphérique. De plus, comme expliqué précédemment et représenté sur la figure 6.11, l'information de fréquence est contenue dans le numéro de bande l et l'orientation est contenue dans le paramètre m. Plus la bande l est élevée et plus il s'agit d'une fréquence élevée. Pour chaque bande l, le paramètre m correspond à une orientation ; plus l est élevé et plus il y a d'orientations possibles. Du fait de cette paramétrisation, l'utilisation des filtres de Gabor n'est pas nécessaire. Le module des paramètres $|f_l^m|$, qui constituent le module du spectre de l'image sphérique, est directement utilisé comme descripteur. Ces coefficients sont stockés linéairement dans un vecteur et constituent le descripteur global de l'environnement.

Le nombre de bandes utilisé pour décrire l'image est un paramètre important qu'il faut prendre en compte. Dans le cas de la TFD 2D, la taille du spectre est définie par la taille de l'image. Dans le cas des harmoniques sphériques, rien ne détermine le nombre de bandes nécessaires. De plus, le nombre de coefficients suit une loi quadratique du nombre de bandes choisies. Dans le cas de la figure 6.11, $l = 5$ et nous avons donc $l^2 = 25$ coefficients. Choisir un trop grand nombre de bandes résulterait en un descripteur de taille déraisonnable. En conservant les notations de la section 6.2.5, la taille du descripteur est :

$$S_d = l^2 \tag{6.53}$$

Dans le cadre du calcul d'éclairage dans le domaine de l'infographie, trois bandes sont suffisantes du fait d'un paramètre d'atténuation exponentiel qui rend négligeable les bandes de plus grande fréquence [Green, 2003]. Dans notre cas, il n'existe aucun paramètre d'atténuation et il est difficile de déterminer combien de bandes seront nécessaires. Pour cela, nous nous basons sur les travaux de [Friedrich *et al.*, 2007] dans lesquels ils effectuent de la localisation à partir d'harmoniques sphériques. Les auteurs utilisent seulement les cinq premières bandes et parviennent à effectuer une localisation assez précise. Étant donné que nous cherchons une description globale de l'environnement, les cinq premières bandes devraient garantir une information suffisante. Le descripteur global sera donc de dimension $S_d = 25$.

6.3.2 Implémentation

Le calcul exact des harmoniques sphériques est problématique du fait de la présence d'une intégration comme le montre l'équation 6.52. Même si celle-ci donne lieu à une sommation lors du calcul discret, le temps de calcul est très élevé. Comme il existe l'algorithme Fast Fourier Transform (FFT) ou transformée de Fourier rapide pour la transformée de Fourier, il existe une méthode de calcul rapide des harmoniques sphériques basée sur l'intégration de Monte Carlo, des tables pré-calculées et les propriétés des polynômes de Legendre associés. Cette méthode est très utilisée dans le calcul de l'illumination dans les rendus temps-réel en infographie. Les mécanismes permettant une implémentation efficace du calcul des harmoniques sphériques sont détaillés dans [Green, 2003].

Le principe de l'intégration de Monte Carlo est très simple et permet la simplification de nombreux calculs. Son principe repose sur une approche statistique. Toute fonction utilisant une valeur aléatoire possède une valeur moyenne, appelée espérance, qui se calcule comme suit :

$$E[f(x)] = \int f(x)p(x)dx \tag{6.54}$$

où $p(x)$ est une densité de probabilité telle que :

$$\int_{-\infty}^{+\infty} p(x)dx = 1 \tag{6.55}$$

Un autre moyen de calculer la valeur moyenne de la fonction f est de prendre la moyenne d'un grand nombre d'échantillons aléatoires de la fonction. Lorsque le nombre d'échantillons tend vers l'infini, d'après la loi des grands nombres, ce calcul de moyenne tend vers l'espérance :

$$E[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{6.56}$$

L'intégration de Monte Carlo fait le lien entre les deux équations de calcul de l'espérance.

$$\int f(x)dx = \int \frac{f(x)}{p(x)}p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \tag{6.57}$$

En adoptant une notation de somme pondérée par $w(x) = \dfrac{1}{p(x)}$, l'équation de calcul de l'intégrale de f devient :

$$\int f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)w(x_i) \tag{6.58}$$

Si $p(x)$ est une distribution uniforme sur l'espace à échantillonner, il suffit de sommer les échantillons de la fonction $f(x_i)$, de diviser par le nombre d'échantillons et de multiplier par le poids w. Une bonne approximation du calcul de l'intégrale est ainsi obtenue. Dans notre cas, c'est la surface de la sphère unitaire qui est échantillonnée suivant une distribution uniforme. Le poids est donc $w = 4\pi$. L'intégration suivante est alors obtenue :

$$\int f(x)dx \approx \frac{4\pi}{N}\sum_{i=1}^{N} f(x_i) \tag{6.59}$$

Dans le cas du calcul des coefficients f_l^m, l'intégration de Monte Carlo devient :

$$f_l^m \approx \frac{4\pi}{N}\sum_{i=1}^{N} f(\eta_i)\overline{Y_l^m(\eta_i)} \tag{6.60}$$

Le problème suivant est le calcul des harmoniques sphériques Y_l^m du fait des multiples factorielles et des dérivées d'ordre important dans le calcul des polynômes de Legendre associés. En ce qui concerne les factorielles, une table de factorielles pré-calculée et chargée au début du programme permet de considérablement réduire le temps d'obtention de la valeur de la factorielle. Le calcul factoriel se résume ainsi à un simple accès dans un tableau. En ce qui concerne les dérivées d'ordre important, les propriétés de calcul incrémental des polynômes de Legendre associés sont utilisées pour réduire le temps de calcul. Ces propriétés se résument en trois règles de calcul :

1.
$$(l-m)P_l^m(x) = x(2l-1)P_{l-1}^m(x) - (l+m-1)P_{l-2}^m(x) \tag{6.61}$$
Cette première équation permet de calculer à partir des deux bandes précédentes $l-1$ et $l-2$ une bande de rang supérieur l.

2.
$$P_m^m(x) = (-1)^m(2m-1)!!(1-x^2)^{m/2} \tag{6.62}$$
Cette deuxième équation est d'importance capitale car c'est la seule qui ne nécessite pas de valeurs précédentes pour être calculée. Elle permet également d'initialiser le calcul avec $P_0^0(x) = 1$.

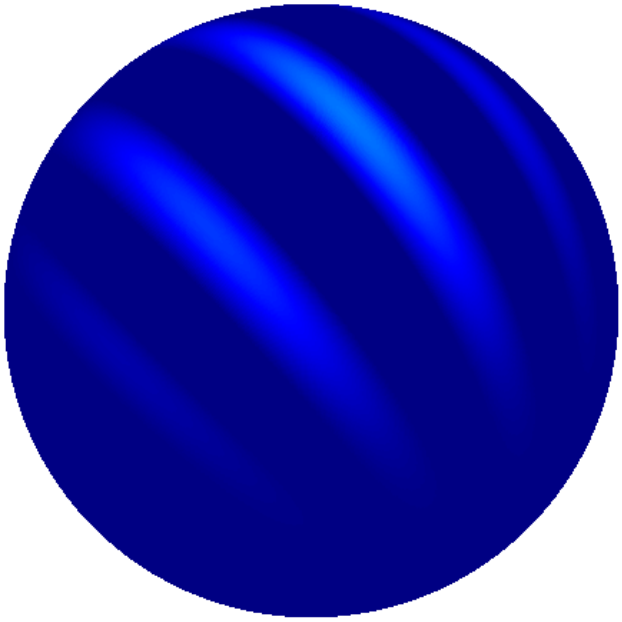
3.
$$P_{m+1}^m(x) = x(2m+1)P_m^m(x) \tag{6.63}$$
Cette troisième équation permet de calculer la valeur d'un terme dans une bande supérieure.

L'ensemble de toutes ces règles de calcul permet de déterminer efficacement, et avec un temps de calcul assez faible, le spectre sphérique (ou transformée de Fourier sphérique) d'une fonction f. Dans notre cas d'étude, la fonction définie sur la sphère est directement l'image sphérique. Les valeurs des $f(x_i)$ sont donc les intensités des pixels de l'image. Un exemple de module de spectre, avec $l = 70$, obtenu à partir d'une image est montré sur la figure 6.12(b). L'image d'origine est un filtre de Gabor sphérique d'orientation $\theta = \frac{2\pi}{3}$ (*cf.* figure 6.12(a)). Il s'agit d'un spectre à titre d'exemple et n'est aucunement représentatif des spectres des images traitées. Les pixels blancs correspondent aux valeurs les plus élevées tandis les pixels noirs sont les valeurs les plus faibles. Dans ce cas les coefficients ne sont que des valeurs positives.

6.4 Conclusion

Cette section constitue un rappel des deux descripteurs de structure utilisés :

– Le descripteur GIST est obtenu à partir d'une version modifiée du GIST développé par [Oliva et Torralba, 2001]. La modification est une adaptation à la représentation sphérique utilisée dans le cadre de cette thèse. Le procédé consiste à calculer la transformée de Fourier de l'image



(a) Filtre de Gabor sphérique.

(b) Spectre du filtre de Gabor sphérique.

FIG. 6.12 – (a) Représentation du filtre de Gabor sphérique d'orientation $\theta = \frac{2\pi}{3}$. (b) Module du spectre du filtre de Gabor sphérique d'angle $\theta = \frac{2\pi}{3}$. Les bandes sont représentées pour $|m| \leq l$ et $l \in [\![0, 70]\!]$.

sphérique. Ensuite, une banque de filtres de Gabor permet d'extraire les fréquences et orientations principales de l'image à partir de son spectre. Le résultat global de la réponse de chaque filtre est concaténée dans un vecteur constituant le descripteur de structure de l'environnement.

– Le descripteur à base d'harmoniques sphériques est simplement la concaténation dans un vecteur des valeurs du spectre de l'image sphérique. Le spectre est celui obtenu par passage dans le domaine fréquentiel de l'image sphérique par l'intermédiaire des harmoniques sphériques.

Les deux descripteurs obtenus pour une même image sont illustrés sur la figure 6.13. Les deux méthodes donnent des résultats très différents. Les avantages et inconvénients de chacune des deux méthodes sont résumés dans le tableau 6.1.

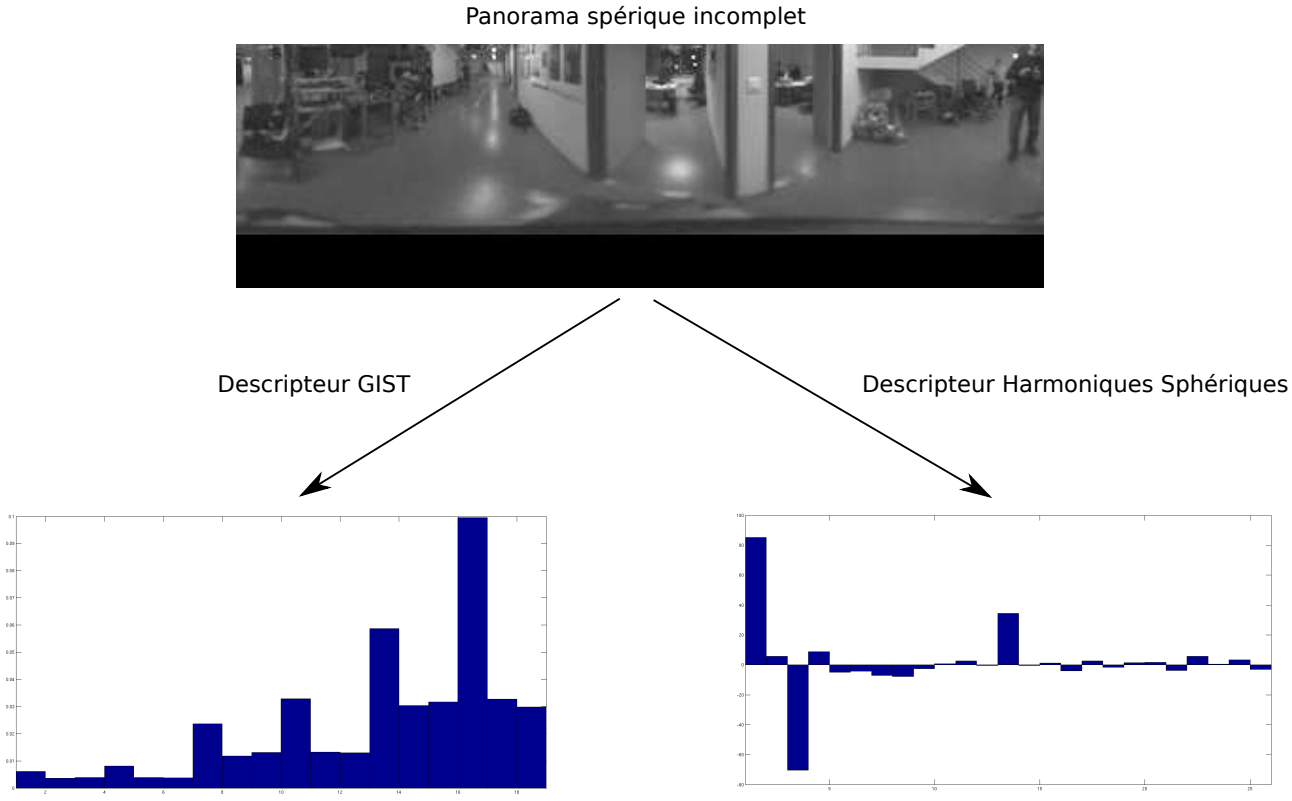


FIG. 6.13 – Comparaison des descripteurs de structure de l'environnement obtenus avec les deux méthodes présentées dans ce chapitre.

	Avantages	Inconvénients
Descripteur GIST	– Descripteur de structure efficace – Très faible dimension – Temps de calcul	– Invariance seulement à la rotation d'axe z – Prise en compte uniquement de la périodicité spatiale suivant l'angle de gisement
Descripteur Harmoniques Sphériques	– Descripteur de structure efficace – Très faible dimension – Temps de calcul – Robustesse aux changements d'illumination (modèle d'illumination affine, *cf.* section 7.4) – Adaptation complète à la représentation sphérique – Utilisation directe du spectre (pas de filtrage)	– Invariance seulement à la rotation d'axe z – Détermination empirique du nombre de bandes nécessaires

TAB. 6.1 – Comparaison des deux descripteurs de structure de l'environnement.

Chapitre 7

Segmentation en ligne de séquences d'images

Sommaire

7.1 Principes de la détection de changement de modèle

Pour chaque nouvelle image, le descripteur global de structure, GIST ou harmoniques sphériques, est calculé. Il en découle alors un signal évoluant au cours du temps et en fonction du déplacement du robot. Afin de détecter les changements de structure de l'environnement, il faut pouvoir détecter les changements significatifs dans ce signal multidimensionnel (la taille du descripteur plus la dimension temps : $S_d + 1$). Étant donné qu'il s'agit d'un déplacement d'un lieu à un autre, les transitions ne sont pas brutales. Il s'agit d'un signal évoluant lentement. Les zones de transitions entre deux lieux seront donc constituées d'un mélange des structures des deux environnements. La longueur des zones appartenant à un lieu donné et les longueurs des zones de transition sont variables et difficilement détectables. Des méthodes robustes de détection de changement de modèle sont alors utilisées pour traiter efficacement le signal. Ces méthodes permettent de déterminer la séparation idéale entre deux ensembles.

Avant de présenter les méthodes principales de détection de changement de modèle, quelques éléments de statistique nécessaires à la compréhension de la théorie sont présentés. Le principe fondamental est le test d'hypothèses concernant les paramètres du processus. H_0 est l'hypothèse nulle dénotant une situation normale tandis H_1 est l'hypothèse alternative dénotant une situation anormale ou alternative. Il s'agit donc de détecter à quel moment le processus vérifie l'hypothèse H_0, l'hypothèse H_1 et par conséquent le changement d'hypothèse. Soit T la statistique de test calculée à partir des données :

$$T(y) = f(y_1, ..., y_N) \tag{7.1}$$

avec y_i les différents échantillons et f la fonction permettant de calculer la statistique de test. y représente la séquence des échantillons tel que $y = [y_1, ..., y_N]$. $d\,(T\,(y)) \in \{0, 1\}$ est la fonction de décision déterminant si la statistique de test est dans un intervalle convenable, c'est-à-dire que l'hypothèse H_0 est vérifiée. 0 est la situation normale et 1 est une alarme indiquant un changement. d permet donc de classifier les valeurs de y. La fonction de décision est basée sur l'utilisation de valeurs critiques dénommées seuils bas T_L et haut T_H tel que :

$$d(T) = \begin{cases} 1 \text{ si } T < T_L \text{ ou } T > T_H \\ \\ 0 \text{ sinon} \end{cases} \tag{7.2}$$

La fonction de décision peut aussi être plus informative en ayant un résultat dans l'intervalle $d(T) \in [0, 1]$. Comme tout système de décision, l'issue de la fonction de décision d n'est pas forcément valide, elle peut donner lieu à une alarme en situation normale ou alors ne pas lever d'alarme alors que la situation change. Le premier cas est appelé fausse alarme ou faux positif tandis que le second est un faux négatif. Ces situations sont résumées dans le tableau suivant :

	Situation normale	Alarme
H_0 est vraie	OK	faux positif
H_1 est vraie	faux négatif	OK

TAB. 7.1 – Issues de la fonction de décision d à partir de la statistique de test T

L'ensemble des données y pour lesquelles l'hypothèse H_0 est rejetée constitue la région critique dénotée C. Les notations suivantes permettent de définir les probabilités de faux positifs et de faux négatifs par rapport à cette région C :

$$\begin{aligned} P(\text{faux positif}) &= \alpha && (7.3)\\ &= P(y \in C|H_0) && (7.4)\\ P(\text{faux négatif}) &= \beta && (7.5)\\ &= P(y \notin C|H_1) && (7.6) \end{aligned}$$

Soit α_ϕ, respectivement β_ϕ, la probabilité d'obtenir un faux positif, respectivement un faux négatif, en considérant le test ϕ. L'objectif est de choisir un test ϕ qui minimise β_ϕ et en ayant pour contrainte un α_ϕ faible, idéalement nul. Il s'agit de minimiser les deux grandeurs mais il existe une contrainte beaucoup plus forte sur α_ϕ car les faux positifs sont critiques dans les tests d'hypothèses tandis que les faux négatifs sont tolérables. De plus, réduire β_ϕ entraîne généralement une augmentation de α_ϕ. Ainsi, il ne s'agit pas de déterminer le minimum du couple $(\alpha_\phi, \beta_\phi)$ mais plutôt d'imposer une valeur faible pour α_ϕ et ensuite de minimiser β_ϕ en fonction de cette contrainte. La puissance d'un test permet d'exprimer efficacement cet objectif :

$$\pi_\phi(\theta) = \begin{cases} \alpha_0 \text{ si } \theta \in H_0 \\ 1 - \beta_0 \text{ si } \theta \in H_1 \end{cases} \quad (7.7)$$

θ représente l'ensemble des paramètres caractérisant une hypothèse. Dans le cas où l'hypothèse est une distribution de probabilité, θ représente les paramètres de la distribution. La puissance du test

idéal est définie telle que :

$$\pi_\phi(\theta) = \begin{cases} 0 \text{ si } \theta \in H_0 \\ 1 \text{ si } \theta \in H_1 \end{cases} \tag{7.8}$$

Pour pouvoir utiliser la statistique de test, il est nécessaire de transformer les données sous forme de grandeurs statistiques. Pour cela, la fonction de vraisemblance permet de connaître la densité de probabilité des données (si la distribution est connue) :

$$\mathcal{L}_\theta(y) = P(y|\theta) \tag{7.9}$$

Dans le cas d'une distribution normale, à titre d'exemple, avec $\theta = (\mu, \sigma^2)$, la fonction de vraisemblance s'écrit :

$$\mathcal{L}_\theta(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2}} \tag{7.10}$$

Dans ce cas, H_0 est spécifiée en fonction des paramètres μ et σ.
Pour rappel, y représente la séquence des échantillons tel que $y = [y_1, ..., y_N]$. Si les échantillons sont indépendamment et identiquement distribués (i.i.d.), la fonction de vraisemblance peut s'écrire :

$$\mathcal{L}_\theta(y_1, y_2, ..., y_N) = \mathcal{L}_\theta(y_1)\mathcal{L}_\theta(y_2)...\mathcal{L}_\theta(y_N) \tag{7.11}$$

Le ratio de vraisemblance permet d'obtenir la vraisemblance relative des données. Il s'agit alors d'indiquer si les données appartiennent plutôt à une distribution qu'à une autre. θ_0 et θ_1 correspondent respectivement à H_0 et H_1. Le ratio de vraisemblance logarithmique (*log likelihood ratio* en anglais) est généralement utilisé :

$$s_i = ln\left(\frac{\mathcal{L}_{\theta_1}(y_i)}{\mathcal{L}_{\theta_0}(y_i)}\right) \tag{7.12}$$

L'interprétation du ratio de vraisemblance logarithmique est le suivant :

$$E(s_i) \begin{cases} < 0 \text{ si } \theta \in H_0 \\ > 0 \text{ si } \theta \in H_1 \\ \approx 0 \text{ si } H_0 \approx H_1 \end{cases} \tag{7.13}$$

où $E(x)$ est l'espérance mathématique.

La dernière partie de ce rappel concerne la puissance du test telle que définie précédemment. Le test le plus puissant est le test qui satisfait au mieux les conditions idéales du test. ϕ^* est le test le plus puissant si et seulement si pour tout ϕ tel que $\pi_\phi(\theta_0) \leq \pi_{\phi^*}(\theta_0)$ alors $\pi_{\phi^*}(\theta_1) \geq \pi_\phi(\theta_1)$. D'après le lemme de Neyman-Pearson [Neyman et Pearson, 1933], ϕ^* est le test le plus puissant s'il est construit de la manière suivante :

$$\exists\ h \text{ tel que } y \in C \text{ si et seulement si } \phi^* = \frac{\mathcal{L}_{\theta_1}(y)}{\mathcal{L}_{\theta_0}(y)} > h \tag{7.14}$$

C est la région critique regroupant l'ensemble des données y rejetant l'hypothèse H_0. h est le seuil de décision associé au test et permettant de définir à partir de quelle valeur l'hypothèse H_0 est rejetée. En adoptant les notations précédentes, la fonction de décision associée au test le plus puissant s'écrit :

$$d(\phi^*) = \begin{cases} 1 \text{ si } \phi^* > h \\ \\ 0 \text{ sinon} \end{cases} \tag{7.15}$$

Il s'agit d'un récapitulatif assez succinct qui permet de regrouper les éléments fondamentaux nécessaires aux explications des méthodes de détection de rupture de modèle. Tous ces éléments seront abordés à nouveau dans la présentation des différentes méthodes et notamment dans la présentation des algorithmes.

7.1.1 Principe des méthodes hors ligne

Les méthodes hors-ligne reposent sur la disponibilité de toutes les données acquises lors de l'expérimentation préalable. Il est alors possible de traiter toute l'information afin d'optimiser la segmentation. Le problème consiste à identifier des intervalles stationnaires. C'est-à-dire effectuer un partitionnement contraint avec des partitions homogènes. Les ruptures seront donc telles que la variance à l'intérieur d'une partition est inférieure à la variance entre les partitions.

La détection des ruptures de modèles sur des données hors-ligne est due aux travaux de Fisher [Fisher, 1958]. Détecter k ruptures de modèles revient à faire du k-partitionnement.

$$P = (P_1, P_2, ..., P_k) \tag{7.16}$$

$$1 \leq P_1 < P_2 < ... < P_k < N \tag{7.17}$$

N est l'indice de la dernière observation. P_i représente l'indice de l'observation définissant une rupture entre deux partitions. Afin de déterminer ces partitions, la somme des carrés ajustée au sein d'une partition est définie :

$$A_{SQ}[m..n] = \sum_{j=m}^{n} (y_i - \overline{y}[m..n])^2 \tag{7.18}$$

où $1 < m < n < N$ et $\overline{y}[m..n]$ est la moyenne d'un ensemble d'observations :

$$\overline{y}[m..n] = \frac{y_m + ... + y_n}{n - m + 1} \tag{7.19}$$

La somme de carrés ajustée permet d'exprimer le degré d'homogénéité d'une partition et correspond à un facteur près au calcul de l'écart type. À partir de cette équation, la figure de mérite des ruptures est définie par :

$$D_P = A_{SQ}[P_k..N] + \sum_{j=1}^{k-1} A_{SQ}[P_j..P_{j+1} - 1] \tag{7.20}$$

où $A_{SQ}[P_k..N]$ représente la somme des carrés ajustée pour l'ensemble des observations comprises entre le dernier élément de la partition k d'indice P_k et N. $A_{SQ}[P_j..P_{j+1} - 1]$ représente la somme des carrés ajustée pour l'ensemble des observations comprises entre le dernier élément de la partition j d'indice P_j et le dernier élément de la partition $j + 1$ d'indice P_{j+1} ; il s'agit de l'ensemble des éléments de la partition P_{j+1}. P est un partitionnement optimal en k partitions s'il n'y a pas de k-partitionnement P' tel que $D_{P'} < D_P$. D_P est toujours positif et si P est un N-partitionnement alors $D_P = 0$. Le N-partitionnement correspond au fait que chacune des observations appartient à une partition différente, résultant ainsi en autant de partitions que d'observations. Afin d'obtenir le meilleur partitionnement, il faut obtenir un compromis sur le nombres de partitions k. Ce nombre doit être suffisamment grand pour trouver toutes les ruptures et suffisamment petit pour ne pas avoir de détection de ruptures où il n'en existe pas. La complexité computationnelle pour trouver le k-partitionnement optimal est C_N^k.

Il existe de nombreux algorithmes de partitionnement optimaux ou non qui permettent d'effectuer ce travail (*k-means*, *k-medians*, *k-medoids*, *silhouette clustering*). Cette partie n'est pas plus développée étant donné que l'approche développée dans la thèse repose sur une approche en ligne. Toutefois, cette section est utile dans la mesure où elle présente les principes de base du partitionnement. Elle est aussi

utile du fait que l'approche utilise des concepts de méthodes en ligne mais dérive d'une méthode hors ligne.

7.1.2 Principe des méthodes en ligne

Les méthodes en ligne reposent sur des données présentées séquentiellement. Il est impossible de disposer de toute l'information au moment du traitement. Il s'agit alors de détecter quand les paramètres du processus varient. Dans cette approche, il est important de comprendre la différence entre l'instant noté t_a qui est l'instant auquel l'alarme est levée et l'instant noté t_c qui est l'instant exact du changement de modèle. Dans tous les cas nous avons $t_c \leq t_a$ et idéalement $t_c = t_a$. Il est impossible de retrouver t_c à partir de t_a car il n'existe pas de relation entre les deux instants. Le temps séparant les deux instants est fonction de l'évolution des observations.

Le problème est formalisé comme suit. Soit p la distribution actuelle, p_{θ_0} la distribution sous l'hypothèse H_0 et p_{θ_1} la distribution sous l'hypothèse H_1. En considérant y_k l'observation à un instant k comme appartenant à H_0 alors nous avons :

$$H_0 : p(y_k|y_{k-1}, ..., y_1) = p_{\theta_0}(y_k|y_{k-1}, ..., y_1) \tag{7.21}$$

Si les deux hypothèses H_0 et H_1 sont présentes dans l'ensemble de données jusqu'à l'instant k, il existe un instant t_c tel que :

$$1 \leq i \leq t_c - 1 \quad : \quad p(y_i|y_{i-1}, ..., y_1) = p_{\theta_0}(y_i|y_{i-1}, ..., y_1) \tag{7.22}$$

$$t_c \leq i \leq k \quad : \quad p(y_i|y_{i-1}, ..., y_{t_c}) = p_{\theta_1}(y_i|y_{i-1}, ..., y_{t_c}) \tag{7.23}$$

L'alarme t_a est définie comme le plus petit instant k permettant de choisir H_1 plutôt que H_0. Comme présenté dans les rappels sur les statistiques de détection de changement de modèle, une statistique de test est utilisée. Dans le cas des méthodes en ligne, la statistique de test est la somme des ratios de vraisemblance logarithmiques définie par :

$$S_i^k = \sum_{j=i}^{k} s_j = \sum_{j=i}^{k} ln\left(\frac{L_{\theta_1}(y_j)}{L_{\theta_0}(y_j)}\right) \tag{7.24}$$

Pour résoudre ce problème, il existe trois méthodes communes :

1. Les diagrammes de Shewhart [Shewart, 1931].
2. La moyenne géométrique glissante [Roberts, 1959].
3. Le CUSUM [Page, 1954].

7.1.3 Diagrammes de Shewhart

Le principe des diagrammes de Shewhart repose sur des décisions indépendantes prises sur des sous-ensembles d'échantillons de taille N. La prise de décision se fait donc tous les N échantillons et il n'y a pas de détection intermédiaire. Soit k l'indice du sous-ensemble courant, la détection se fait suivant le principe de la somme des ratios de vraisemblance logarithmiques :

$$S_{(k-1)N+1}^{kN} > h \tag{7.25}$$

h est le seuil de décision tel que défini dans les rappels sur les détections de changement de modèle. Dans le cas d'une détection, une alarme est levée indiquant un changement d'hypothèse. La granularité de cette approche est déterminée par la taille du sous-ensemble d'observations N. Un élément important à déterminer est la valeur du seuil de décision h. Il est nécessaire d'avoir un délai court si il existe une rupture de modèle mais un temps long avant une alarme s'il n'y a pas de rupture. Le critère communément utilisé est le temps d'exécution moyen (Average Run Lenght en anglais *i.e* ARL) qui permet de déterminer le nombre d'observations moyen avant qu'une alarme soit levée. Pour déterminer l'ARL, deux valeurs sont définies :

- L'ARL_0 qui est le temps avant une alarme s'il n'y a pas eu de rupture.
- L'ARL_1 qui est le délai de l'alarme, c'est-à-dire le temps avant qu'une alarme soit levée depuis la dernière alarme.

Pour ces deux valeurs, la grandeur α_0 est définie telle que :

$$\alpha_0 = P(S_1^N > h | H_0) \tag{7.26}$$

Cette équation permet d'obtenir la probabilité d'une alarme à l'instant kN [Shewart, 1931] :

$$P(t_a = kN | H_0) = (1 - \alpha_0)^{k-1} \alpha_0 \tag{7.27}$$

Les espérances d'ARL_0 et d'ARL_1 sont obtenues par les formules suivantes :

$$E(ARL_0) = \frac{N}{\alpha_0} \tag{7.28}$$

$$E(ARL_1) = \frac{N}{P(S_1^N > h|H_1)} \tag{7.29}$$

Le seuil h est déterminé en fonction de la valeur de l'ARL_0 désiré. La méthode repose soit sur une estimation à partir d'un modèle des distributions de probabilité des observations soit sur une estimation empirique.

7.1.4 Moyenne géométrique glissante

La méthode de la moyenne géométrique glissante présente l'avantage d'accorder plus d'importance aux observations les plus récentes tout en conservant l'influence des observations les plus anciennes. Il ne s'agit pas d'un calcul sur un sous-ensemble comme la méthode des diagrammes de Shewhart. Toutefois, les observations les plus anciennes ont une influence pour ainsi dire négligeable suivant la paramétrisation du calcul de la moyenne. Soit le paramètre d'oubli exponentiel γ tel que $0 < \gamma < 1$. Le calcul de la moyenne géométrique glissante s'obtient par l'équation suivante :

$$g_k = (1-\gamma)g_{k-1} + \gamma s_k \tag{7.30}$$

$$= \gamma \sum_{n=0}^{k} (1-\gamma)^n s_{k-n} \tag{7.31}$$

L'instant de l'alarme t_a est déterminé par l'instant k où $g_k \geq h$. Pour déterminer la valeur du seuil h, il faut prendre en compte que g_k suit une loi normale suivant l'hypothèse H_0. h représente alors l'écart à la moyenne limite avant de privilégier l'hypothèse H_1. Comme la méthode des diagrammes de Shewhart, la moyenne géométrique glissante se calcule en temps constant mais présente l'avantage supplémentaire d'être calculable de manière incrémentale. Elle prend en compte toutes les observations via un système de pondération impliquant un oubli des valeurs les plus anciennes.

7.1.5 Méthode du CUSUM

La dernière méthode est le CUSUM signifiant CUmulative SUM. Le nom complet de la méthode est le diagramme de contrôle de la somme cumulée. Le problème des méthodes précédentes est que S_i^k présente une dérive vers les valeurs négatives sous l'hypothèse H_0. Il en résulte que l'ARL_1 peut

devenir beaucoup plus long que nécessaire. La solution apportée par cette approche est d'ajuster S_i^k pour qu'il ne devienne pas trop petit. Le calcul devient alors :

$$g_k = S_1^k - m_k \tag{7.32}$$

avec

$$m_k = \min_{1 \leq j \leq k} (S_1^j) \tag{7.33}$$

L'instant d'alarme est alors obtenu par $t_a = min(k|g_k \geq h)$, ou encore $t_a = min(k|S_1^k \geq m_k + h)$.

Il existe de nombreuses autres méthodes pour la détection de changement de modèle plus ou moins complexes et résolvant plus ou moins bien les problèmes des approches classiques. Parmi ces différents problèmes, nous trouvons l'estimation des densités de probabilité, la dérive de l'estimation vers les valeurs négatives, la méthode de l'ARL qui est très controversée (de nombreux travaux estiment qu'il ne s'agit pas d'une mesure significative) ou encore la prise en compte des données multidimensionnelles. Ces méthodes ne sont pas développées étant donné que les méthodes classiques sont utilisées pour élaborer l'approche développée.

7.2 Première application à la détection de changement de lieu : Preuve de concept

Le descripteur GIST modifié est utilisé dans la première approche développée. Il s'agit de faire une preuve de concept sur la détection de changement de lieu basée sur une information structurelle globale représentée par un descripteur de très faible dimension.

7.2.1 Hypothèses et simplification de l'équation de détection de changement de lieu

Soit y_i la valeur du descripteur GIST modifié à chaque instant i. Pour rappel, y_i est un vecteur puisque le signal est multidimensionnel. En reprenant les notations précédemment introduites, la série d'observations se note $y_1, y_2, ..., y_i, ..., y_n$ avec n le dernier instant connu. L'objectif est de déterminer s'il existe une rupture de modèle au sein de ces observations. Nous émettons alors l'hypothèse que les observations jusqu'à l'instant t_c suivent l'hypothèse H_0 et possèdent une distribution de probabilité f_0 tandis que les observations comprises entre $t_c + 1$ et n suivent l'hypothèse H_1 et possèdent une

distribution de probabilité f_1. Il s'agit d'une hypothèse générale dont il s'agit de vérifier la validité. L'appartenance des observations à une distribution ou l'autre est testée en utilisant le ratio de vraisemblance logarithmique tel que défini précédemment. Dans ce cas, la fonction de vraisemblance se note $\mathcal{L}_\theta(y) = f(y|\theta)$. Le ratio s'exprime alors :

$$s_i = ln\left(\frac{f_1\left(y_i|\theta_1\right)}{f_0\left(y_i|\theta_0\right)}\right) \tag{7.34}$$

D'après le lemme de Neyman-Pearson, le test le plus puissant ϕ^* est satisfait en définissant la statistique de test comme la somme des ratios de vraisemblance logarithmiques.

$$S_\tau^n = \sum_{j=\tau}^{n} s_j \tag{7.35}$$

$$= \sum_{j=\tau}^{n} ln\left(\frac{f_1\left(y_j|\theta_1\right)}{f_0\left(y_j|\theta_0\right)}\right) \tag{7.36}$$

Le test $S_\tau^n > h$ permet de déterminer si les observations à partir de t_c appartiennent effectivement à l'hypothèse H_1 ou non. L'équation précédente permet d'établir le test d'hypothèse suivant :

$$t_c = min\{n : \arg\max_\tau(S_\tau^n > h, 0 \leq \tau \leq n)\} \tag{7.37}$$

Si $t_c = n$ alors il n'y a pas de changement de modèle et toutes les observations appartiennent à l'hypothèse H_0. Par contre, si $t_c = \tau$ alors t_c est l'instant de rupture de modèle présentant le maximum de dissimilarité entre les hypothèses H_0 et H_1. Cette approche correspond à une approche hors-ligne et permet de satisfaire le meilleur partitionnement qui soit connaissant les n observations disponibles. Il n'est toutefois pas possible de prédire sur les observations à venir. Le partitionnement avec plus d'observations ne sera alors plus nécessairement idéal. Pour obtenir un meilleur partitionnement, une ré-estimation complète des ruptures de modèles doit être effectuée. Le seuil h, comme abordé dans les rappels de statistique, est fortement lié à la mesure ARL. Il peut aussi être considéré comme un réglage de la précision de la détection : plus h sera faible, respectivement élevé, et plus, respectivement moins, de ruptures seront détectées. Il est alors nécessaire de trouver un bon compromis entre le fait de détecter un maximum de ruptures tout en ne détectant pas des ruptures inexistantes. Il s'agit de satisfaire une puissance de test optimale en ayant peu ou pas du tout de faux positifs et faux négatifs. Pour un maximum de cohérence, il doit n'y avoir aucun faux positif quitte à ne pas détecter certaines

ruptures (taux de faux négatifs acceptable).

Un point critique de cette approche est que toutes les hypothèses de ruptures de modèles sont testées, *i.e.* chaque échantillon étant une hypothèse de rupture. Le temps d'estimation de la rupture de modèle optimale est alors élevé. Lorsque le nombre d'échantillons croît, il devient rapidement impossible d'effectuer l'algorithme en temps-réel. De plus, les densités de probabilité f_0 et f_1 n'étant pas connues, il est nécessaire de les estimer à chaque nouvel échantillon et pour chaque hypothèse. Ceci induit un temps de calcul supplémentaire non négligeable. La méthode s'exécute donc hors-ligne du fait du temps de calcul mais fonctionne sur une présentation séquentielle des observations (principe de fonctionnement des méthodes en-ligne). L'avantage majeur de la méthode est de permettre de connaître exactement l'instant de rupture t_c.

Une approche en-ligne telle que le CUSUM permet de pallier le problème du temps de calcul. En apportant de légères modifications à l'approche précédente, l'équation incrémentale d'estimation du CUSUM s'écrit :

$$D_n = max \left\{ 0 : D_{n-1} + ln \left(\frac{f_1 \left(y_n | \theta_1 \right)}{f_0 \left(y_n | \theta_0 \right)} \right) \right\} \text{ avec } D_0 = 0 \tag{7.38}$$

Pour savoir si un changement de modèle est présent, il suffit de comparer la valeur du CUSUM à un seuil h tel que $D_n > h$. L'inconvénient de cette approche est qu'elle permet d'obtenir l'instant de l'alarme t_a mais il est impossible de retrouver l'instant exact de la rupture t_c. Ceci peut convenir pour certaines approches où un retard est toléré. Dans le cas où nous cherchons des modifications dans la structure de l'environnement, le retard n'est pas toléré au risque de trouver des changements de lieu décalés par rapport à leur position réelle. La segmentation de l'environnement ne serait alors pas cohérente.

L'algorithme développé suit donc une approche différente pour créer un algorithme en-ligne à partir du lemme de Neyman-Pearson. Il est d'abord nécessaire de faire des hypothèses sur la nature des signaux. Un exemple du type de signaux obtenus à partir de l'évolution du GIST modifié est affiché sur la figure 7.1. L'exemple présente l'évolution de 3 des 18 dimensions du GIST modifié.

Une première hypothèse raisonnable est de supposer que les signaux suivent une distribution gaussienne. Les signaux sont donc définis par une valeur moyenne correspondant à la grandeur caractérisant

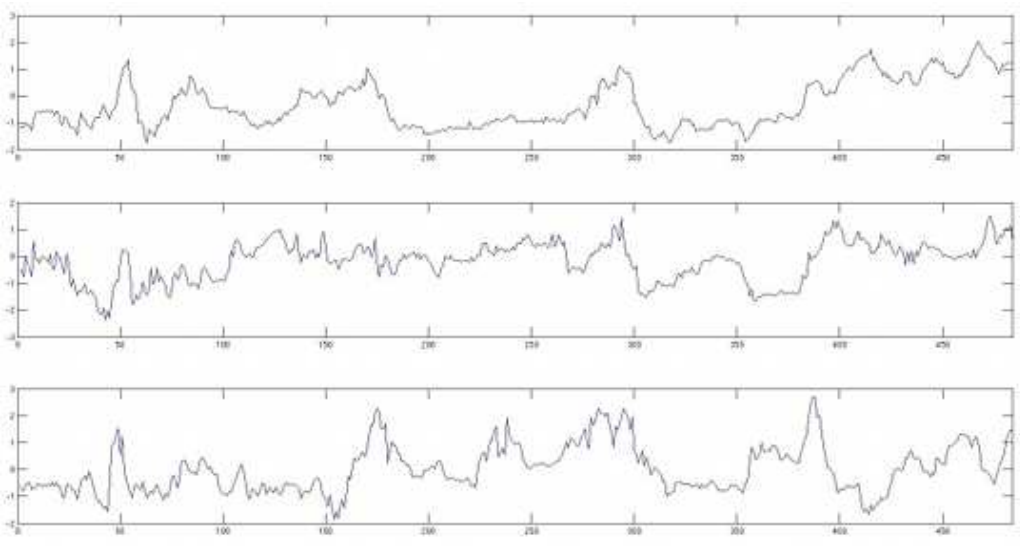


FIG. 7.1 – Le premier signal correspond à une haute fréquence d'orientation $\theta = 0°$. Le signal central est un signal de fréquence moyenne suivant l'orientation $\theta = 0°$. Le dernier signal est de basse fréquence et d'orientation $\theta = 30°$.

l'environnement à un instant donné et un écart-type caractérisant un bruit additif gaussien. Il s'agit d'un modèle de signal communément utilisé.

L'hypothèse d'indépendance entre les dimensions du signal est adoptée dans cette première approche. Il s'agit d'une hypothèse forte mais elle permet des simplifications intéressantes de l'équation de détection de rupture de modèle. La prise en compte d'un signal multidimensionnel n'est pas simple, il est plus aisé de faire des tests statistiques sur des signaux unidimensionnels. Dans le cas multidimensionnel, il faut soit ramener la statistique de test à une grandeur prenant en compte les différentes dimensions, méthode suivie pour la deuxième approche (*cf.* section 7.3), soit il faut pouvoir établir des tests significatifs suivant chacune des dimensions (méthode adoptée pour cette première approche). Dans ce deuxième cas, la prise en compte de la corrélation entre les dimensions du signal implique d'avoir une loi régissant les différents seuils h. Un pré-traitement des signaux est donc ajouté afin d'assurer une pseudo-indépendance et de garantir l'équivalence de décision d'un seuil h sur les différentes dimensions. Ce pré-traitement consiste simplement en une normalisation en ligne suivant chacune des dimensions. En considérant chaque dimension comme un signal indépendant, la normalisation du signal est effectuée en soustrayant la moyenne globale et en divisant par l'écart-type global. La distribution du signal devient donc une gaussienne centrée réduite $\mathcal{N}(0, 1)$. La moyenne globale et l'écart-type global sont calculés en-ligne de manière incrémentale afin de ne pas impacter le temps de calcul de l'algo-

rithme. L'inconvénient de cette méthode est qu'elle entraîne une instabilité de la réponse au début du traitement puisqu'il y a trop peu d'observations pour obtenir une estimation stable. Toutefois, cette instabilité disparaît rapidement et il suffit de ne pas prendre en compte les éventuelles ruptures de modèle détectées au tout début de l'expérimentation.

Les signaux (exemples sur la figure 7.1) présentent une allure stable avec un bruit de mesure. Ceci confirme alors la première hypothèse. Les variations importantes correspondent à des variations majeures de la structure de l'environnement et donc aux instants de rupture de modèle.

En prenant en compte les hypothèses précédentes, les fonctions de vraisemblance des hypothèses H_0 et H_1 suivent des lois normales. Leur paramétrisation est donc $\theta_i = (\mu_i, \sigma_i)$. Les fonctions de vraisemblance s'écrivent alors :

$$\begin{aligned} \mathcal{L}_{\theta_0}(y) &= f(y|\theta_0) && (7.39)\\ &= \frac{1}{\sqrt{2\pi\sigma_0}} e^{-\frac{1}{2}\frac{(y-\mu_0)^2}{\sigma_0^2}} && (7.40)\\ \mathcal{L}_{\theta_1}(y) &= f(y|\theta_1) && (7.41)\\ &= \frac{1}{\sqrt{2\pi\sigma_1}} e^{-\frac{1}{2}\frac{(y-\mu_1)^2}{\sigma_1^2}} && (7.42) \end{aligned}$$

Étant donné que σ_i représente le bruit de mesure, nous avons $\sigma_0 = \sigma_1 = \sigma$. Le bruit de mesure est supposé identique suivant les deux hypothèses. C'est-à-dire qu'il est indépendant des hypothèses, il est intrinsèque au mécanisme d'estimation de la structure. L'équation de la somme des ratios de vraisemblance logarithmiques, à partir des équations précédentes, est réécrite :

$$\begin{aligned} S_\tau^n &= \sum_{j=\tau}^{n} ln\left(\frac{f_1\left(y_j|\theta_1\right)}{f_0\left(y_j|\theta_0\right)}\right) && (7.43)\\ &= \sum_{j=\tau}^{n} ln\left(\frac{e^{-\frac{1}{2}\frac{(y_j-\mu_1)^2}{\sigma^2}}}{e^{-\frac{1}{2}\frac{(y_j-\mu_0)^2}{\sigma^2}}}\right) && (7.44) \end{aligned}$$

L'équation se simplifie en :

$$S_\tau^n = \frac{(n-\tau+1)(\mu_1-\mu_0)^2}{2\sigma^2} \quad (7.45)$$

Le lecteur intéressé pourra se référer aux annexes en B pour obtenir la preuve de l'équation simplifiée. Il est intéressant de noter que le résultat dépend uniquement de la différence au carré des moyennes des hypothèses. En effet, le terme $(n - \tau + 1)$ est un facteur constant dans le cadre de cette approche (*cf.* la prochaine section 7.2.2). Il représente le nombre d'échantillons supposés appartenir à l'hypothèse H_1. Quant au terme $\frac{1}{2\sigma^2}$, il s'agit aussi d'un facteur constant. S_τ^n est donc simplement un vecteur de nature différentielle quadratique mesurant l'écart entre deux hypothèses.

L'équation 7.45 est établie pour un signal unidimensionnel. Or, le signal analysé est multidimensionnel et chacune des dimensions est traitée indépendamment. Pour obtenir l'équation multidimensionnelle, il suffit de considérer μ_0, respectivement μ_1, comme le vecteur des moyennes de chacune des dimensions suivant l'hypothèse H_0, respectivement H_1. S_τ^n est alors un vecteur de somme de ratios de vraisemblance logarithmiques. La décision de rupture de modèle s'effectue sur ce vecteur.

7.2.2 Fenêtre glissante et filtrage de Kalman

Bien qu'ayant l'équation permettant de prendre une décision en fonction d'un seuil h, il reste toutefois un élément important. En effet, la fonction de vraisemblance est une gaussienne, donc caractérisée par une moyenne et un écart-type. Il est nécessaire d'estimer ces deux grandeurs pour les hypothèses H_0 et H_1. D'après les calculs de maximum de vraisemblance, les meilleurs estimateurs de ces grandeurs pour une distribution gaussienne correspondent pour la moyenne à la moyenne des observations et pour l'écart-type à l'écart-type des observations. Il faut donc déterminer quelles sont les observations, ainsi que la quantité, qui seront prises en compte pour l'estimation.

D'autre part, un choix astucieux des observations permet de rendre l'équation de somme de ratios de vraisemblance uniquement dépendante de la différence de moyennes. L'équation actuelle 7.45 constitue une approche hors-ligne car il est nécessaire de tester toutes les hypothèses de rupture. Ceci est traduit par le fait que l'instant de rupture τ puisse correspondre à n'importe laquelle des observations de la séquence, soit $\tau \in [\![1, n]\!]$. Cette équation, suivie du test d'hypothèse $t_c = min\{n : \arg\max_\tau(S_\tau^n > h, 0 \leq \tau \leq n)\}$, mène au partitionnement optimal mais engendre un temps de calcul intraitable. L'objectif est alors de ramener cet algorithme à une considération en-ligne avec un instant d'alarme qui correspond au mieux à l'instant de rupture. Le partitionnement n'est pas forcément optimal mais il doit s'en rapprocher.

Pour résoudre l'ensemble de ces problèmes nous introduisons un concept simple mais efficace : la fenêtre glissante. La fenêtre glissante est un intervalle d'estimation de taille fixe qui se déplace sur le signal. Elle permet d'estimer les densités de probabilité à l'instant n à partir des échantillons de l'intervalle $[\![n-N+1, n]\!]$, N étant la largeur de la fenêtre. Dans notre cas, la fenêtre glissante est divisée en deux parties égales : le première moitié de la fenêtre représente les échantillons pour l'estimation de $f_0(y|\theta_0)$ et la deuxième moitié représente les échantillons pour l'estimation de $f_1(y|\theta_1)$. La fenêtre glissante contient alors les deux hypothèses H_0 et H_1. Une représentation de la fenêtre glissante est donnée sur la figure 7.2.

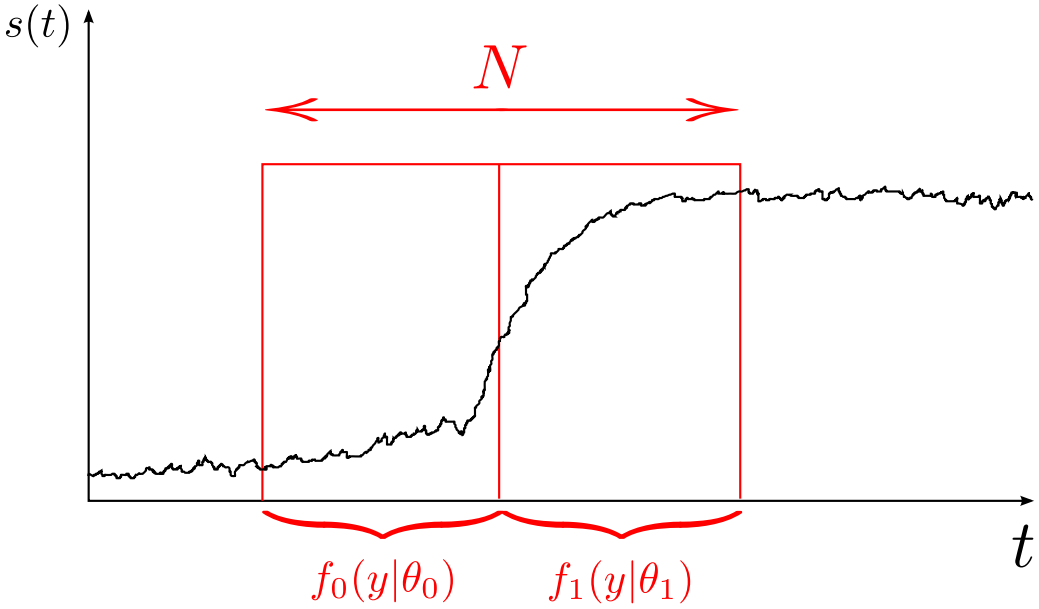


FIG. 7.2 – Représentation de la fenêtre glissante en rouge et séparée en deux moitié pour l'estimation des densités de probabilités correspondant aux hypothèses H_0 et H_1.

L'hypothèse de rupture τ est donc testée au milieu de la fenêtre glissante. Les avantages de cette approche sont :

– Estimation en temps constant des paramètres des distributions de probabilités.

– Un seul test de changement de modèle.

Ceci rend l'approche très efficace en terme de temps de calcul avec des hypothèses raisonnables. Par contre, la rupture de modèle n'est pas forcément optimale du fait d'une estimation pas forcément optimale des distributions de probabilité, les estimations étant simplement locales autour du point de rupture. Toutefois, cette approximation reste convenable. En effet, autour du point de rupture,

deux distributions différentes sont présentes. De plus, il n'est pas nécessaire d'avoir les distributions représentant complètement les lieux précédent et à venir. Néanmoins, cela change la granularité de l'approche. Elle est maintenant locale et peut éventuellement détecter des changements moins abrupts au sein d'un même lieu. Tant que la distribution ne change pas au sein d'un même lieu, ce qui est l'hypothèse de départ de cette approche, l'approche locale est équivalente à une approche plus globale.

Cette méthode permet d'obtenir l'instant exact de la rupture t_c et non l'instant d'alarme t_a. Une légère approximation de la localisation de la rupture (*i.e.* t_c proche de la valeur optimale) est bien plus bénéfique qu'un instant d'alarme ne permettant pas de relocaliser la rupture (*i.e.* impossibilité d'obtenir t_c à partir de t_a). En ce qui concerne l'équation de somme des ratios de vraisemblance, étant donné que le terme $n-\tau+1$ correspond au nombre d'échantillons permettant d'estimer $f_1(y|\theta_1)$, il s'agit d'une valeur constante. L'équation est alors uniquement dépendante de la différence de moyennes.

$$S_\tau^n = \frac{(n-\tau+1)(\mu_1-\mu_0)^2}{2\sigma^2} \tag{7.46}$$

$$= k(\mu_1-\mu_0)^2 \tag{7.47}$$

k est une valeur constante qui ne présente pas d'importance, elle se résume à un amplificateur de la différence de moyennes. L'inconvénient de cette approche est qu'il faut choisir une taille de fenêtre glissante correcte. Il faut que cette dernière soit suffisamment grande pour permettre une estimation correcte des paramètres de distributions. Elle doit aussi être de taille relativement petite pour ne pas entraîner un temps de calcul trop long et pour ne pas engendrer un retard de détection trop grand. Le dernier instant de signal connu est l'instant n correspondant au dernier échantillon de la fenêtre et appartenant à l'hypothèse H_1. L'hypothèse de rupture étant faite au centre de la fenêtre, ceci entraîne un écart de détection de $N/2$ échantillons. Ce paramètre est à relativiser avec la vitesse de déplacement du robot et la fréquence d'acquisition et de traitement des données. L'importance de la taille de la fenêtre glissante sera rediscutée dans la partie sur les résultats expérimentaux.

Étant données les hypothèses faites, seules les moyennes μ_0 et μ_1 sont à estimer. Le processus devient donc extrêmement simple. Le signal reste par contre assez bruité comme le dénote la figure 7.1. Un filtre de Kalman est ajouté pour modéliser l'évolution des moyennes et imposer un modèle de bruit. Les détails du filtre de Kalman ne sont pas exposés car il s'agit d'un cas simple d'utilisa-

tion ne requérant pas une connaissance approfondie de la théorie. Grâce au filtre, les moyennes sont estimées avec un minimum de variance donnant alors la meilleure estimation en considérant le bruit. Le modèle d'évolution des moyennes est simple. Au sein d'un même lieu, les moyennes sont sensées rester constantes. Le modèle d'évolution du filtre de Kalman discret est alors la matrice d'identité I. Elle permet de modéliser le fait que la prochaine valeur de moyenne sera la même que l'ancienne. Le filtre possède un bruit de modèle très faible imposant un respect assez strict du modèle suivant l'hypothèse H_0 mais un bruit de mesure un peu plus important permettant la prise en compte du passage de l'hypothèse H_0 à l'hypothèse H_1 (traduit par un changement de moyenne). Les deux moyennes estimées par le filtre de Kalman sont reportées au centre de la fenêtre glissante donnant lieu à un écart de moyenne représentant la valeur S_τ^n à un facteur multiplicatif constant près. Le modèle de filtre de Kalman utilisé se résume aux équations suivantes :

$$\begin{cases} x_n = x_{n-1} + w \\ y_n = x_n + v \end{cases} \tag{7.48}$$

x_n est le vecteur d'état contenant les moyennes μ_0 et μ_1 pour chacune des dimensions, y_n correspond aux valeurs du signal étudié. w est le bruit de modèle et v le bruit de mesure. À partir de ces équations, la meilleure estimée du vecteur d'état $\hat{x_n}$ est déterminée. Les équations du filtre de Kalman sont alors les suivantes :

– Estimation

$$x_{n|n} = x_{n|n-1} + K * ([\mu_0, \mu_1]^T - x_{n|n-1}) \tag{7.49}$$

$$P_{n|n} = (I - K) * P_{n|n-1} \tag{7.50}$$

– Prédiction

$$x_{n+1|n} = x_{n|n} \tag{7.51}$$

$$P_{n+1|n} = P_{n|n} + W \tag{7.52}$$

$$K = P_{n+1|n}(P_{n+1|n} + V)^{-1} \tag{7.53}$$

– Notations

$x_{n|n}$ est l'estimation optimale du vecteur d'état : $\hat{x_n}$

$x_{n+1|n}$ est la prédiction du vecteur d'état à l'instant suivant

$P_{n|n}$ est la matrice de covariance de l'erreur d'estimation

$P_{n+1|n}$ est la matrice de covariance de prédiction de l'erreur d'estimation

K est le gain de Kalman optimal

W est la matrice de covariance du bruit de modèle

V est la matrice de covariance du bruit de mesure

Le filtre de Kalman permet d'obtenir les signaux présentés sur la figure 7.3.

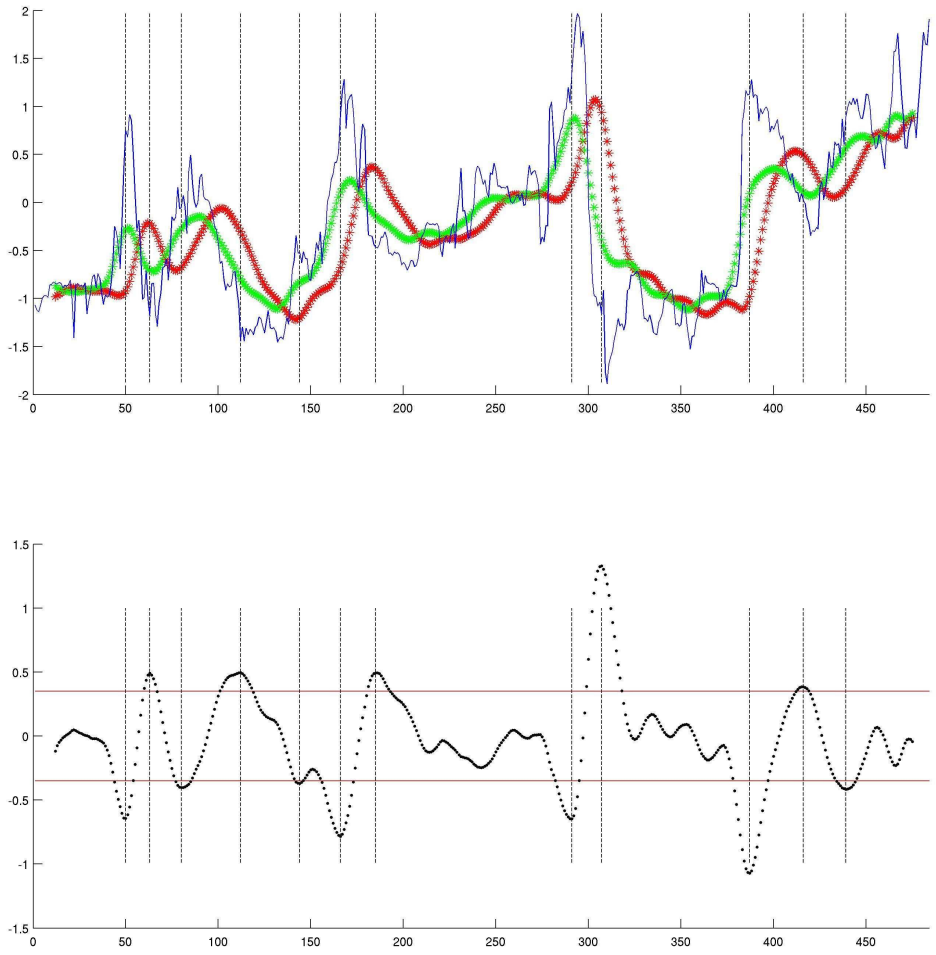


FIG. 7.3 – Le graphique du haut montre l'évolution des moyennes μ_0 en rouge et μ_1 en vert. Le graphique du bas représente l'évolution de la différence des moyennes et le seuil h choisi.

L'exemple de la figure correspond à un signal suivant une dimension mais permet de visualiser l'évolution des deux moyennes et de leur différence. Il est clairement visible que les différences les plus importantes correspondent aux variations du signal d'origine les plus importantes. Ces graphiques permettent de valider l'approche et surtout les hypothèses émises pour élaborer le système. L'analyse complète de l'approche sera discutée dans la partie concernant les résultats. Pour décider s'il y a rupture de modèle, il suffit de comparer les pics du signal à la valeur du seuil de décision h. Dans le graphique, les ruptures sont notées par une barre verticale correspondant aux cas où $S_\tau^n \geq h$. Il est intéressant de noter que la localisation de la rupture de modèle ne dépend pas de la valeur du seuil h choisi. Le seuil influe simplement sur le taux de segmentation. Ceci confirme le fait que nous obtenons l'instant exact t_c de la rupture et non un instant d'alarme t_a. t_c reste toutefois l'instant exact relativement aux hypothèses émises et n'est pas forcément l'instant t_c optimal. Cette décision est prise sur une seule dimension. Il faut élaborer un système permettant de choisir s'il y a rupture mais basé sur une conjonction de décisions des différentes dimensions.

7.2.3 Méthode empirique de sélection des changements de lieu

La méthode permettant de prendre une décision unique à partir des décisions des différentes dimensions est une méthode empirique ne reposant pas sur un fondement théorique. Elle permet cependant, dans cette première approche, d'avoir un système opérationnel et de démontrer la validité de celle-ci. La sortie du système de filtrage et de détection précédemment décrits est un ensemble de décisions suivant chacune des dimensions. Un système intuitif est de prendre une décision de rupture globale lorsque suffisamment de dimensions indépendantes indiquent une rupture. Nous avons opté pour un quota de 30% des dimensions. Il faut que ce quota ne soit pas trop faible afin d'éviter d'avoir des décisions pour chacune des dimensions, l'environnement serait sur-segmenté. Si le quota est trop grand, alors aucune décision de rupture globale n'est prise. En effet, il est possible d'avoir une variation importante de texture horizontale dénotant un changement de lieu sans pour autant avoir une variation significative suivant la verticale. Une décision de rupture générale basée sur une décision de rupture de chacune des dimensions est alors beaucoup trop restrictive. Le quota choisi est un bon compromis entre ces deux extrêmes.

Toutefois, la décision de rupture globale basée sur un ensemble de décisions communes suivant les différentes dimensions est une problématique. En effet, les décisions de rupture suivant les différentes

dimensions ne sont pas forcément prises exactement au même instant. Il se peut qu'il y ait par exemple un écart d'un échantillon. Il semble évident qu'il s'agit de décisions pour la même rupture. Pour pallier ce problème, toutes les décisions proches dans un certain intervalle sont considérées comme étant les décisions pour la même rupture. L'instant de rupture choisi est alors l'instant moyen des différentes décisions. La taille de l'intervalle est de quelques échantillons, environ une dizaine. Cet intervalle peut être considéré comme une fenêtre d'observation de l'ensemble des dimensions du signal. Au sein de cette fenêtre, si le quota précédemment défini est atteint, alors une décision de rupture globale de modèle est prise. Le mécanisme est illustré par la figure 7.4.

Cette méthode empirique est issue de constatations, d'études des signaux générés et des décisions de ruptures prises. Elle est le résultat d'hypothèses logiques sur le comportement de l'algorithme corrélé à l'environnement étudié.

7.3 Raffinement du modèle de détection

Bien que l'approche précédente ait permis de démontrer la validité des hypothèses émises, elle pâtit de contraintes et approximations fortes. Notamment, le descripteur GIST modifié n'est pas complètement adapté à la représentation sphérique. De même, l'hypothèse d'indépendance des dimensions fait perdre l'information de corrélation qui existe au sein de l'environnement. Dans cette deuxième approche, nous utilisons le spectre de l'image en termes d'harmoniques sphériques. Le vecteur de description de l'environnement est, pour rappel, la concaténation des 25 premières composantes du spectre f_l^m (*cf.* section 6.3). L'équation de calcul de la somme des ratios de vraisemblance est revue afin de prendre en compte l'interdépendance des différentes dimensions.

7.3.1 Révision de l'équation de détection de changement de lieu

Cette section est consacrée à la révision de l'équation de détection de changement de modèle basée sur la somme des ratios de vraisemblance logarithmiques. Nous allons y introduire la variance propre de chaque dimension qui, pour rappel, était considérée comme un bruit de mesure avec $\sigma_0 = \sigma_1 = \sigma$. L'interdépendance des dimensions sera introduite grâce à la covariance. Pour prendre en compte ces deux considérations, la matrice de covariance de la distribution des observations est prise en compte au sein du système. L'équation générale est la suivante :

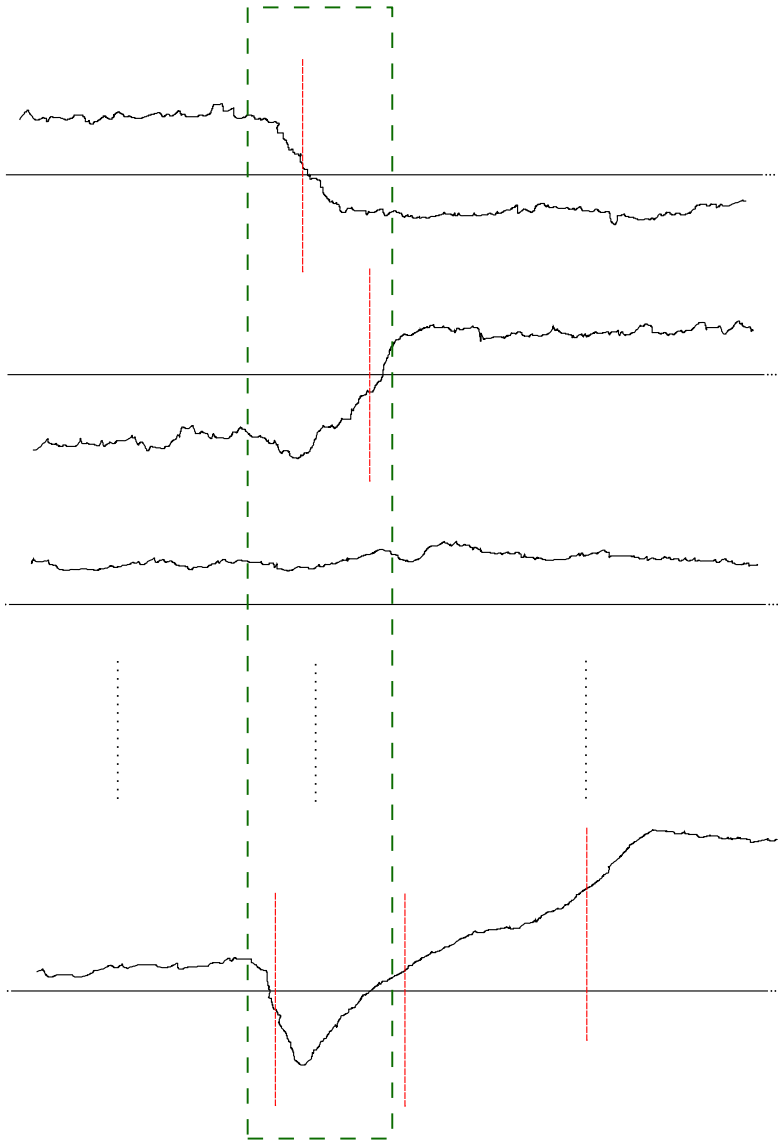


FIG. 7.4 – Mécanisme empirique de décision de rupture globale de modèle à partir des décisions de rupture des différentes dimensions. Les barres verticales correspondent aux ruptures de modèle suivant chacune des dimensions. Le rectangle vert est la fenêtre d'observation permettant de regrouper les multiples décisions en une seule.

$$S_\tau^n = \sum_{j=\tau}^{n} ln\left(\frac{f_1\left(y_j|\theta_1\right)}{f_0\left(y_j|\theta_0\right)}\right) \tag{7.54}$$

Dans l'approche précédente, les fonctions de vraisemblance étaient des distributions gaussiennes univariées. Afin de prendre en compte la covariance, les fonctions de vraisemblance seront désormais des distributions gaussiennes multivariées. L'équation 7.54 devient :

$$S_\tau^n = \sum_{j=\tau}^{n} ln \left(\frac{\frac{1}{(2\pi)^{k/2}|\Sigma_1|^{1/2}} e^{-\frac{1}{2}(y_j-\mu_1)^T \Sigma_1^{-1}(y_j-\mu_1)}}{\frac{1}{(2\pi)^{k/2}|\Sigma_0|^{1/2}} e^{-\frac{1}{2}(y_j-\mu_0)^T \Sigma_0^{-1}(y_j-\mu_0)}} \right) \tag{7.55}$$

où $\theta_0 = (\mu_0, \Sigma_0)$, respectivement $\theta_1 = (\mu_1, \Sigma_1)$, est la paramétrisation de la distribution gaussienne multivariée suivant l'hypothèse H_0, respectivement H_1. μ_i et Σ_i sont respectivement le vecteur des moyennes et la matrice de covariance de la distribution $f_i(y|\theta_i)$. k est le nombre de dimensions. Ce paramètre ne présente pas d'intérêt car il disparaît dans l'équation finale. L'équation simplifiée prend la forme suivante :

$$\begin{aligned} S_\tau^n = & \frac{n-\tau+1}{2} ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) + \frac{n-\tau+1}{2} \left(\mu_0^T \Sigma_0^{-1} \mu_0 + \mu_1^T \Sigma_1^{-1} \mu_1 - 2\mu_0^T \Sigma_0^{-1} \mu_1 \right) + \\ & \frac{1}{2} \sum_{j=\tau}^{n} \left(y_j^T \left(\Sigma_0^{-1} - \Sigma_1^{-1} \right) y_j \right) \end{aligned} \tag{7.56}$$

Le lecteur intéressé pourra se référer à l'annexe C pour obtenir les détails permettant d'obtenir l'équation simplifiée. Cette équation est bien plus riche que celle établie lors de la première approche. Quelques points importants sont à noter sur cette équation. En considérant le cas où $\Sigma_0 = \Sigma_1 = \Sigma$, correspondant au fait que les deux moitiés de la fenêtre glissante suivent la même distribution et donc la même hypothèse H_0, l'équation devient :

$$S_\tau^n = \frac{n-\tau+1}{2} (\mu_0 - \mu_1)^T \Sigma^{-1} (\mu_0 - \mu_1) \tag{7.57}$$

L'équation obtenue est pour ainsi dire la même que celle obtenue pour la première approche (équation 7.45). Si de plus Σ est une matrice diagonale, ne contenant alors que la variance suivant chacune des dimensions, l'équation devient exactement la même. L'intérêt de cette constatation est d'observer le lien entre les deux approches et de mettre en évidence l'hypothèse forte émise pour la première expérimentation. Dans le cas simplifié (équation 7.57), le changement de lieu dépend uniquement de la différence de moyennes tandis que dans le cas avec l'équation plus complète, l'étalement de la distribution est pris en compte et pondère l'écart de moyennes. La prise en compte de l'étalement des distributions est importante dans la mesure où un petit écart entre deux moyennes de gaussiennes peu étalées peut être plus significatif qu'un grand écart de moyennes entre deux gaussiennes très étalées. La première approche peut donc entraîner la détection de ruptures non significatives et ne pas détecter des ruptures bien plus importantes. Ceci limite donc très fortement la première approche étant donné

que les contraintes de faibles taux de faux positifs et de faux négatifs ne sont pas satisfaites. Les résultats de la seconde approche sont bien meilleurs (*cf.* section 8.2.2).

Une analyse des différents termes de l'équation complète 7.56 permet de comprendre son comportement et sa relation avec les changements de modèles.

- Le premier terme, $\frac{n-\tau+1}{2} ln\left(\frac{|\Sigma_0|}{|\Sigma_1|}\right)$, est un ratio des déterminants des matrices de covariance qui s'annule si elles sont identiques. Ce terme est donc directement lié à l'étalement de chacune des distributions gaussiennes. Il permet d'ajuster la valeur S_τ^n en fonction de ce rapport ; comme mentionné précédemment, il permet de relativiser l'écart de distributions vis-à-vis des étalements de ces dernières. Si $f_0(y|\theta_0)$ est plus étalée que $f_1(y|\theta_1)$ alors $|\Sigma_0| > |\Sigma_1|$. Le ratio de vraisemblance alors augmente. Réciproquement, si $f_1(y|\theta_1)$ est plus élevée que $f_0(y|\theta_0)$ alors la valeur du ratio de vraisemblance va fortement diminuer. Le deuxième cas possède un effet plus fort du fait de la loi logarithmique.
- Le second terme, $\frac{n-\tau+1}{2}\left(\mu_0^T\Sigma_0^{-1}\mu_0 + \mu_1^T\Sigma_1^{-1}\mu_1 - 2\mu_0^T\Sigma_0^{-1}\mu_1\right)$, est directement la différence quadratique des moyennes pondérées par les matrices de covariance de chacune des distributions. Il n'est pas possible de le factoriser sous cette forme du fait de la pondération mais par contre il conserve cette propriété de relativiser l'écart des moyennes par rapport à l'étalement des distributions.
- Le dernier terme, $\frac{1}{2}\sum_{j=\tau}^{n}\left(y_j^T\left(\Sigma_0^{-1} - \Sigma_1^{-1}\right)y_j\right)$, correspond à la somme des observations au carré sous l'hypothèse H_1 pondérées par l'écart de covariance entre les deux hypothèses.

Le dernier point intéressant de cette équation est qu'elle solutionne automatiquement le problème de décisions multidimensionnelles. En effet, l'équation est basée sur les observations y_j (vecteurs), les moyennes μ_i (vecteurs) et les matrices de covariance Σ_i (matrices) mais le résultat final S_τ^n est un scalaire. Cette équation permet donc de prendre en compte l'aspect multidimensionnel du signal d'origine et de concaténer toute l'information dans une seule valeur. En considérant l'évolution de S_τ^n au cours du temps, il résulte un signal unidimensionnel. Il suffit alors de détecter les maxima de ce signal et de comparer ces maxima à la valeur du seuil choisi h. S'ils sont supérieurs alors il y aura changements de modèles aux instants t_c déterminés par les maxima. Un exemple de signal $S_\tau^n(t)$ obtenu est affiché sur la figure 7.5. Dans l'exemple, le signal est lissé par un filtre gaussien glissant. Ce dernier permet

de supprimer le bruit (amplitude faible, fréquence élevée) tout en conservant la netteté des variations du signal (amplitude élevée, fréquence faible). Le filtre à moyenne glissante lisse trop les variations significatives du signal. Le signal originel est peu bruité mais présente toutefois de multiples pics aux endroits où il ne devrait y en avoir qu'un seul. Le bruit présent est donc suffisamment important pour nécessiter du filtrage.

Le signal d'exemple permet d'approfondir la notion de décision sur le résultat de cette nouvelle équation. L'avantage est qu'il permet de mettre en évidence la netteté du signal et la localisation très précise des pics. Le signal $S_\tau^n(t)$ constitue en fait un ensemble d'hypothèses réparties dans le temps et surtout l'espace, chaque instant t étant une hypothèse de rupture avec la possibilité d'avoir $t = t_c$. En reprenant l'équation 7.37, l'instant t_c est l'instant correspondant à la valeur maximale de l'hypothèse de rupture pourvu que cette valeur soit supérieure à un seuil h. Cela confirme qu'il suffit de relever les maxima locaux du signal et de les comparer au seuil de décision h pour déterminer les ruptures de modèle.

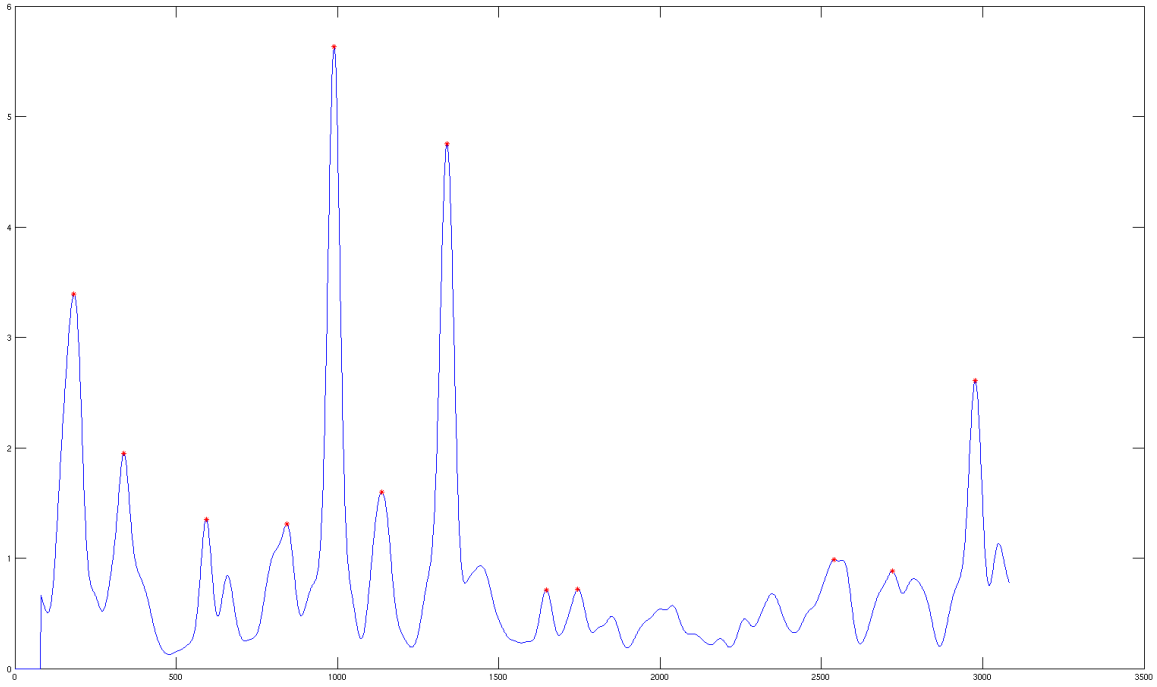


FIG. 7.5 – Exemple de signal $S_\tau^n(t)$ obtenu en considérant l'équation complète de détection de changement de lieu. Les croix rouges correspondent aux changements de lieu détectés.

7.3.2 Estimation des densités de probabilité

Bien que les implications de l'utilisation d'une fenêtre glissante ait déjà été évoquées dans la section 7.2.2, il reste toutefois un point important à aborder. Dans chaque moitié de la fenêtre glissante, une densité de probabilité est estimée. Dans le cas présent, il s'agit de gaussiennes multivariées caractérisées par une moyenne et une matrice de covariance. Comme énoncé précédemment, les meilleurs estimateurs de ces paramètres, d'un point de vue maximum de vraisemblance, sont la moyenne des échantillons et la matrice de covariance des échantillons. Pour utiliser ces estimateurs, et d'une manière générale lors de l'estimation de densité de probabilité, deux hypothèses doivent être satisfaites :

– Le nombre d'échantillons doit être suffisamment élevé.
– Les échantillons doivent être indépendamment et identiquement distribués, *i.e. i.i.d.*

Ces conditions nécessaires influent sur le paramétrage de la fenêtre glissante et la façon d'échantillonner l'environnement. La nécessité d'une fenêtre glissante suffisamment large pour permettre l'estimation de deux densités de probabilité n'est pas rediscutée. Par contre, l'échantillonnage *i.i.d* est abordé car très important pour la qualité des résultats. L'hypothèse suffisante communément utilisée, et admise ici, est de supposer que les observations sont indépendantes. En ce qui concerne la répartition identique, cela signifie que les observations doivent occuper l'ensemble de l'espace occupé par la densité de probabilité et en proportion de cette dernière. En effet, il est impossible d'estimer correctement une gaussienne si toutes les observations obtenues se situent dans la queue de la gaussienne ou si elles sont plus nombreuses à une position excentrée par rapport à la moyenne. Dans le cas de la fenêtre glissante pour l'estimation de la structure de l'environnement, il est nécessaire de prendre certaines précautions car les cas précités peuvent facilement apparaître. Si le robot ne bouge pas pendant un intervalle de temps, plusieurs observations vont être extraites de l'environnement. Comme ces observations décrivent la structure de l'environnement, elles vont toutes se situer au même endroit, au bruit de mesure près, dans l'espace des probabilités biaisant ainsi l'estimée de la distribution. Il est donc nécessaire d'avoir des observations à partir de différentes positions dans le même endroit afin de bien échantillonner la structure et avoir des observations identiquement réparties. Le robot doit alors avoir un déplacement minimal entre deux observations afin de satisfaire cette condition. Il en résulte que toutes les observations consécutives ne seront pas forcément prises en compte dans la fenêtre glissante. Toutefois, il faut toujours suffisamment d'échantillons pour une estimation correcte. La fenêtre glissante utilise donc un ensemble d'échantillons séparés par une distance minimale. Nous dirons que la fenêtre glissante nécessite une distance minimale d'estimation des densités de probabilité.

7.4 Invariance aux changements de luminosité

Lors de nos expérimentations et d'une manière générale lors de l'utilisation d'images, des problèmes d'illumination apparaissent. En effet, lors du déplacement du robot, en changeant de pièce, en passant d'un environnement d'intérieur à environnement d'extérieur, des changements dans l'illumination ambiante apparaissent dans les images. Dans le cadre des expérimentations, le shutter automatique de la caméra était activé afin de gérer de manière automatique les variations de lumière et par conséquent la quantité de lumière acquise et la qualité de l'image. Il s'est avéré que cela entraîne d'importantes variations d'intensité dans les images résultantes. Il est possible d'avoir une image claire à un instant et d'avoir une image sombre quelques instants plus tard (*cf.* figure 7.6) du fait de reflets ou positions particulières de la caméra par rapport à la lumière des bureaux.

FIG. 7.6 – Effet du shutter automatique de la caméra sur le rendu des images. Deux images proches dans l'espace peuvent avoir des variations d'intensité importantes l'une par rapport à l'autre.

La solution la plus simple utilisée dans un premier temps est l'égalisation d'histogramme. Cette méthode permet de répartir équitablement les différents niveaux de gris (images en noir et blanc) améliorant ainsi l'information perçue de l'image. Une image sombre dans laquelle il est difficile de voir ce qu'elle représente devient beaucoup plus lisible après égalisation de son histogramme. Bien que satisfaisante au niveau visuel, cette méthode est désastreuse en ce qui concerne l'approche d'extraction d'information de structure de l'environnement. Comme expliqué en 6.2.3.1, l'information de structure de l'environnement est décrite par le contenu fréquentiel, que ce soit par l'intermédiaire du GIST ou des harmoniques sphériques. Le principe de l'égalisation d'histogramme est de considérer l'histogramme de l'image comme une distribution quelconque et de transformer cette distribution en une distribution uniforme des intensités afin d'utiliser toutes les valeurs disponibles pour décrire l'image. L'inconvénient de cette méthode est qu'elle dépend du contenu de l'image en terme du nombre de pixels par intensité mais absolument pas de l'aspect fréquentiel. Il s'agit d'une application non linéaire propre à chaque image et dont la transformation ne conserve pas les fréquences originelles. De ce fait,

égaliser les histogrammes des images avant d'en extraire le contenu fréquentiel détruit complètement le processus en décorrélant les images successives. Le problème était difficilement identifiable dans la première approche utilisant l'équation de détection de changement de lieu simplifiée car le signal était déjà très bruité et l'hypothèse forte engendrait des fluctuations indésirables. L'égalisation d'histogramme semblait même apporter un mieux à l'algorithme. Cette amélioration provient certainement du fait que chacune des dimensions était considérée indépendamment, ne préservant pas ainsi une cohérence suffisante du contenu de l'image. L'égalisation d'histogramme et le calcul de l'hypothèse de rupture reposent alors tous deux sur une information d'image non complètement cohérente. Il est alors possible qu'il y ait eu amélioration d'un signal indépendamment du contenu de l'image. Le système de détection empirique du lieu de rupture de modèle ajouté à cela permet de filtrer les détections de ruptures trop incohérentes. Cet ensemble a donc contribué à une amélioration apparente difficile à prouver.

En ce qui concerne la deuxième approche avec l'équation complète de détection de changement de lieu, la différence est très importante entre l'utilisation ou non de l'égalisation d'histogramme. La figure 7.7 permet d'illustrer les signaux filtrés obtenus de $S_\tau^n(t)$ avec et sans égalisation d'histogramme. Avec l'égalisation d'histogramme, le signal devient extrêmement bruité. Certains pics ont alors une amplitude qui diminue fortement, ils sont donc moins significatifs. L'amplitude de ces pics est du même niveau voire plus faible que celle du bruit. Ils sont indétectables à moins de diminuer le seuil de détection. Cette solution n'est pas satisfaisante dans la mesure où beaucoup de fausses ruptures de modèle seraient détectées. Même en lissant le signal, comme pour le cas sans l'égalisation d'histogramme, cela reste inexploitable car trop bruité. La solution adoptée est alors l'extraction de l'information de structure sans pré-traitement des images, donc sans égalisation d'histogramme. Il est intéressant de remarquer que le processus est insensible à la variation d'illumination provoquée par le shutter automatique. Ceci s'explique assez aisément avec l'équation de calcul des harmoniques sphériques :

$$f_l^m = \int_{\eta \in S^2} f(\eta) \overline{Y_l^m(\eta)} d\eta \tag{7.58}$$

et en supposant un modèle classique d'illumination affine :

$$g(\eta) = a.f(\eta) + b \tag{7.59}$$

Le modèle d'illumination affine est un modèle d'illumination global qui modélise correctement

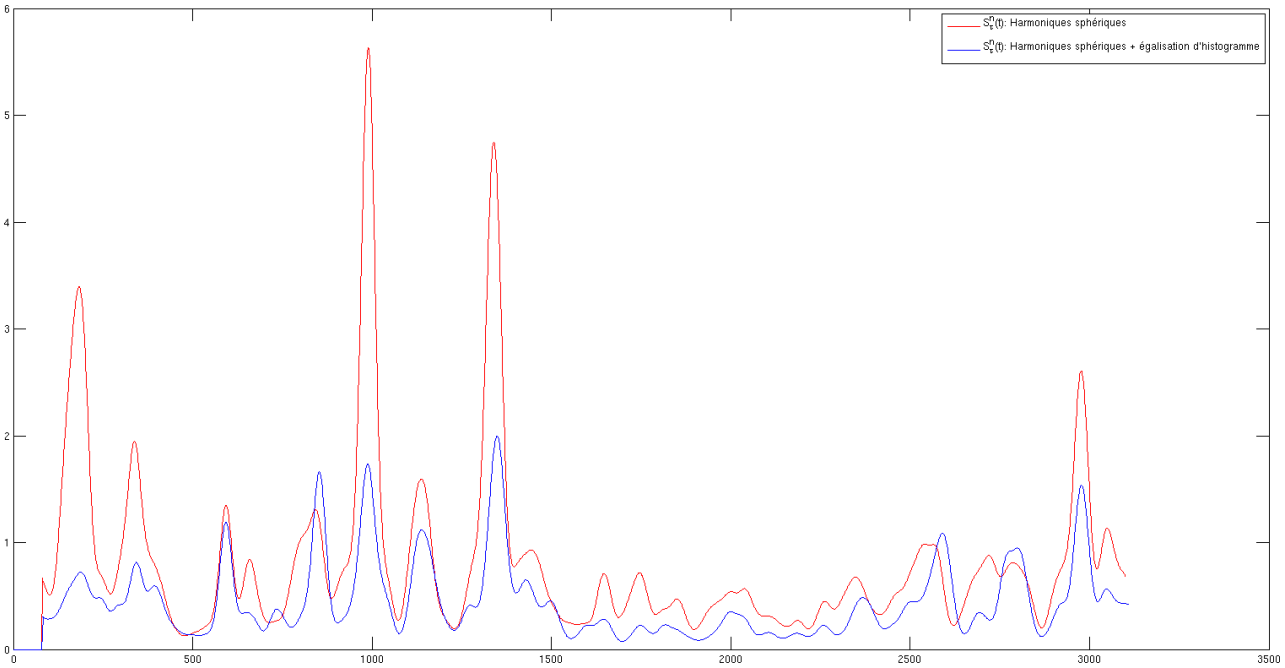


FIG. 7.7 – La courbe rouge représente la courbe de $S^n_\tau(t)$ obtenue à partir de l'approche avec les harmoniques sphériques mais sans l'égalisation d'histogramme. La courbe bleue correspond à la même approche mais avec en plus l'égalisation d'histogramme. La différence majeure réside dans la disparition de pics lors de l'utilisation de l'égalisation d'histogramme. Ces derniers se trouvent noyés dans le bruit.

la transformation effectuée par le shutter automatique sur l'image (augmentation ou diminution de l'intensité globale). Le spectre de l'image transformée s'obtient par l'équation suivante :

$$g_l^m = \int_{\eta\in S^2} g(\eta)\overline{Y_l^m(\eta)}d\eta \tag{7.60}$$

$$= \int_{\eta\in S^2} (a.f\,(\eta) + b)\,\overline{Y_l^m\,(\eta)}d\eta \tag{7.61}$$

$$= a\int_{\eta\in S^2} f(\eta)\overline{Y_l^m(\eta)}d\eta + b\int_{\eta\in S^2} \overline{Y_l^m(\eta)}d\eta \tag{7.62}$$

Le spectre final est alors :

$$g_l^m = \begin{cases} a.f_l^m + b \text{ si } l = m = 0 \\ a.f_l^m \text{ sinon} \end{cases} \tag{7.63}$$

Pour modéliser la variation d'intensité, le paramètre b du modèle affine est utilisé impliquant la considération $a = 1$. Autrement dit, le spectre de l'image transformée par l'effet du shutter est le même que l'image sans sauf pour la composante continue qui possède un offset. Ceci est logique puisque si

une intensité constante est ajoutée ou soustraite à l'image, seule la composante continue est modifiée. Le système est donc pour ainsi dire insensible à l'effet du shutter. Au final, en considérant le cas général d'illumination affine, le spectre de l'image transformée est pour ainsi dire égal au spectre de l'image d'origine à un facteur multiplicatif près. Étant donné le traitement effectué ensuite, le spectre est une variable aléatoire appartenant à une distribution gaussienne multivariée. Ce facteur multiplicatif a aura donc pour effet de multiplier la moyenne par a et la matrice de variance-covariance par a^2. Toutefois, en prenant en compte que le phénomène ne concerne qu'un nombre limité d'images lors du déplacement du robot, seules certaines observations sont concernées par cette transformation. De ce fait, étant donnée l'estimation des paramètres de la distribution, il en résultera un léger décalage de la moyenne et un léger étalement supplémentaire de la gaussienne. La description structurelle statistique du lieu n'est donc pas fondamentalement modifiée. Par contre, si le nombre d'observations concernées est trop important par rapport au nombre d'observations total, l'impact sur l'estimation de la distribution peut être plus important en fonction de la valeur du facteur a.

Cette justification convient pour l'effet du shutter et montre la robustesse du modèle à une illumination globale affine. Par contre, une illumination locale aura nécessairement un effet négatif sur l'estimation de la structure de l'environnement. Aucune étude des effets d'un changement d'illumination local (un reflet par exemple) n'est effectuée : ni au niveau de l'étude de son influence sur la segmentation ni au niveau des moyens possibles d'améliorer la robustesse de l'algorithme. Une piste serait les travaux de [Friedrich *et al.*, 2008] qui traitent des problèmes de changement d'illumination dans l'extraction des harmoniques sphériques pour la localisation du robot dans son environnement.

Chapitre 8

Résultats expérimentaux

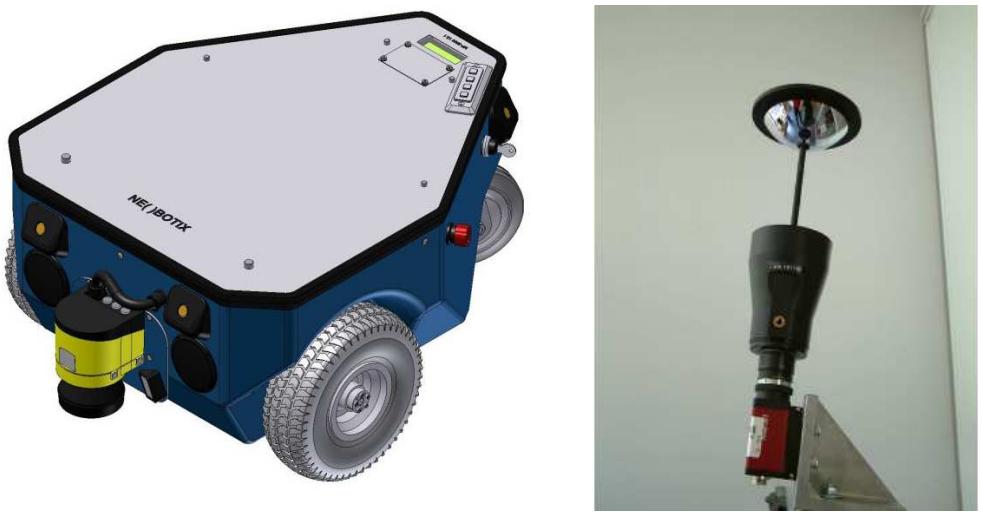
Sommaire

8.1 Présentation des expériences

Les expérimentations de détection de changement de lieu topologique se déroulent en plusieurs étapes. Les expérimentations portent sur les deux algorithmes : la preuve de concept utilisant le GIST et l'algorithme utilisant les harmoniques sphériques. Dans le cas du test de l'algorithme de preuve conceptuelle, des tests en environnements d'intérieur et d'extérieur ont été effectués. L'algorithme à base d'harmoniques sphériques a aussi été testé dans les deux types d'environnement afin de pouvoir comparer significativement les deux méthodes.
L'environnement d'extérieur et la plateforme robotique utilisée ne sont pas présentés ici car il s'agit exactement de la même séquence que celle utilisée pour les expérimentations de détection de fermeture de boucle. Les informations sont disponibles dans la partie 4.1.

La plateforme robotique est un petit robot mobile qui convient aux environnements d'intérieur. Il s'agit de la plateforme Neobotix MP-500 présentée sur la figure 8.1. Le système d'acquisition est une simple caméra omnidirectionnelle (*cf.* figure 8.1). L'image obtenue est une demi-sphère, il est alors très facile de la considérer comme une représentation sphérique. Toutefois, la représentation ne contiendra de l'information que dans la moitié inférieure de la sphère. L'avantage de cette caméra par rapport au système multi-caméras est qu'elle est beaucoup plus simple à utiliser. Le système multi-caméras sert aussi pour d'autres travaux dans lesquels il est nécessaire de calculer la carte de profondeur de l'environnement. Par contre, l'avantage du système multi-caméras est qu'il fournit de l'information sur une surface plus grande de la sphère et permet donc d'obtenir des caractéristiques plus précises de l'environnement.

L'environnement d'intérieur correspond au niveau 0 du bâtiment Kahn, nos bureaux sur le site de l'INRIA Sophia Antipolis. Le plan de masse affiché sur la figure 8.2 permet de donner une idée de l'environnement dans lequel le robot a évolué pour les expérimentations. L'inconvénient de ce schéma est qu'il ne représente pas le contenu des lieux. Les bureaux, l'ensemble des robots présents dans la halle robotique, ...*etc*, ne sont pas visibles sur un tel plan. Toutefois, il donne une idée globale suffisante pour comprendre l'essentiel des changements de lieu détectés. Il est notamment aisé de comprendre les changements de lieu associés aux changements de pièce. L'avantage de l'environnement d'intérieur sur l'environnement d'extérieur est qu'il est plus facile de comprendre un changement de lieu associé à une rupture de la continuité structurelle de l'environnement.



(a) Néobotix MP-500 (b) Caméra omnidirectionnelle

FIG. 8.1 – Robot d'expérimentation en environnement intérieur : plateforme Neobotix MP-500 et caméra omnidirectionnelle.

Comme expliqué dans la partie 7.3.2, l'estimation des densités de probabilité est un élément important de l'algorithme. La bonne estimation repose sur un bon échantillonnage de l'environnement. Dans notre cas, ce dernier repose sur l'utilisation de la fenêtre glissante permettant de sélectionner un certain nombre d'échantillons séparés par une distance minimale. Dans les expérimentations, une fenêtre de 80 échantillons est utilisée. La distance minimale choisie est de 0.015 m. La distance d'estimation de la fenêtre glissante est donc 1.2 m, soit une distance de 0.6 m pour l'estimation de distribution de probabilité dans chacun des deux environnements différents supposés. Il en résulte qu'il est impossible de détecter deux changements de lieu à une distance inférieure à 0.6 m. La distance 0.6 m semble raisonnable pour une estimation de la densité de probabilité d'un lieu. Elle est suffisamment grande pour permettre une estimation cohérente de l'environnement. Elle est aussi suffisamment petite pour permettre la détection de plus petits environnements, une distance trop grande lisserait l'estimation d'un petit environnement dans un plus grand le rendant ainsi indétectable.

En ce qui concerne les images traitées par l'algorithme, la séquence est constituée de 21473 images acquises à la fréquence de 25 Hz. La taille des images est de 426x128. Il s'agit donc de petites images. D'autre part, elles sont en noir et blanc. La couleur n'est pas utilisée pour décrire la structure de

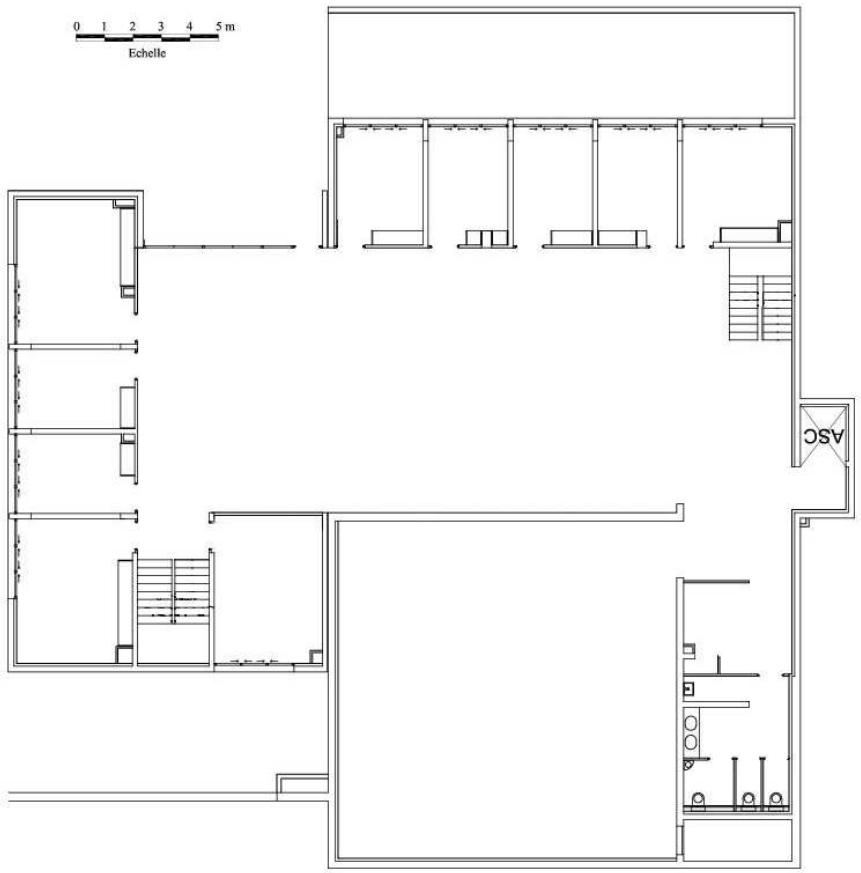


FIG. 8.2 – Environnement d'intérieur pour les expérimentations de détection de changement de lieu : Niveau 0 du bâtiment Kahn sur le site de l'INRIA Sophia Antipolis.

l'environnement.

Dans les analyses de résultats, les explications sont essentiellement données en termes de changements de lieu du fait de l'approche basée sur une méthode de détection de rupture de modèle. Il est toutefois à noter que, si les changements de lieu sont importants, les lieux topologiques définis entre deux changements de lieu sont aussi importants. Il sera alors intéressant de remarquer dans les résultats la consistance des lieux topologiques obtenus à partir des changements de lieu détectés.

8.2 Résultats et analyses

8.2.1 Première approche : utilisation du GIST

Analyse en environnement intérieur

L'algorithme a d'abord été testé sur un environnement d'intérieur afin de valider notre approche avec des changements de lieu facilement identifiables. Les résultats de l'expérimentation sont présentés sur la figure 8.3. Il s'agit du plan de masse du bâtiment sur lequel sont superposés la trajectoire suivie par le robot ainsi que les changements de lieu (représentés par des croix) identifiés par notre algorithme. Les lieux topologiques sont donc définis entre deux croix. Il est à noter que la première rupture détectée n'est pas une vraie rupture et correspond à des mouvements très proches de la caméra lors de l'initialisation du robot pour l'acquisition.

De manière très intéressante, nous remarquons qu'aucune rupture n'est détectée à l'emplacement des portes mais plutôt juste avant et juste après la porte. Le phénomène est observable sur les couples d'images (406, 751) et (2446, 2731) de la figure 8.3. Ceci signifie que du point de vue de l'algorithme, les pas de portes sont considérés comme des lieux topologiques à part entière. Ce résultat n'est pas intuitif mais il s'explique très bien de par le fait que les portes constituent des transitions et sont très différentes des environnements se trouvant de part et d'autre de la porte. Aux emplacements des portes, les images contiennent essentiellement les montants des portes, les portes elles-mêmes et une petite partie de chacun des environnements des deux côtés. En termes de fréquences et d'orientations, les composantes les plus significatives sont les orientations de 0°, représentant les lignes verticales, dans le domaine des hautes fréquences, en relation avec les bords bien marqués des montants de porte. La porte elle-même est représentée par des basses fréquences dans toutes les orientations étant donné qu'elle n'est pas texturée dans notre expérimentation. Avant et après l'emplacement des portes, ces orientations et fréquences sont moins significatives car les pas de portes ne constituent qu'une petite partie de l'image. Avant et après, les images sont caractérisées par des fréquences et orientations propres à chaque environnement. Toutefois, d'un point de vue construction de carte topologique, le pas de porte est très significatif en tant que moyen d'accès (donc une arête du graphe) mais pas en tant que lieu topologique (nœud du graphe).

Un élément important de ce type d'algorithme est que lorsque le robot revient dans des environ-

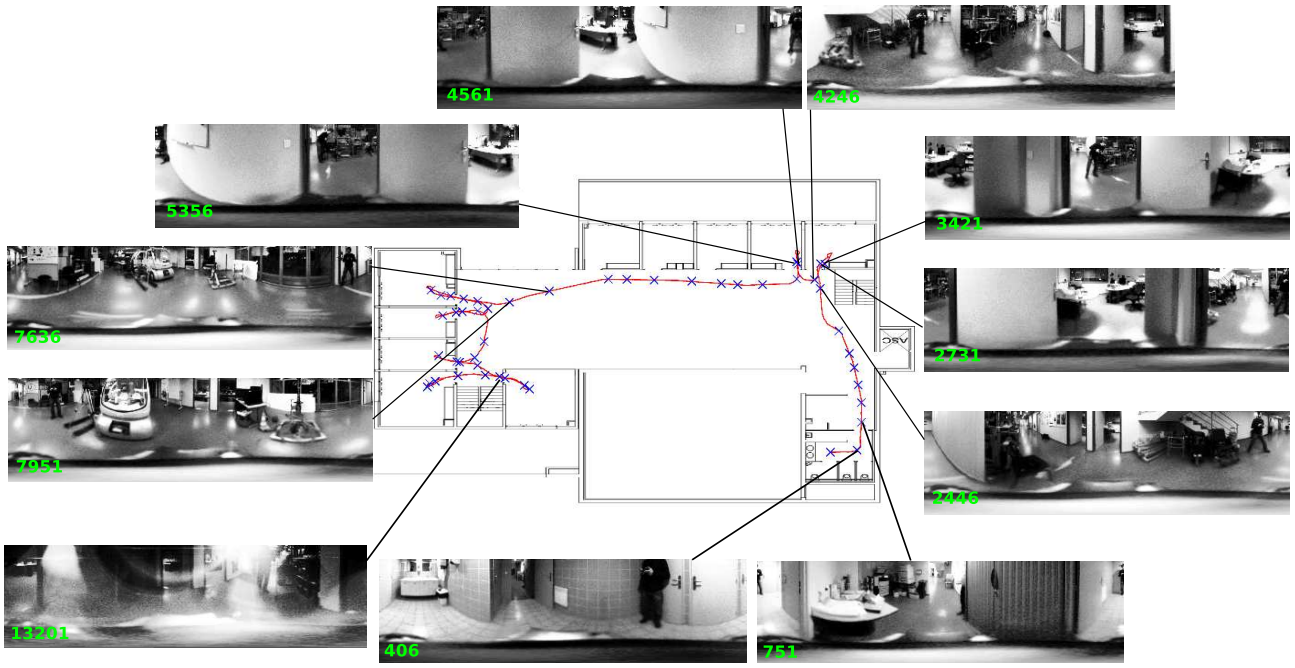


FIG. 8.3 – Résultats de la méthode de segmentation de l'environnement à base de GIST sur un milieu d'intérieur : le bâtiment Kahn.

nements déjà visités, les ruptures doivent être détectées aux mêmes endroits. Les couples d'images (2731, 3421) et (4561, 5356) montrent des exemples de ruptures localisées au même endroit lors de la revisite. Les exemples donnés ici montrent le cas lorsque le robot entre dans un bureau, fait demi-tour et quitte le bureau. De plus, les images (2446, 4246) sont proches spatialement et visuellement très similaires. Elles correspondent aux images lorsque le robot entre dans le premier bureau et lorsqu'il le quitte. La détection de changement de lieu est donc stable vis-à-vis de l'environnement. Quelque soit le chemin suivi par le robot, les changements de lieu se situent à des endroits très proches. Ces résultats démontrent bien la capacité de l'algorithme de faire de la segmentation de l'environnement en lieux topologiques à partir de la structure extraite de l'environnement et indépendamment du chemin suivi par le robot.

Des changements de lieu sont détectés tout au long de la trajectoire du robot et certains ne correspondent pas à des emplacements de portes. Ils n'en sont pas moins de véritables lieux de changement correspondant à d'importantes variations structurelles dans l'environnement perçu. De tels changements de lieu sont illustrés par les images 7636 et 7951 sur la figure 8.3. Dans ces deux images le même environnement est observé. Cependant, la première image semble décrire un environnement avec beau-

coup d'espace disponible. Le lieu topologique défini entre ces deux images est lié à cet espace ouvert. Le second changement de lieu décrit un lieu beaucoup plus encombré. Bien que les mêmes objets (que dans l'image de changement de lieu précédente) soient présents, le robot est plus près de ceux-ci. Ils sont d'importance plus élevée dans l'environnement perçu et ont donc une influence plus conséquente sur le descripteur GIST. Il y a alors passage d'un environnement ouvert à un environnement très encombré. Ce type de segmentation permet de décrire à quelle distance (en terme de variation dans la structure extraite) le robot se trouve d'un endroit. Il s'agit d'une dépendance de l'algorithme au paramètre d'échelle. En effet, le descripteur GIST ne sera pas le même si le robot se trouve près d'un mur ou s'il se trouve au milieu de la pièce.

Des problèmes d'illumination sont apparus sur la partie en bas à gauche de la trajectoire. Ceci a engendré d'importantes variations d'intensité rendant parfois les images inexploitables. Toutefois, cela ne semble pas avoir eu un impact important sur la détection de changement de lieu. Les effets sont des ruptures supplémentaires et des ruptures sans localisation correcte aux alentours des portes. L'image 13201 sur la figure 8.3 est un exemple de ces problèmes d'illumination. Bien qu'il y ait des ruptures supplémentaires non significatives, elles sont relativement peu nombreuses par rapport à la quantité de problèmes d'illumination et la dégradation importante des images.

La robustesse de l'algorithme vis-à-vis du seuil de décision de rupture ν est illustrée par la figure 8.4. Les valeurs de seuil testées sont 0.6, 0.35 et 0.25.

- Les croix bleues sont les changements détectés pour les trois seuils.
- Les croix vertes sont les changements détectés pour les seuils de 0.35 et 0.25.
- Les croix noires sont les changements détectés pour le seuil 0.25.
- Les croix rouges quant à elles sont les changements détectés pour le seuil de 0.35 qui ont disparu avec le seuil de 0.25.

Aucun changement ne disparaît en passant du seuil de 0.6 au seuil de 0.35. Comme attendu d'après la théorie, plus le seuil est faible et plus l'algorithme détecte des changements de lieu. Très peu de changements de lieu disparaissent lorsque le seuil est abaissé. Changer la valeur du seuil influe seulement sur le nombre de changements de lieu détectés mais pas sur la localisation de ces derniers. Ce point révèle que l'algorithme est robuste à la localisation de ces changements qui sont vraiment caractéristiques de l'environnement. La disparition de certains changements de lieu provient du fait que

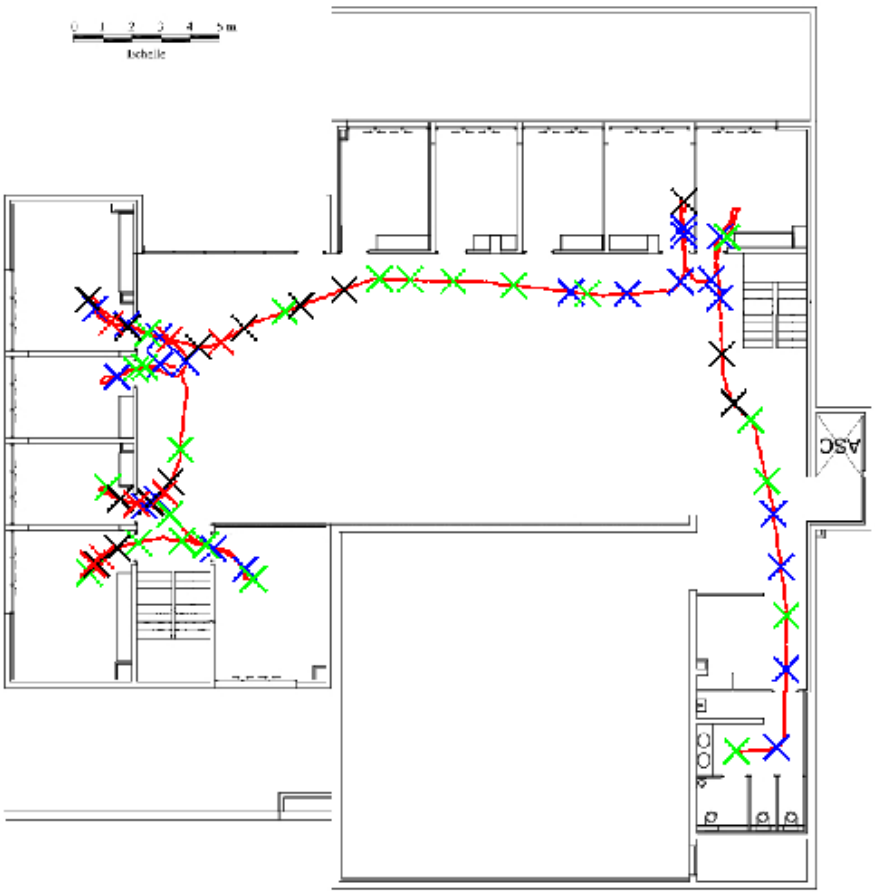


FIG. 8.4 – Analyse de la stabilité des lieux de coupures vis-à-vis du choix du seuil détection

seuls les changements dont au moins six dimensions convergent vers une détection sont considérés comme de vrais changements de lieu. Pour chacune des dimensions prise séparément, il est impossible d'avoir une variation de la localisation du changement de lieu vis-à-vis du choix du seuil. Durant la phase de convergence des six dimensions, une distance correspondant à la moitié de la fenêtre glissante est imposée entre deux changements de lieu consécutifs afin d'éviter la sursegmentation et la sensibilité au bruit. Si un changement est détecté en abaissant le seuil de décision et que sa distance avec le prochain changement obtenu avec un seuil plus élevé est inférieure à une demi fenêtre glissante, alors le changement obtenu avec le seuil plus élevé disparaît. Nous constatons sur la figure 8.4 qu'il y a toujours des croix noires proches des croix rouges. Ceci signifie que les changements détectés avec le seuil de 0.35 qui disparaissent sont remplacés par des changements très proches détectés avec le seuil de 0.25. Ces informations sont récapitulées dans le tableau 8.1.

Analyse en environnement extérieur

L'algorithme a aussi été testé dans un environnement d'extérieur pour valider notre définition de la place topologique. Comme déjà évoqué, les changements de lieu ne sont pas aussi bien définis que dans le cas des environnements d'intérieur. Une vue satellite de l'environnement de test est affichée sur la figure 8.5. Comme précédemment, la trajectoire suivie par le robot est superposée en rouge et les croix correspondent aux changements de lieu détectés. La trajectoire commence en bas de l'image et se termine en haut.

Un cas facilement identifiable de lieu topologique en environnement extérieur est le cas du lieu défini devant un bâtiment. Les images 109, 173 et 205 de la figure 8.5 montrent un exemple de ce cas. L'image 109 correspond à l'entrée dans le lieu en face du bâtiment et l'image 205 correspond à la sortie de ce lieu. Un lieu topologique pourrait être défini par ces deux changements mais un changement de lieu est détecté entre les deux (*cf.* l'image 173). Dans la première image (*i.e.* image 109), le lieu est principalement composé du bâtiment, d'un muret et de la végétation. Dans la deuxième image (*i.e.* image 173), le muret disparaît pour laisser place à une route. En termes de fréquences et d'orientations, le muret est composé d'orientations à 90°, représentant les lignes horizontales, de hautes fréquences du fait de ses angles bien marqués. La route quant à elle est composée de basses fréquences dans toutes les orientations. Le reste de l'environnement reste inchangé. Ces changements d'orientations et de fréquences définissent le changement de lieu.

Le passage du milieu naturel au milieu urbain, et réciproquement, est aussi un changement de lieu bien défini. Les images 347 et 409 représentent des changements de lieu à l'entrée et à la sortie d'un

Seuil	0.6	0.35	0.25
Nombre de ruptures détectées	22	51	58
Nombre de ruptures disparues	X	0	7
Nombre de ruptures communes	22		
	X	44	

Tab. 8.1 – Récapitulatif sur la stabilité des changements de lieu (ruptures de modèle) détectés vis-à-vis du seuil de décision. Le nombre de ruptures disparues correspond au nombre de ruptures détectées pour le seuil supérieur qui ont disparu avec le seuil inférieur. Ainsi, le chiffre 7 indique que 7 ruptures détectées avec le seuil 0.35 ne le sont pas avec le seuil de 0.25. Le nombre de ruptures communes correspond au nombre de ruptures détectées pour les différents seuils considérés.

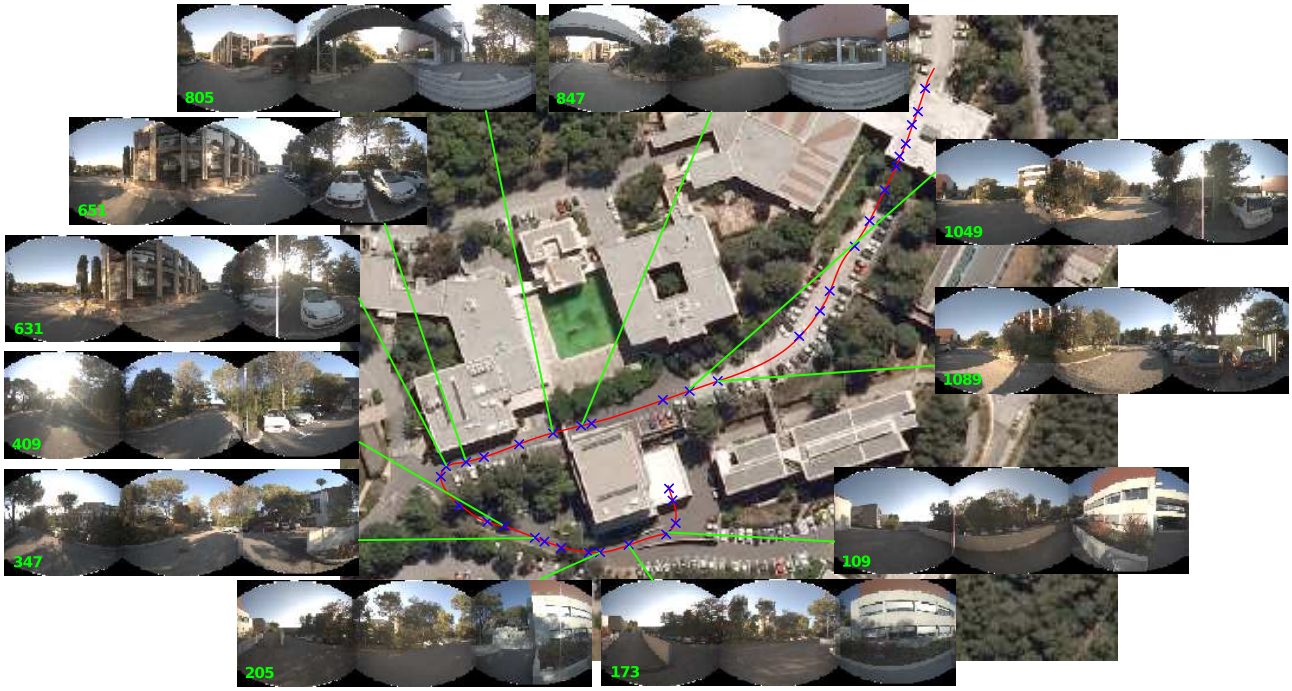


FIG. 8.5 – Résultats de la méthode de segmentation de l'environnement à base de GIST sur un milieu d'extérieur : le campus INRIA Sophia Antipolis

parking situé au milieu de la végétation (*cf.* vue satellite 8.5). Les routes d'accès constituent une petite portion de l'environnement avant et après le parking.

Le cas du pas de porte peut aussi être rencontré en environnement extérieur. Les images 805 et 847 sont des changements de lieu détectés avant et après être passé sous une passerelle. Il s'agit bien d'un cas très similaire. Les images traduisent des environnements très différents avant et après la passerelle.

Pendant l'expérimentation, d'importants éblouissements dus à la basse altitude du soleil dans le ciel ont généré des lignes verticales de lumière. Des exemples sont visibles sur les images 409, 631 et 1049 sur la figure 8.5. Ils renforcent parfois la probabilité de changement de lieu en ajoutant de l'évidence aux faibles détections de changement, c'est-à-dire des changements détectés dans moins de six dimensions. Ces éblouissements ajoutent des fortes fréquences horizontales dans les images mais leur impact sur la détection de changement de lieu est difficile à évaluer.

Dans les environnements d'extérieur, il est difficile d'évaluer l'exactitude des changements de lieu

détectés. Certains semblent intuitifs comme le cas de la passerelle et le cas où le robot passe devant un bâtiment. Les autres cas sont plus difficiles à justifier. Les images 631 et 651 sont deux changements de lieu détectés en face d'un bâtiment. Le premier est facile à justifier puisque le robot entre dans la zone devant le bâtiment. Le second n'est pas simple à justifier. De forts éblouissements ont été rencontrés dans cette partie de la trajectoire mais il n'est pas possible d'affirmer qu'elles sont responsables des changements de lieu détectés. Bien que non affiché, le changement de lieu suivant est aussi difficile à justifier.

Temps de calcul

L'algorithme est implémenté sous Matlab sans optimisation particulière. Le calcul du descripteur GIST prend environ 80ms. La détection de changement de lieu prend moins de 10ms la plupart du temps. Quelques pics sont observés à 15ms. Le processus complet prend donc environ 100ms pour détecter des changements de lieu dans l'environnement. Les images peuvent être traitées jusqu'à 10Hz, ce qui est suffisant pour un robot se déplaçant lentement. Une fréquence élevée d'acquisition n'engendrerait que peu de déplacement entre les différentes images. Une implémentation C++ permettrait d'avoir un calcul temps-réel très efficace utilisable pour la segmentation en ligne même pour des robots se déplaçant vite avec une fréquence d'acquisition de la caméra plus élevée.

8.2.2 Deuxième approche : utilisation des harmoniques sphériques

Pour cette deuxième approche, un environnement d'intérieur, le même que précédemment (*i.e.* le bâtiment Kahn), a d'abord été testé. Comme précédemment, l'avantage d'un environnement d'intérieur est de pouvoir comparer facilement les lieux intuitivement trouvés par l'homme avec ceux obtenus par l'algorithme. Dans cette nouvelle expérimentation, un test supplémentaire de la cohérence dans la détection de changement de lieu est ajouté. Il s'agit de déterminer si l'algorithme est capable de trouver un changement de lieu au même endroit lorsqu'il revient dans un environnement déjà visité. Le test est ici plus approfondi que la simple vérification au niveau des portes en entrant et sortant d'un bureau.

Ensuite, un test en environnement extérieur, le même que pour la première méthode (*i.e.* le campus de l'INRIA Sophia Antipolis) a été effectué. L'objectif de cette expérimentation est, comme

précédemment, de montrer la validité de la définition et de l'algorithme de segmentation pour des environnements complexes d'intérieur et d'extérieur.

Analyse en environnement intérieur - Trajectoire incomplète

Dans un premier temps, la même trajectoire que pour la première expérience est utilisée afin de valider l'approche et d'analyser le comportement de l'algorithme et les résultats fournis. Le figure 8.6 permet de visualiser la trajectoire suivie par le robot superposée sur la carte du bâtiment. Les croix rouges correspondent aux changements de lieu détectés par l'algorithme.

Avant de faire une analyse détaillée, il est intéressant de noter que chacun des changements de lieu détectés correspond à une variation importante dans la structure de l'environnement. En effet, ils se situent tous aux endroits de portes et de changements de volume d'une pièce (c'est-à-dire passer d'un recoin à un endroit plus large d'une pièce mais sans séparation avec une porte). Le parcours au milieu d'un grand espace est, quant à lui, peu segmenté. De plus, la découpe de l'environnement n'est pas sursegmentée contrairement à ce que nous obtenions avec la première approche.

Comme précédemment, la trajectoire du robot commence en bas à droite. Un changement de lieu important n'est pas détecté par l'algorithme au changement de pièce. Le problème ne provient pas de l'algorithme lui même, le changement apparaissant dans le signal. Par contre, un filtre permettant de lisser le signal (*cf.* section 7.3.1) a été ajouté afin de supprimer les perturbations et de conserver les variations significatives. Ce filtre induit un retard de détection au début de la trajectoire du fait d'un nombre insuffisant d'échantillons pour l'estimation de la valeur filtrée du signal. Ce changement est donc perdu du fait de ce filtre.

Le cas le plus classique pour valider un mécanisme de détection de changement de lieu est le cas du pas de porte. Il est illustré par les images 2680, 4326, 5328, 10455, 11954 et 12322. L'algorithme est donc capable de détecter le passage d'une pièce à une autre lorsque celles-ci sont séparées par une porte. Contrairement à notre première approche, il n'existe qu'un seul changement de lieu au niveau des portes. Le pas de porte constitue alors le changement de lieu lui-même à la place d'être considéré

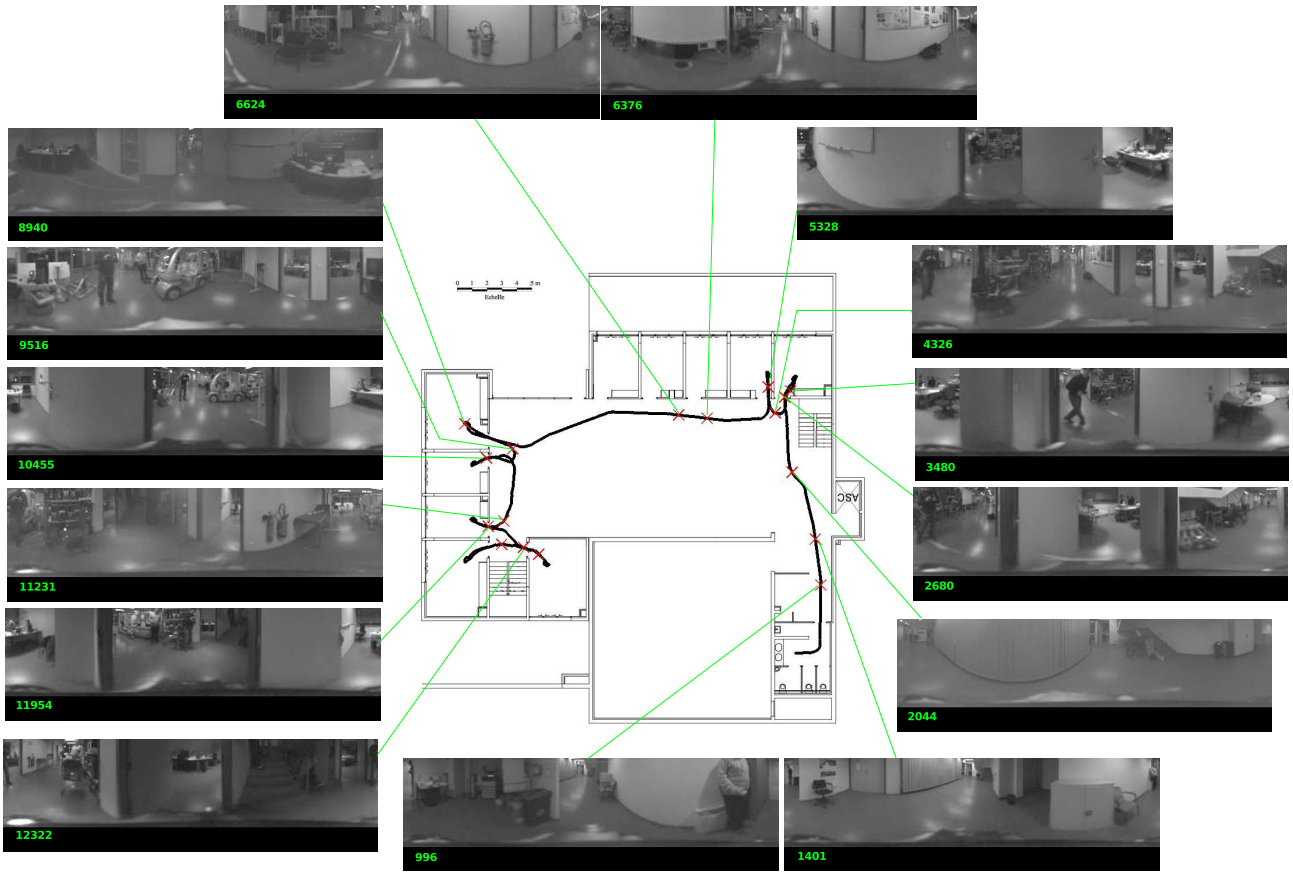


FIG. 8.6 – Résultats de détection de changement de lieu obtenus avec la méthode basée sur les harmoniques sphériques.

comme un lieu topologique particulier. Bien que la précédente approche avec deux changements de lieu aux alentours des portes se comprenne et se justifie, une approche contenant seulement un changement de lieu aux positions des portes est préférable. Le résultat est plus intuitif et l'environnement est moins segmenté.

Les exemples de changements de lieu des images 996, 1401 et 2044 correspondent au cas du changement détecté par variation de l'espace environnant. L'image 996 dénote la présence d'un mur sur la gauche. Il s'agit d'un cas pour ainsi dire similaire au cas du pas de porte du fait du rétrécissement. Toutefois, ce dernier est bien moins important que dans le cas du pas de porte. L'image 1401 montre le passage d'un endroit relativement restreint à un lieu plutôt ouvert. Il est à noter qu'à ce niveau, le recoin contenant la cage d'ascenseur entre dans le champ de vision. L'image 2044 est encore un cas différent. Il s'agit cette fois de quitter un endroit ouvert pour entrer dans un environnement plus étroit

de type couloir. Ceci n'est pas indiqué sur le plan de masse du bâtiment mais un rideau est présent et fait la séparation avec la partie gauche. Ce rideau est visible sur l'image de changement de lieu. D'ailleurs, nous observons que le robot entre dans un endroit plus étroit et plus encombré compris entre le rideau à gauche, un escalier à droite et une porte de bureau en face. Le cas des images 6376 et 6624 est assez similaire à ce dernier cas. Un grand panneau sépare la partie dans laquelle circule de robot de ce qui est plus en bas, constituant ainsi à nouveau un environnement de type couloir. Le panneau est visible dans l'image 6376 à gauche. Ces changements de lieu proviennent d'un changement des fréquences constituant l'image. Le robot passe d'un environnement contenant principalement un panneau blanc et des murs blancs à un environnement contenant une multitude d'objets. Le changement de lieu dépend du contenu de l'environnement mais les objets sont constitutifs de la structure de l'environnement, à moins de pouvoir les supprimer par filtrage. Ils ont d'autant plus d'importance qu'ils constituent une part importante de l'image. Toutefois, il est intéressant de noter l'influence de ceux-ci sur la segmentation de l'environnement car un lieu est décrit par sa structure mais aussi par son contenu. Les deux éléments sont liés car le type d'objet observé dépend du type d'environnement dans lequel le robot se trouve.

Un paramètre important est la capacité de l'algorithme à trouver les changements de lieu aux mêmes endroits lorsque le robot revient dans des lieux déjà visités. Ce point est étudié plus profondément ci-après mais une première approche est visible ici avec les allers-retours dans les bureaux. Les images 3480 et 4326 sont un exemple montrant un comportement souhaitable de l'algorithme. Les deux changements de lieu sont détectés très proches et au niveau du pas de porte. Toutefois, nous remarquons que globalement, nous n'obtenons qu'un seul changement de lieu lors des allers-retours dans les bureaux. C'est le cas des images 5328, 10455 et 11954. Il est à noter que le robot n'entre pas complètement au milieu du bureau, notamment du fait de la présence de mobilier, engendrant ainsi une estimation incomplète de la nouvelle pièce. Ces changements de lieu sont souvent détectés lorsque le robot entre dans le bureau mais pas lorsqu'il en sort. La distance parcourue par le robot ne permet pas d'avoir suffisamment d'échantillons de la nouvelle pièce pour une estimation correcte de la distribution de probabilité. Les échantillons extraits du bureau sont alors mélangés avec les échantillons provenant de la pièce suivante. Ces derniers prenant peu à peu une importance prépondérante, le changement de lieu n'est pas détecté. Ce problème entre dans le cadre des problèmes liés à l'estimation de distributions de probabilité (*cf.* section 7.3.2).

Le cas de l'image 8940 est un peu particulier. En effet, ce changement de lieu n'est pas significatif. D'autre part, les changements de lieu significatifs au niveau du pas de porte ne sont pas détectés. Malgré une relative invariance aux changements d'illumination, les fortes variations engendrent d'importants problèmes d'estimation. Les effets d'illumination sont particulièrement importants dans cette zone de la trajectoire comme le dénote l'image 8940. Ces effets sont certainement responsables des erreurs de détection dans cet endroit de la carte. En comparaison, l'image 12322 montre une détection de changement de lieu réussie malgré les variations de luminosité présentes dans cette zone.

Les images 9516 et 11231 quant à elles définissent le couloir sur la partie gauche de la trajectoire, devant les bureaux. Il n'est pas noté sur le plan du bâtiment car il est défini par l'encombrement de la zone et notamment la présence des Cycabs. Cette zone est intéressante car elle définit une transition entre l'espace ouvert du hall robotique, avant l'image 9516, et un couloir. Le couloir se termine lorsque le robot approche du bout du couloir avec une zone plus restreinte donnant accès à deux bureaux et un escalier, image 11231. La localisation de ce changement de lieu n'est pas tout à fait exacte du fait du passage dans un bureau avant d'aller au fond du couloir et du fait des effets de lumière présents dans cette partie de la trajectoire.

Analyse en environnement intérieur - Trajectoire complète

La trajectoire complète suivie par le robot dans le bâtiment est désormais considérée. Le plan du bâtiment avec la trajectoire complète superposée est montré sur la figure 8.7. Pour plus de clarté, la trajectoire de retour du robot est affichée en vert. Les changements de lieu détectés sur le retour sont représentés par des croix bleues. Le changement de lieu correspondant à l'image 14086 est représenté par une croix bleue mais sur une trajectoire noire. Ce phénomène est dû au nombre d'échantillons nécessaires à l'algorithme pour pouvoir détecter un changement de lieu. Le début de la trajectoire de retour permet de déterminer l'existence d'un changement de lieu à la fin de la trajectoire d'aller.

L'avantage d'avoir une trajectoire d'aller-retour est qu'elle permet d'étudier le comportement de l'algorithme lors de la détection de changement de lieu dans un environnement déjà étudié. L'objectif

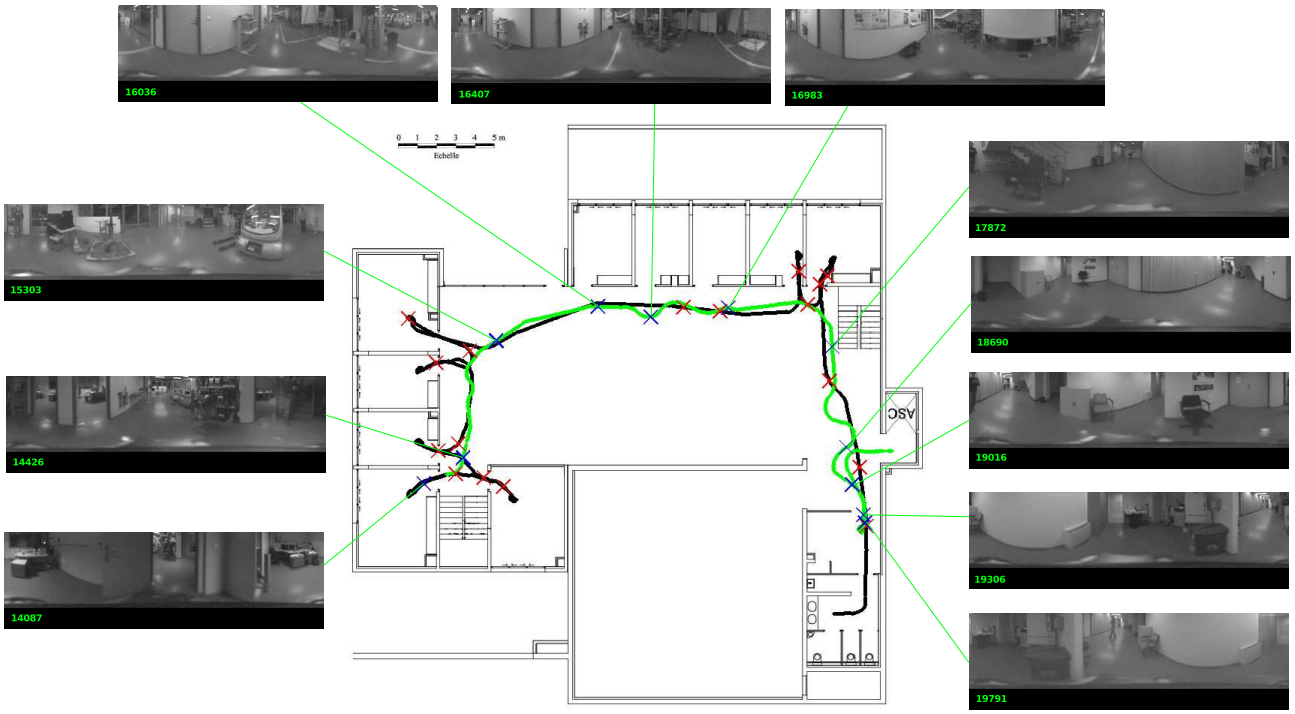


FIG. 8.7 – Résultats de détection de changement de lieu avec la méthode basée sur les harmoniques sphériques. La trajectoire présente un aller-retour afin d'étudier la stabilité des changements de lieu vis-à-vis de l'environnement. Le trajet aller est représenté en noir avec les changements de lieu matérialisés par des croix rouges. Le trajet de retour est en vert avec les changements de lieu matérialisés par des croix bleues.

est d'analyser la stabilité de l'algorithme vis-à-vis de l'environnement. Idéalement, lors du trajet de retour, l'algorithme doit trouver des changements de lieu aux mêmes emplacements que lors du trajet d'aller. Les couples d'images (14426, 11231), (15303, 9516), (16983, 6376), (19016, 1401) et le triplet d'images (19306, 19791, 996) sont des changements de lieu qui sont détectés au même endroit à l'aller et au retour. En terme de données quantitatives, l'analyse de ces changements de lieu donne :

- 28 changements de lieu sur la trajectoire complète
- 10 changements de lieu qui doivent se situer au même endroit d'un point de vue vérité terrain
- 7 changements de lieu détectés sur les 10 qui doivent se situer au même endroit
- 2 changements de lieu détectés sur la trajectoire de retour mais pas sur la trajectoire d'aller

Les localisations des changements de lieu entre l'aller et le retour ne sont pas forcément identiques mais sont très proches en terme de distance. Visuellement les images appartenant à un couple ou triplet sont très similaires ; le contenu est facilement reconnaissable entre les images. Ces changements de lieu dénotent un bon comportement de l'algorithme avec de bons résultats. Il serait souhaitable

que les changements de lieu identiques soient encore plus proches afin de déterminer précisément la localisation du changement au lieu d'avoir une zone de changement entre deux lieux. Toutefois, il est facilement concevable que la séparation entre deux pièces non délimitées par une porte est assez floue. De ce fait, la localisation du changement de lieu n'est pas très précise. D'autre part, bien que l'algorithme présente l'avantage d'être indépendant à l'orientation du robot lorsque celui-ci se déplace dans le plan, la localisation du changement de lieu dépendra quand même du sens dans lequel le robot traverse les deux pièces. L'estimation de l'environnement n'est pas tout à fait la même du fait d'une perception différente de l'environnement. Des éléments importants de l'environnement, apparaissant tardivement dans l'estimation des distributions de l'environnement lors du passage du robot dans un sens, apparaîtront assez tôt lors du passage en sens inverse, augmentant fortement la chance d'avoir un changement de lieu. L'algorithme peut donc être amené à trouver le changement de lieu plus tôt créant ainsi un léger décalage de localisation avec la précédente localisation de changement de lieu. Cet effet est moins marqué lorsque les pièces sont séparées par une porte du fait d'un changement très net au niveau du pas de la porte. Les couples d'images (16407, 6624) et (17872, 2044) sont des exemples où l'écart entre les deux changements de lieu est assez élevé mais pourtant dénotent le même changement de lieu. Le premier cas suit une courbure de la trajectoire du robot vers un endroit un peu encombré avant de rejoindre le couloir précédemment défini. Le deuxième cas est le passage du couloir délimité par l'escalier et le rideau à un espace plus ouvert. Étant assez proche de l'escalier, le fait de s'approcher de l'espace ouvert rend le changement dans l'estimation de la distribution des paramètres de l'environnement assez brutal, bien plus que dans le cas inverse.

L'image 16036 est un changement de lieu détecté uniquement sur le retour. Il y a ici un problème de cohérence sur l'estimation de changement de lieu entre l'aller et le retour. Il semblerait que ce soit une fausse détection de l'algorithme car il n'existe pas de changement marquant de structure de l'environnement à cet endroit. Le changement important a été trouvé aux alentours des images (16407, 6624).

Le dernier cas est le changement de lieu illustré par l'image 18690. Ce cas est très intéressant car il montre le comportement de l'algorithme lorsque le robot s'approche des murs ou d'autres objets. En fait, ce changement de lieu pourrait être groupé avec les changements de lieu des images 19016 et 1401 et ils constitueraient une localisation floue du changement de lieu. Dans le cas du trajet de retour, le robot fait une excursion sur la gauche. L'explication la plus probable est que la partie de

l'excursion sur la gauche constitue un lieu du fait du rapprochement du mur et donc d'une estimation de la structure différente. Le robot sortirait brièvement du lieu dans lequel il se trouve. Comme déjà évoqué dans la précédente approche, ces méthodes ne sont pas invariantes au phénomène d'échelle. Si le robot se rapproche d'un mur, l'estimation de l'environnement sera forcément différente de celle faite s'il reste au milieu de la pièce. La segmentation en pièce se fait donc lorsque la structure de l'environnement change entre deux pièces mais aussi lorsque le robot se rapproche d'un mur. Ceci définit un effet d'éloignement par rapport aux éléments constitutifs de l'environnement.

Il est intéressant de noter que le robot est capable de faire des oscillations au sein d'un lieu sans que l'algorithme ne détecte de changement de lieu. L'invariance de l'algorithme vis-à-vis d'une rotation autour de l'axe z est ainsi vérifiée.

Analyse en environnement extérieur

Comme pour l'approche précédente, la méthode utilisant les harmoniques sphériques a été testée dans un environnement d'extérieur. Une vue satellite de l'environnement de test est affichée sur la figure 8.8. Comme précédemment, la trajectoire suivie par le robot est superposée en rouge et les croix correspondent aux changements de lieu détectés. La trajectoire commence en bas de l'image et se termine en haut.

Cette expérimentation est particulièrement intéressante car tous les changements de lieu détectés sont compréhensibles et, de plus, dans un environnement d'extérieur. En effet, chacune des ruptures s'explique aisément étant donnée la définition du lieu topologique établie. En regroupant par type de rupture (ou lieu topologique défini entre deux ruptures), l'analyse des résultats (*cf.* figure 8.8) donne :

– Le cas du lieu défini devant un bâtiment est illustré par les couples d'images (139,229), (630, 842) et (842, 954). Comme précédemment, il s'agit d'un lieu défini par le changement de lieu lorsque le robot entre dans la zone devant le bâtiment et par le changement de lieu lorsque le robot quitte cette zone. Les exemples de ce cas sont très significatifs dans cette expérimentation. Le premier couple correspond au même lieu défini lors de l'expérimentation avec le descripteur GIST modifié. Par contre le couple d'images (630, 842) définit très correctement un lieu devant une façade de bâtiment. Dans l'expérimentation précédente, ce lieu était sursegmenté en lieux

non significatifs. Enfin, l'intérêt du dernier couple d'images est qu'il montre une fois de plus la définition correcte du lieu et la différence avec la méthode précédente. En effet, cette fois, le lieu est complètement déterminé devant le bâtiment tandis que dans l'expérimentation précédente, la détection de changement de lieu à la sortie n'avait pas été détectée. L'image 842 correspond au changement de lieu défini lorsque le robot passe sous une passerelle. Ce cas est toujours assimilable au cas du pas de porte.

– Le changement de lieu, très significatif, défini par un passage nature/environnement urbain ou inversement est illustré par les images 325 et 403. Ces changements avaient aussi été détectés par la première méthode mais les lieux étaient moins bien définis en raison de la sursegmentation.
– Le cas du couple d'images (1035, 1450) est un cas très intéressant car il permet de mettre en exergue plusieurs avantages de l'algorithme. Tout d'abord, le lieu défini est très significatif puisqu'il s'agit d'un parking. De plus, les changements de lieu sont bien positionnés à l'entrée et à la sortie du parking. Il est intéressant de noter la possibilité de rotation du robot sans engendrer de rupture de modèle. Ceci permet de confirmer, comme pour l'environnement d'intérieur, l'indépendance du spectre d'harmoniques sphériques à la rotation d'axe z du robot. D'autre part, d'après la définition du lieu topologique établie, il doit être possible de créer un lieu topologique indépendamment de la covisibilité des caractéristiques du lieu (*cf.* section 6.1). Dans le cas de ce parking, aucune information contenue dans l'image 1035 n'est visible dans l'image 1450. La définition de ce lieu permet de valider l'indépendance à la covisibilité. Le lieu est défini uniquement par ses caractéristiques propres : les paramètres structurels.

Seuls les changements de lieu et lieux les plus significatifs ont été exposés. Toutefois, les autres lieux définis par des combinaisons différentes de changements de lieu consécutifs sont aussi très significatifs dans le cadre de cette expérimentation. Par exemple, les lieux définis par les couples d'images (229, 325) et (403, 630) représentent des environnements de nature et se distinguent parfaitement des environnements urbains qui les entourent.

Un aspect négatif est toutefois à remarquer. Comme pour l'expérimentation précédente, le filtre de lissage entraîne un retard d'estimation empêchant la détection des ruptures de modèle au début de l'expérimentation. En effet, le robot commençant sa trajectoire dans le bâtiment, il aurait été logique d'avoir une rupture au passage de la porte ; lorsque le robot quitte l'environnement d'intérieur pour l'environnement d'extérieur.

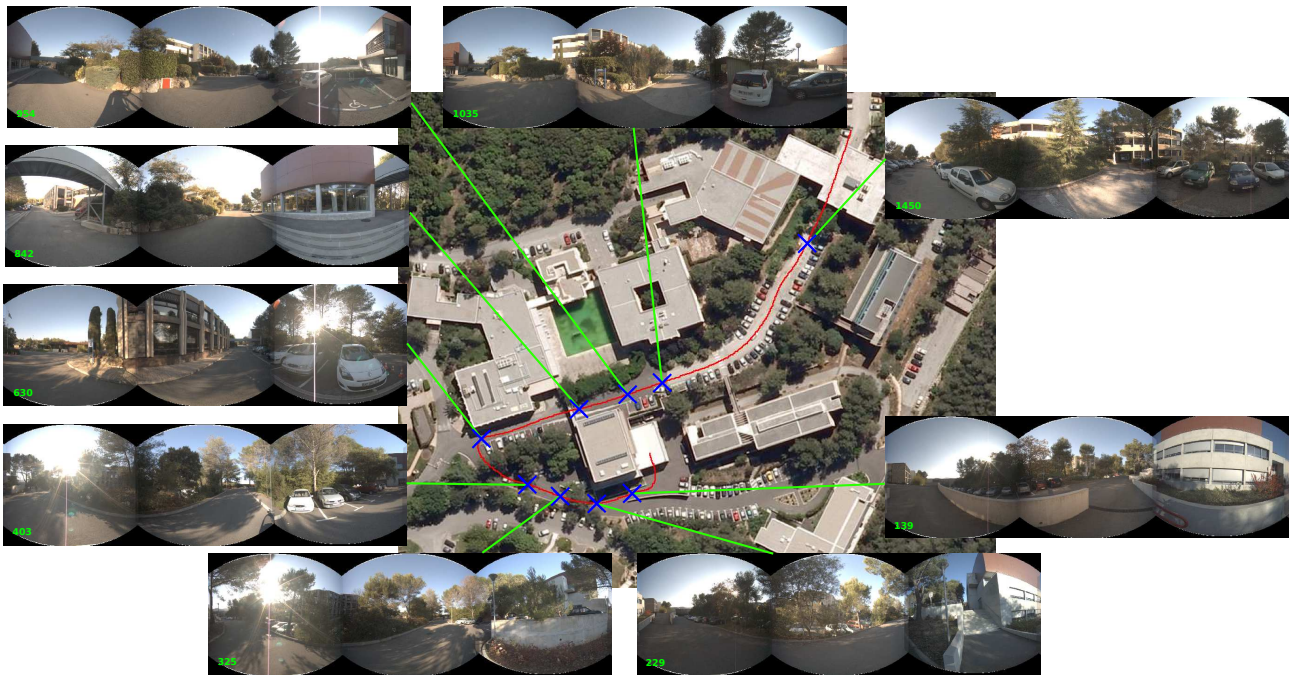


FIG. 8.8 – Résultats de détection de changement de lieu en environnement extérieur avec la méthode basée sur les harmoniques sphériques. Les changements de lieu sont matérialisés par les croix bleues.

Temps de calcul

Le dernier élément à considérer est le temps de calcul de l'algorithme. L'approche basée sur les harmoniques sphériques est codée en Matlab sans optimisation ni parallélisation du code. La seule optimisation utilisée est le calcul particulier des harmoniques sphériques expliqué dans la partie 6.3.2. En ce qui concerne l'échantillonnage de la sphère, 62500 points résultant d'une distribution uniforme bidimensionnelle de 250 points suivant les angles de gisement et d'élévation sont utilisés. Un calcul de coefficients pour chacun des points est nécessaire pour l'estimation des harmoniques sphériques. Ce calcul prend environ une minute mais présente l'avantage de ne devoir être effectué qu'une seule fois. Il sera donc effectué dans une phase d'initialisation et n'impactera pas le calcul lors de la détection de changement de lieu à chaque image. Lors du déplacement du robot, il faut, pour chaque nouvelle image acquise, calculer le spectre et déterminer s'il y a changement de lieu ou non. Le calcul du spectre se fait en environ 290 ms tandis que le calcul de détection ne requiert que 10 ms. Le processus de détection de changement de lieu pour chaque nouvelle image acquise prend alors en moyenne 300 ms permettant ainsi une acquisition à 3.3 Hz. Toutefois, le code est hautement parallélisable pour le calcul du spectre. Une implémentation en C/C++ avec une parallélisation devrait accroître considérablement

les capacités de calcul de l'algorithme.

8.3 Discussion

Dans ce chapitre nous avons exposé nos résultats et analyses pour nos deux méthodes de détection de changement de lieu. La première montre des résultats intéressants mais présente divers inconvénients :

- Sursegmentation de l'environnement
- Changements de lieu pas toujours significatifs
- Hypothèse forte d'indépendance des dimensions du descripteur
- Système de convergence empirique d'au moins six dimensions pour la prise de décision (règle heuristique)
- Calcul du GIST pas complètement adapté à la sphère

Malgré ces défauts, ce premier algorithme présente de nombreux avantages dont le plus important est la validation de la preuve de concept quant à l'utilisation de l'information structurelle de l'environnement. Les différents avantages de cet algorithme sont alors :

- Définition du lieu topologique bien établie
- Preuve de concept de l'utilisation de la structure de l'environnement
- Algorithme temps-réel
- Utilisation d'un descripteur global de très faible dimension
- Résultats prometteurs pour les environnements d'intérieur et d'extérieur

Suite à cette première approche, nous avons élaboré une seconde approche basée sur le calcul d'harmoniques sphériques pour la description de la structure de l'environnement. Cette deuxième méthode est mieux formulée et prend bien en compte les différents aspects telles la représentation sphérique et la dépendance entre les dimensions du descripteur. Les résultats obtenus sont largement supérieurs à ceux obtenus lors de la preuve de concept et concrétisent l'amélioration de l'algorithme. Malgré quelques inconvénients, nous obtenons un système efficace de détection de changement de lieu ne sursegmentant pas l'environnement. Les lieux obtenus sont significatifs et correspondent à l'intuition que nous pouvons avoir de la segmentation de l'environnement. Les inconvénients de cette nouvelle approche sont essentiellement :

- Une relative dépendance aux variations de luminosité

– Descripteur indépendant seulement à la rotation suivant l'axe z

Toutefois, ces inconvénients étaient déjà présents pour l'approche utilisant le GIST. Ces problèmes sont abordés seulement dans cette deuxième approche une fois la majorité des problèmes évoqués pour la première approche résolus. En ce qui concerne les avantages de cette nouvelle approche, ils sont nombreux :

– Définition du lieu topologique bien établie
– Pas de sursegmentation de l'environnement
– Changements de lieu significatifs
– Mécanisme de détection de rupture bien défini : prise en compte de la corrélation entre les dimensions du descripteur
– Décision de l'algorithme sur un signal unidimensionnel très peu bruité. Les pics sont très significatifs
– Descripteur global sphérique bien adapté à la représentation sphérique
– Algorithme temps-réel
– Utilisation d'un descripteur global de très faible dimension
– Algorithme hautement parallélisable permettant une implémentation très efficace en terme de temps de calcul

Il reste un élément à approfondir, il s'agit de la dépendance au facteur d'échelle de l'algorithme. Bien que cette dépendance permette de définir une notion de distance au sein d'un lieu vis-à-vis des limites physiques du lieu, elle engendre par là même de nouveaux lieux en fonction de la proximité. Il serait nécessaire de supprimer cette dépendance afin de pouvoir définir un lieu topologique dont les limites correspondent aux limites physiques. Nous éviterions alors la création de lieux supplémentaires au fur et à mesure que le robot s'approche d'une structure.

Conclusion et perspectives

1 Conclusion

Ce manuscrit présente nos contributions dans le contexte du SLAM topologique. Celles-ci se focalisent sur deux algorithmes fondamentaux pour la création de cartes topologiques consistantes. La première contribution concerne l'algorithme crucial de la détection visuelle de fermeture de boucle. La deuxième traite de la segmentation de l'environnement en lieux topologiques.

L'algorithme de détection visuelle de fermeture de boucle développé repose sur l'utilisation efficace de la représentation sphérique de l'environnement. Basée sur une approche d'inférence bayésienne présentant d'excellentes qualités de calcul temps-réel, d'implémentation incrémentale et de robustesse à l'aliasing perceptuel, nous avons élaboré une méthode permettant de rendre la détection invariante aux conditions de prise de vue. Il en résulte un algorithme extrêmement efficace de détection de fermeture de boucle ne contraignant pas la trajectoire du robot lors de la navigation, ou exploration, dans l'environnement. D'autre part, la méthode est purement qualitative et repose donc exclusivement sur l'utilisation de distributions de probabilité. Il en résulte un excellente robustesse à l'erreur d'estimation. Ainsi, aucune vérification de consistance de transformation géométrique entre les vues de fermeture de boucle n'est nécessaire. Le mécanisme de mise à jour des hypothèses de fermeture de boucle utilise une modélisation efficace de répartition de l'information dans l'environnement, en exploitant les propriétés de la représentation sphérique, renforçant la consistance de la détection de fermeture de boucle.

L'algorithme de segmentation est un mécanisme de partitionnement de l'environnement en lieux topologiques significatifs. Afin de bien établir l'algorithme, nous avons élaboré une définition générique du lieu topologique basée sur la structure de l'environnement. Une première approche a été réalisée en utilisant le descripteur global GIST, adapté à la représentation sphérique, pour décrire la structure de l'environnement. Un algorithme de détection de rupture de modèle appliqué sur la variation du

descripteur GIST permet de détecter les changements de lieu. Cette première approche est une preuve de concept réussie fournissant des résultats satisfaisants à la fois pour des environnements d'intérieur et d'extérieur. Afin de pallier les défauts de la première approche, une seconde approche a été développée. Celle-ci utilise comme descripteur de l'environnement le spectre d'harmoniques sphériques qui présente l'avantage d'être parfaitement adapté à la sphère. Ensuite, l'algorithme de détection de rupture de modèle est corrigé pour prendre en compte l'interdépendance des signaux constitués par les dimensions du descripteur de structure. Il en résulte un algorithme très efficace présentant une segmentation de l'environnement très satisfaisante. L'algorithme présente de plus de bonnes propriétés comme une localisation précise des changements de lieux, validée par la segmentation lors de la revisite du robot dans un environnement précédemment segmenté.

2 Perspectives

Les algorithmes proposés présentent de bons résultats mais certains éléments sont encore à améliorer. La première partie des perspectives concerne les améliorations possibles de l'algorithme de détection de fermeture de boucle. Le seconde partie concerne les améliorations de la détection de changement de lieu. Enfin, la dernière partie concerne les extensions possibles des algorithmes et leur inclusion dans un algorithme plus large tel que le SLAM topologique ou encore l'ajout de la notion de sémantique.

2.1 Amélioration de la détection visuelle de fermeture de boucle

Robustesse de l'algorithme

L'algorithme de détection visuelle de fermeture de boucle développé présente déjà des résultats très satisfaisants. Toutefois, les expérimentations ont mis en évidence qu'il est susceptible de générer des fausses fermetures de boucle. Il est donc nécessaire d'accroître la robustesse de l'algorithme face à l'aliasing perceptuel pour le rendre fiable. Aucune fausse alarme ne doit subsister pour que l'algorithme puisse être utilisé efficacement dans des algorithmes de navigation ou de SLAM. L'objectif est cependant de conserver la même approche qualitative, c'est à dire sans réinsérer un mécanisme de vérification de consistance géométrique.

Gestion des expérimentations à long terme

En ce qui concerne la robustesse de l'algorithme dans les expérimentations à long terme (*cf.* section 1.2.5), il est capable de gérer des environnements différents du fait de l'utilisation d'un dictionnaire dynamique. Toutefois, il ne possède pas réellement de mécanisme de gestion des variations de l'environnement. Il en résulte une possibilité de détection de fermeture de boucle mais elle n'est pas garantie. À l'échelle des saisons, il n'est pas garanti qu'un même endroit soit reconnu entre un premier apprentissage en été et une revisite en hiver. D'autre part, un environnement changeant entraîne une croissance importante de la taille du dictionnaire, tout en conservant des mots visuels pas forcément significatifs. Une amélioration de l'algorithme présenté serait alors d'ajouter ces mécanismes pour obtenir une robustesse face aux variations de l'environnement. Une solution serait la méthode développée par [Dayoub et Duckett, 2008]. Elle offre de bons résultats et permet de gérer la taille du dictionnaire en ne conservant que les mots visuels les plus pertinents.

Expérimentations avec des drones

L'objectif de l'algorithme de détection de fermeture de boucle élaboré est de fournir une solution permettant de rendre la détection indépendante de l'orientation du robot. De plus, la solution proposée fournit une méthode générale applicable à n'importe quel type de robot dans n'importe quel type d'environnement. Sans être une amélioration, conduire des expérimentations avec un drone permettrait de compléter l'analyse de l'indépendance de la détection à l'orientation du robot. Pour cela, le drone devra effectuer des manœuvres engendrant des rotations suivant les trois axes et ce aux endroits où des fermetures de boucle doivent être détectées.

2.2 Amélioration de la détection de changement de lieu

Invariance complète à l'orientation

Que ce soit le descripteur GIST modifié ou le spectre d'harmoniques sphériques, tous deux ne sont invariants qu'à une rotation d'axe z du robot. Ainsi, une rotation autour de l'axe x ou de l'axe y engendre une modification du descripteur de structure de l'environnement. L'application de la segmentation de l'environnement est donc limitée aux robots mobiles évoluant dans des zones planes ou des zones dont les dénivelés sont suffisamment faibles pour faire une hypothèse de planéité locale.

L'algorithme n'est alors pas utilisable pour des drones lors de la cartographie et la segmentation, par exemple, d'immeubles. Pour obtenir un algorithme complètement générique, et par conséquent utilisable avec n'importe quel type de robot dans n'importe quel type d'environnement, il est nécessaire de rendre le descripteur de structure indépendant des rotations du robot. Le GIST modifié repose sur l'utilisation de la transformée de Fourier 2D de l'image. De fait, ce descripteur présente de fortes limitations et ne sera pas adaptable à la représentation sphérique. En ce qui concerne le spectre d'harmoniques sphériques, il est bien adapté à la représentation. Il reste à le rendre indépendant aux rotations d'axe x et y. Il est possible à partir du spectre d'harmoniques sphériques d'une image et du spectre de la même image sur laquelle des rotations ont été effectuées de retrouver la transformation entre les deux images. Une solution pour rendre l'algorithme indépendant aux rotations du robot serait alors de ré-estimer le spectre de l'image courante en supprimant les rotations par rapport à l'image précédente. Une fois les rotations supprimées, la distribution de probabilité du lieu courant peut être estimée en ajoutant le dernier spectre obtenu. L'avantage de cette méthode est qu'elle apporte une solution simple pour rendre l'estimation de la structure de l'environnement invariante aux rotations du robot. Son inconvénient est que, de par son estimation des transformations, elle s'oppose à l'approche purement qualitative développée dans cette thèse.

Indépendance à l'effet d'échelle

Le problème de l'effet d'échelle a été abordé dans les expérimentations en environnement d'intérieur pour les descripteurs GIST modifié et spectre d'harmoniques sphériques. Il est résumé dans la conclusion sur la segmentation topologique de l'environnement (*cf.* section 8.3). Comme expliqué dans la conclusion, cette dépendance à l'échelle permet de définir une notion de distance vis-à-vis des limites physiques de l'environnement. L'objectif est cependant de créer des lieux topologiques dont les limites correspondent aux limites physiques. C'est à dire sans créer de nouveaux lieux au fur et à mesure que le robot se rapproche d'une structure comme par exemple un mur.

Robustesse aux effets d'illumination

Le descripteur de structure de l'environnement à base d'harmoniques sphériques est relativement robuste aux variations affines d'illumination. Il n'est toutefois pas robuste à tous les types de changements d'illumination. Comme évoqué dans la section 7.4, les travaux de [Friedrich *et al.*, 2008]

constituent une première approche pour améliorer la robustesse aux effets d'illumination. La méthode repose sur une analyse en composantes principales afin de supprimer les variations dans le spectre d'harmoniques sphériques dues à des changements d'illumination.

2.3 Extensions des algorithmes

Couplage des deux algorithmes

L'objectif est de coupler le système de détection de fermeture de boucle avec celui de détection de changement de lieu. Un tel couplage serait bénéfique dans le processus de décision de chacun des deux algorithmes. Le système de détection de fermeture de boucle pourrait bénéficier d'un a priori du lieu dans lequel se trouve le robot. L'information structurelle complémentaire permettrait ainsi de réduire les fausses fermetures de boucle avec des lieux bien distincts. En ce qui concerne le système de détection de changement de lieu, il pourrait bénéficier de la détection de fermeture de boucle à l'image près pour stabiliser la localisation du changement de lieu lors du passage dans un endroit déjà visité.

SLAM Topologique

Les deux algorithmes couplés sont destinés à être inclus dans un système de SLAM topologique. Cela permettra de clairement définir les lieux représentant les nœuds du graphe d'accès qui constituent la carte. Un nœud sera défini comme un lieu constitué d'un ensemble d'images plutôt que d'avoir chaque image associée à un nœud. Le système obtenu aura alors un niveau d'abstraction supplémentaire vis-à-vis de l'information bas niveau. La carte sera alors beaucoup plus significative et, du fait des performances de la détection de fermeture de boucle, non contrainte par la trajectoire suivie par le robot lors de la phase d'exploration.

Sémantique de l'environnement

La représentation de l'environnement développée pourrait ensuite servir pour apprendre et reconnaître des catégories sémantiques. En environnement intérieur, ces catégories seraient : couloir, chambre, cuisine, salon, bureau, ... En environnement extérieur, elles seraient : route, croisement, espace ouvert, parking, ... Pour cela, il est alors nécessaire de réaliser de la classification à partir du spectre d'harmoniques sphériques, descripteur de la structure de l'environnement dont la dimension

est très faible. Chaque classe, ou catégorie sémantique, serait caractérisée par une distribution de probabilité de spectres d'harmoniques sphériques.

Annexes

Annexe A

Détermination de l'expression du filtre de Gabor dans le domaine fréquentiel

L'équation du filtre de Gabor dans le domaine spatial, exprimée en coordonnées cartésiennes, est la suivante :

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\left(\frac{x-x_0}{\sigma_x}\right)^2+\left(\frac{y-y_0}{\sigma_y}\right)^2\right)} e^{j2\pi(u_0x+v_0y)} \tag{A.1}$$

L'équation de la transformée de Fourier du filtre s'exprime alors par :

$$\begin{aligned} G(u,v) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy & \text{(A.2)} \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{1}{\sigma_x^2}(x-x_0)_r^2+\frac{1}{\sigma_y^2}(y-y_0)_r^2\right)} e^{j2\pi(u_0x+v_0y)} e^{-j2\pi(ux+vy)}dxdy & \text{(A.3)} \end{aligned}$$

Cette forme d'équation n'est pas évidente à calculer étant donné qu'il ne sera pas possible de séparer les variables x et y pour calculer deux intégrales indépendantes. L'équation en écriture vectorielle s'obtient en posant :

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \qquad U = \begin{bmatrix} u \\ v \end{bmatrix} \qquad R = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \frac{1}{\sigma_x} & 0 \\ 0 & \frac{1}{\sigma_y} \end{bmatrix} \tag{A.4}$$

La forme vectorielle équivalente est donc :

$$\begin{aligned} G(U) &= \frac{1}{2\pi\sigma_x\sigma_y}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}(\Sigma R(X-X_0))^T(\Sigma R(X-X_0))} e^{j2\pi U_0^T X} e^{-j2\pi U^T X}dX & \text{(A.5)} \\ &= \frac{1}{2\pi\sigma_x\sigma_y}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}(\Sigma R(X-X_0))^T(\Sigma R(X-X_0))} e^{-j2\pi(U-U_0)^T X}dX & \text{(A.6)} \end{aligned}$$

Soit le changement de variable permettant de simplifier l'expression de la gaussienne dans le calcul de la transformée de Fourier :

$$\tilde{X} = \Sigma R\,(X - X_0) \qquad d\tilde{X} = \Sigma R dX \qquad X = (\Sigma R)^{-1}\,\tilde{X} + X_0 \tag{A.7}$$

$$\begin{aligned} G(U) &= \frac{1}{2\pi\sigma_x\sigma_y||\Sigma R||}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left(\tilde{X}^T\tilde{X}\right)} e^{-j2\pi(U-U_0)^T\left((\Sigma R)^{-1}\tilde{X}+X_0\right)} d\tilde{X} && \text{(A.8)}\\ &= \frac{1}{2\pi} e^{-j2\pi(U-U_0)^T X_0}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left(\tilde{X}^T\tilde{X}\right)} e^{-j2\pi(U-U_0)^T(\Sigma R)^{-1}\tilde{X}} d\tilde{X} && \text{(A.9)}\\ &= \frac{1}{2\pi} e^{-j2\pi(U-U_0)^T X_0}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left(\tilde{X}^T\tilde{X}\right)} e^{-j2\pi\left((\Sigma R)^{-T}(U-U_0)\right)^T\tilde{X}} d\tilde{X} && \text{(A.10)} \end{aligned}$$

où $||.||$ est le déterminant de la matrice avec $||\Sigma R|| = ||\Sigma||.||R|| = \dfrac{1}{\sigma_x\sigma_y}$ car R est une matrice de rotation et $||R|| = 1$. Le calcul obtenu est alors celui de la transformée de Fourier d'une gaussienne centrée réduite multivariée et de variable fréquentielle $(\Sigma R)^{-T}\,(U - U_0)$. Avant de pouvoir déterminer complètement $G(U)$, il est donc nécessaire de calculer la transformée de Fourier d'une gaussienne multivariée. Soit le cas simple bivarié suivant :

$$\begin{aligned} Gauss(u,v) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left(x^2+y^2\right)} e^{-j2\pi(xu+yv)} dxdy && \text{(A.11)}\\ &= \int_{-\infty}^{+\infty} e^{-\frac{1}{2}x^2} e^{-j2\pi xu} dx \int_{-\infty}^{+\infty} e^{-\frac{1}{2}y^2} e^{-j2\pi yv} dy && \text{(A.12)}\\ &= \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left(x^2+j4\pi xu\right)} dx \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left(y^2+j4\pi yv\right)} dy && \text{(A.13)}\\ &= \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left((x+j2\pi u)^2+4\pi^2u^2\right)} dx \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left((y+j2\pi v)^2+4\pi^2v^2\right)} dy && \text{(A.14)}\\ &= e^{-2\pi^2\left(u^2+v^2\right)}\int_{-\infty}^{+\infty} e^{-\frac{1}{2}(x+j2\pi u)^2} dx \int_{-\infty}^{+\infty} e^{-\frac{1}{2}(y+j2\pi v)^2} dy && \text{(A.15)} \end{aligned}$$

En posant :

$$\tilde{x} = x + j2\pi u \qquad d\tilde{x} = dx \tag{A.16}$$

$$\tilde{y} = y + j2\pi v \qquad d\tilde{y} = dy \tag{A.17}$$

L'équation devient :

$$Gauss(u,v) = e^{-2\pi^2\left(u^2+v^2\right)} \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\tilde{x}^2} d\tilde{x} \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\tilde{y}^2} d\tilde{y} \tag{A.18}$$

Sachant que l'intégrale de Gauss donne $\int_{-\infty}^{+\infty} e^{-\frac{1}{2}x^2} dx = \sqrt{2\pi}$, la transformée de Fourier de la gaussienne bivariée est la suivante :

$$Gauss(u,v) = 2\pi e^{-2\pi^2\left(u^2+v^2\right)} \tag{A.19}$$

Par extension, la transformée de Fourier de la gaussienne multivariée de dimension k est :

$$Gauss(U) = (2\pi)^{k/2}\, e^{-2\pi^2 U^T U} \tag{A.20}$$

En reportant le résultat obtenu dans le calcul du filtre de Gabor, la transformée de Fourier a pour expression :

$$\begin{aligned} G(U) &= \frac{1}{2\pi} e^{-j2\pi(U-U_0)^T X_0} 2\pi e^{-2\pi^2\left((\Sigma R)^{-T}(U-U_0)\right)^T\left((\Sigma R)^{-T}(U-U_0)\right)} && \text{(A.21)} \\ &= e^{-j2\pi(U-U_0)^T X_0} e^{-2\pi^2\left(\Sigma^{-1}R(U-U_0)\right)^T\left(\Sigma^{-1}R(U-U_0)\right)} && \text{(A.22)} \end{aligned}$$

La dernière équation s'obtient du fait que Σ est une matrice diagonale et que R est une matrice de rotation. Soit :

$$\Sigma_U = \Sigma^{-1} = \begin{bmatrix} 2\pi\sigma_x & 0 \\ 0 & 2\pi\sigma_y \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma_u} & 0 \\ 0 & \frac{1}{\sigma_v} \end{bmatrix} \tag{A.23}$$

d'où :

$$G(U) = e^{-j2\pi(U-U_0)^T X_0} e^{-\frac{1}{2}(\Sigma_U R(U-U_0))^T(\Sigma_U R(U-U_0))} \tag{A.24}$$

En transposant dans l'écriture non vectorielle, l'équation du filtre de Gabor fréquentiel est :

$$G(u,v) = e^{-\frac{1}{2}\left(\frac{1}{\sigma_u^2}(u-u_0)_r^2 + \frac{1}{\sigma_v^2}(v-v_0)_r^2\right)} e^{-j2\pi((u-u_0)x_0+(v-v_0)y_0)} \tag{A.25}$$

Annexe B

Détermination de l'équation de la somme des ratios de vraisemblance logarithmiques dans le cas gaussien unidimensionnel

L'équation non simplifiée du calcul de la somme des ratios de vraisemblance logarithmiques appliquée au cas gaussien (*cf.* section 7.2.1) est :

$$S_\tau^n = \sum_{j=\tau}^{n} ln\left(\frac{f_1\left(y_j|\theta_1\right)}{f_0\left(y_j|\theta_0\right)}\right) \tag{B.1}$$

$$= \sum_{j=\tau}^{n} ln\left(\frac{e^{-\frac{1}{2}\frac{(y_j-\mu_1)^2}{\sigma^2}}}{e^{-\frac{1}{2}\frac{(y_j-\mu_0)^2}{\sigma^2}}}\right) \tag{B.2}$$

En développant la deuxième équation :

$$S_\tau^n = \sum_{j=\tau}^{n}\left(ln\left(e^{-\frac{1}{2}\frac{(y_j-\mu_1)^2}{\sigma^2}}\right) - ln\left(e^{-\frac{1}{2}\frac{(y_j-\mu_0)^2}{\sigma^2}}\right)\right) \tag{B.3}$$

$$= \sum_{j=\tau}^{n}\left(-\frac{1}{2}\frac{(y_j-\mu_1)^2}{\sigma^2} + \frac{1}{2}\frac{(y_j-\mu_0)^2}{\sigma^2}\right) \tag{B.4}$$

$$= \frac{1}{2\sigma^2}\sum_{j=\tau}^{n}\left(-(y_j-\mu_1)^2 + (y_j-\mu_0)^2\right) \tag{B.5}$$

$$S_\tau^n = \frac{1}{2\sigma^2}\sum_{j=\tau}^{n}\left(2y_j\left(\mu_1-\mu_0\right)+\mu_0^2-\mu_1^2\right) \tag{B.6}$$

$$= \frac{1}{2\sigma^2}\left(\left(n-\tau+1\right)\left(\mu_0^2-\mu_1^2\right)+2\left(\mu_1-\mu_0\right)\sum_{j=\tau}^{n}y_j\right) \tag{B.7}$$

Les observations y_j appartenant à l'intervalle $[\![\tau, n]\!]$ suivent l'hypothèse H_1, il en résulte que :

$$\sum_{j=\tau}^{n} y_j = (n-\tau+1)\mu_1 \tag{B.8}$$

d'où

$$S_\tau^n = \frac{1}{2\sigma^2}\left(\left(n-\tau+1\right)\left(\mu_0^2-\mu_1^2\right)+2\left(\mu_1-\mu_0\right)\left(n-\tau+1\right)\mu_1\right) \tag{B.9}$$

$$= \frac{\left(n-\tau+1\right)}{2\sigma^2}\left(\mu_0^2-\mu_1^2+2\mu_1^2-2\mu_0\mu_1\right) \tag{B.10}$$

$$= \frac{\left(n-\tau+1\right)}{2\sigma^2}\left(\mu_0^2-2\mu_0\mu_1+\mu_1^2\right) \tag{B.11}$$

L'équation finale de la somme des ratios de vraisemblance logarithmiques dans le cas gaussien unidimensionnel est donc :

$$S_\tau^n = \frac{\left(n-\tau+1\right)\left(\mu_1-\mu_0\right)^2}{2\sigma^2} \tag{B.12}$$

Annexe C

Détermination de l'équation de la somme des ratios de vraisemblance logarithmiques dans le cas gaussien multidimensionnel

L'équation non simplifiée de la somme des ratios de vraisemblance logarithmiques dans le cas gaussien multidimensionnel (*cf.* section 7.3.1) est :

$$S_\tau^n = \sum_{j=\tau}^{n} ln \left(\frac{\frac{1}{(2\pi)^{k/2}|\Sigma_1|^{1/2}} e^{-\frac{1}{2}(y_j-\mu_1)^T \Sigma_1^{-1}(y_j-\mu_1)}}{\frac{1}{(2\pi)^{k/2}|\Sigma_0|^{1/2}} e^{-\frac{1}{2}(y_j-\mu_0)^T \Sigma_0^{-1}(y_j-\mu_0)}} \right) \tag{C.1}$$

Pour obtenir l'équation finale, il s'agit de simplifier l'équation précédente.

$$\begin{aligned} S_\tau^n =& \sum_{j=\tau}^{n} \left(\frac{1}{2} ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) - \frac{1}{2} (y_j - \mu_1)^T \Sigma_1^{-1} (y_j - \mu_1) + \frac{1}{2} (y_j - \mu_0)^T \Sigma_0^{-1} (y_j - \mu_0) \right) && \text{(C.2)} \\ =& \frac{n-\tau+1}{2} ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) + \\ & \sum_{j=\tau}^{n} \left(-\frac{1}{2} (y_j - \mu_1)^T \Sigma_1^{-1} (y_j - \mu_1) + \frac{1}{2} (y_j - \mu_0)^T \Sigma_0^{-1} (y_j - \mu_0) \right) && \text{(C.3)} \end{aligned}$$

Étant donnée la taille de l'équation, posons $X_k = \sum_{j=\tau}^{n} (y_j - \mu_k)^T \Sigma_k^{-1} (y_j - \mu_k)$ avec $k = \{0, 1\}$. L'équation s'écrit alors :

$$S_\tau^n = \frac{n-\tau+1}{2} ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) - \frac{1}{2} X_1 + \frac{1}{2} X_0 \tag{C.4}$$

La simplification de l'expression de X_k donne :

$$X_k = \sum_{j=\tau}^{n} \left(y_j^T \Sigma_k^{-1} y_j - \mu_k^T \Sigma_k^{-1} y_j - y_j^T \Sigma_k^{-1} \mu_k + \mu_k^T \Sigma_k^{-1} \mu_k \right) \quad \text{(C.5)}$$

$$= \sum_{j=\tau}^{n} \left(y_j^T \Sigma_k^{-1} y_j \right) - \mu_k^T \Sigma_k^{-1} \sum_{j=\tau}^{n} y_j - \left(\sum_{j=\tau}^{n} y_j^T \right) \Sigma_k^{-1} \mu_k + (n - \tau + 1)\, \mu_k^T \Sigma_k^{-1} \mu_k \quad \text{(C.6)}$$

Comme pour le cas unidimensionnel, $\sum_{j=\tau}^{n} y_j = (n - \tau + 1)\, \mu_1$.

$$X_k = \sum_{j=\tau}^{n} \left(y_j^T \Sigma_k^{-1} y_j \right) + (n - \tau + 1) \left(\mu_k^T \Sigma_k^{-1} \mu_k - \mu_k^T \Sigma_k^{-1} \mu_1 - \mu_1^T \Sigma_k^{-1} \mu_k \right) \quad \text{(C.7)}$$

En remplaçant dans l'équation de départ C.4, l'expression de la somme des ratios de vraisemblance logarithmiques devient :

$$\begin{aligned} S_\tau^n =& \frac{n - \tau + 1}{2} ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) + \\ & \frac{n - \tau + 1}{2} \left(\mu_0^T \Sigma_0^{-1} \mu_0 - \mu_0^T \Sigma_0^{-1} \mu_1 - \mu_1^T \Sigma_0^{-1} \mu_0 - \mu_1^T \Sigma_1^{-1} \mu_1 + \mu_1^T \Sigma_1^{-1} \mu_1 + \mu_1^T \Sigma_1^{-1} \mu_1 \right) + \\ & \frac{1}{2} \sum_{j=\tau}^{n} \left(y_j^T \Sigma_0^{-1} y_j \right) - \frac{1}{2} \sum_{j=\tau}^{n} \left(y_j^T \Sigma_1^{-1} y_j \right) \end{aligned} \quad \text{(C.8)}$$

$$\begin{aligned} S_\tau^n =& \frac{n - \tau + 1}{2} ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) + \frac{n - \tau + 1}{2} \left(\mu_0^T \Sigma_0^{-1} \mu_0 + \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_0^T \Sigma_0^{-1} \mu_1 - \mu_1^T \Sigma_0^{-1} \mu_0 \right) + \\ & \frac{1}{2} \sum_{j=\tau}^{n} \left(y_j^T \left(\Sigma_0^{-1} - \Sigma_1^{-1} \right) y_j \right) \end{aligned} \quad \text{(C.9)}$$

De plus, $\left(\mu_0^T \Sigma_0^{-1} \mu_1 \right)^T = \mu_1^T \Sigma_0^{-T} \mu_0 = \mu_1^T \Sigma_0^{-1} \mu_0$ car Σ_0 est une matrice de covariance. Elle est donc symétrique avec $\Sigma_0^T = \Sigma_0$, et par conséquent $\Sigma_0^{-1} = \left(\Sigma_0^T \right)^{-1} = \Sigma_0^{-T}$. D'autre part, $\mu_0^T \Sigma_0^{-1} \mu_1$ et $\mu_1^T \Sigma_0^{-1} \mu_0$ sont des scalaires et la transposée d'un scalaire est le scalaire lui-même, d'où $\mu_0^T \Sigma_0^{-1} \mu_1 = \mu_1^T \Sigma_0^{-1} \mu_0$. L'équation de détection finale est alors la suivante :

$$\begin{aligned} S_\tau^n =& \frac{n - \tau + 1}{2} ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) + \frac{n - \tau + 1}{2} \left(\mu_0^T \Sigma_0^{-1} \mu_0 + \mu_1^T \Sigma_1^{-1} \mu_1 - 2 \mu_0^T \Sigma_0^{-1} \mu_1 \right) + \\ & \frac{1}{2} \sum_{j=\tau}^{n} \left(y_j^T \left(\Sigma_0^{-1} - \Sigma_1^{-1} \right) y_j \right) \end{aligned} \quad \text{(C.10)}$$

Bibliographie

H. Andreasson, A. Treptow et T. Duckett : Localization for mobile robots using panoramic vision, local features and particle filter. *In ICRA*, pages 3348–3353. IEEE, 2005. URL http://dblp.uni-trier.de/db/conf/icra/icra2005.html#AndreassonTD05.

A. Angeli : *Détection visuelle de fermeture de boucle et applications à la localisation et cartographie simultanées*. Thèse de doctorat, 2008.

A. Angeli, S. Doncieux, J.-A. Meyer et D. Filliat : Incremental vision-based topological slam. *In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1031–1036, September 2008a.

A. Angeli, S. Doncieux, J.-A. Meyer et D. Filliat : Real-time visual loop-closure detection. *In ICRA*, pages 1842–1847. IEEE, 2008b. URL http://dblp.uni-trier.de/db/conf/icra/icra2008.html#AngeliDMF08.

A. Angeli, D. Filliat, S. Doncieux et J.-A. Meyer : A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 2008c.

A. Angeli, D. Filliat, S. Doncieux et J.-A. Meyer : Visual topological slam and global localization. *In Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2009.

R. Atkinson et R. Shiffrin : Human memory : A proposed system and its control processes. *In K. W. Spence and J. T. Spence (Eds.), The Psychology of learning and motivation : Advances in research and theory (vol. 2).*, pages 89 – 105, 1968.

P. Baker, C. Fermüller, Y. Aloimonos et R. Pless : A spherical eye from multiple cameras (makes better models of the world). *In CVPR (1)*, pages 576–583. IEEE Computer Society, 2001. ISBN 0-7695-1272-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2001-1.html#BakerFAP01.

T. Barfoot : Online visual motion estimation using fastslam with sift features. *In IROS*, pages 579–585. IEEE, 2005. ISBN 0-7803-8912-3. URL http://dblp.uni-trier.de/db/conf/iros/iros2005.html#Barfoot05.

L. Baum, T. Petrie, G. Soules et N. Weiss : A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41 (1):164–171, 1970. ISSN 00034851. URL http://dx.doi.org/10.2307/2239727.

H. Bay, T. Tuytelaars et L. Van Gool : Surf : Speeded up robust features. *In Computer Vision – ECCV 2006*, Lecture Notes in Computer Science, chapitre 32, pages 404–417. 2006. URL http://dx.doi.org/10.1007/11744023_32.

J. Bentley : Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, septembre 1975. ISSN 0001-0782. URL http://doi.acm.org/10.1145/361002.361007.

I. Biederman : Recognition-by-components : A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

J. L. Blanco, J. González et J. A. Fernández-Madrigal : Subjective local maps for hybrid metric-topological SLAM. *Robotics and Autonomous Systems*, 57(1):64–74, 2009.

O. Booij, B. Terwijn, Z. Zivkovic et B. Kröse : Navigation using an appearance based topological map. *In ICRA*, pages 3927–3932. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/icra/icra2007.html#BooijTZK07.

M. Bosse, P. M. Newman, J. J. Leonard et S. Teller : SLAM in Large-scale Cyclic Environments using the Atlas Framework. *The International Journal of Robotics Research*, 23(12):1113–1139, December 2004.

J. Bouguet : Matlab camera calibration toolbox. 2000.

T. Bülow : Multiscale image processing on the sphere. *In* Luc J. Van Gool, éditeur : *DAGM-Symposium*, volume 2449 de *Lecture Notes in Computer Science*, pages 609–617. Springer, 2002. ISBN 3-540-44209-X. URL http://dblp.uni-trier.de/db/conf/dagm/dagm2002.html#Bulow02.

T. Bülow et K. Daniilidis : Surface representations using spherical harmonics and gabor wavelets on the sphere, 2001.

A. Chapoulie, P. Rives et D. Filliat : A spherical representation for efficient visual loop closing. *In Proceedings of the 11th workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS 2011)*, 2011.

A. Chapoulie, P. Rives et D. Filliat : Topological segmentation of indoors/outdoors sequences of spherical views. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'12*, 2012.

R. Chatila et J. Laumond : Position referencing and consistent world modeling for mobile robots. *In Robotics and Automation. Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 138 – 145, mar 1985.

H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch et S. Thrun : *Principles of Robot Motion : Theory, Algorithms, and Implementation*. MIT Press, June 2005. URL http://mitpress.mit.edu/catalog/item/default.asp?ttype=2\&tid=10340.

A. I. Comport, E. Malis et P. Rives : Accurate quadrifocal tracking for robust 3d visual odometry. *In Robotics and Automation, 2007 IEEE International Conference on*, pages 40–45, 2007. URL http://dx.doi.org/10.1109/ROBOT.2007.363762.

G. Csurka, C. Dance, L. Fan, J. Willamowski et C. Bray : Visual categorization with bags of keypoints. *In In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

M. Cummins et P. Newman : Probabilistic appearance based navigation and loop closing. *In ICRA*, pages 2042–2048. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/icra/icra2007.html#CumminsN07.

M. J. Cummins et P. M. Newman : Fab-map : Probabilistic localization and mapping in the space of appearance. *I. J. Robotic Res.*, 27(6):647–665, 2008. URL http://dblp.uni-trier.de/db/journals/ijrr/ijrr27.html#CumminsN08.

F. Dayoub et T. Duckett : An adaptive appearance-based map for long-term topological localization of mobile robots. *In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3364 –3369, sept. 2008.

F. Dayoub, T. Duckett et G. Cielniak : An adaptive spherical view representation for navigation in changing environments. *European Conference on Mobile Robots*, 2009.

F. Dellaert, D. Fox, W. Burgard et S. Thrun : Monte carlo localization for mobile robots. *In IEEE International Conference on Robotics and Automation (ICRA99)*, May 1999.

A. Dempster : Upper and lower probabilities induced by a multivalued mapping. 1967. URL http://dblp.uni-trier.de/db/series/sfsc/sfsc219.html#Dempster08a.

M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg et C. Schmid : Evaluation of gist descriptors for web-scale image search. *In CIVR '09 : Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-480-5. URL http://dx.doi.org/10.1145/1646396.1646421.

H. Durrant-Whyte : Consistent integration and propagation of disparate sensor observations. *In Robotics and Automation. Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1464 – 1469, apr 1986.

E. Eade et T. Drummond : Scalable monocular slam. *In CVPR (1)*, pages 469–476. IEEE Computer Society, 2006. ISBN 0-7695-2597-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2006-1.html#EadeD06.

P. Elinas, R. Sim et J. Little : Stereo vision slam using the rao-blackwellised particle filter and a novel mixture proposal distribution. *In ICRA*, pages 1564–1570. IEEE, 2006. URL http://dblp.uni-trier.de/db/conf/icra/icra2006.html#ElinasSL06.

S. P. Engelson et D. V. McDermott : Error correction in mobile robot map learning. *In Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on Robotics and Automation*, pages 2555 –2560 vol.3, may 1992.

R. Eustice, O. Pizarro et H. Singh : Visually augmented navigation in an unstructured environment using a delayed state history. *In ICRA*, pages 25–32. IEEE, 2004. URL http://dblp.uni-trier.de/db/conf/icra/icra2004-1.html#EusticePS04.

D. Filliat : A visual bag of words method for interactive qualitative localization and mapping. *In Robotics and Automation, 2007 IEEE International Conference on*, pages 3921 –3926, April 2007.

M. A. Fischler et R. C. Bolles : Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, juin 1981. ISSN 0001-0782. URL http://dx.doi.org/10.1145/358669.358692.

W. Fisher : On Grouping for Maximum Homogeneity. *Journal of the American Statistical Association*, 53(284), 1958. ISSN 01621459. URL http://dx.doi.org/10.2307/2281952.

D. Forsyth et J. Ponce : *Computer Vision : A Modern Approach*. Prentice Hall Professional Technical Reference, 2002. ISBN 0130851981.

F. Fraundorfer, C. Engels et D. Nistér : Topological mapping, localization and navigation using image collections. *In IROS*, pages 3872–3877. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/iros/iros2007.html#FraundorferEN07.

H. Friedrich, D. Dederscheck, K. Krajsek et R. Mester : View-based robot localization using spherical harmonics : Concept and first experimental results. *In* Fred A. Hamprecht, Christoph Schnörr et Bernd Jähne, éditeurs : *DAGM-Symposium*, volume 4713 de *Lecture Notes in Computer Science*, pages 21–31. Springer, 2007. ISBN 978-3-540-74933-2. URL http://dblp.uni-trier.de/db/conf/dagm/dagm2007.html#FriedrichDKM07.

H. Friedrich, D. Dederscheck, M. Mutz et R. Mester : View-based robot localization using illumination-invariant spherical harmonics descriptors. *In* Alpesh Ranchordas et Helder Araújo, éditeurs : *VISAPP (2)*, pages 543–550. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008. ISBN 978-989-8111-21-0. URL http://dblp.uni-trier.de/db/conf/visapp/visapp2008-2.html#FriedrichDMM08.

J. Gaspar, N. Winters et J. Santos-Victor : Vision-based navigation and environmental representations with an omnidirectional camera, 2000. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.1593.

C. Geyer et K. Daniilidis : A unifying theory for central panoramic systems and practical applications. *In* David Vernon, éditeur : *ECCV (2)*, volume 1843 de *Lecture Notes in Computer Science*, pages 445–461. Springer, 2000. ISBN 3-540-67686-4. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2000-2.html#GeyerD00.

T. Goedemé, M. Nuttin, T. Tuytelaars et L. Van Gool : Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236, 2007. URL http://dblp.uni-trier.de/db/journals/ijcv/ijcv74.html#GoedemeNTG07.

R. Green : Spherical Harmonic Lighting : The Gritty Details. *Archives of the Game Developers Conference*, march 2003. URL http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.pdf.

G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff et W. Burgard : Efficient estimation of accurate maximum likelihood maps in 3d. *In IROS*, pages 3472–3478. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/iros/iros2007.html#GrisettiGSPB07.

J. Gutmann et K. Konolige : Incremental mapping of large cyclic environments. *In Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on*, pages 318 –325, 1999.

C. Harris et M. Stephens : A combined corner and edge detector. *In Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

W. Hübner et H. Mallot : Metric embedding of view-graphs. *Auton. Robots*, 23(3):183–196, 2007.

P. Hébert, S. Betgé-Brezetz et R. Chatila : Probabilistic map learning : Necessity and difficulties. *In Proceedings of the International Workshop on Reasoning with Uncertainty in Robotics*, Amsterdam, Netherlands, Dec. 4-6 1995.

L. Itti et P. Baldi : A principled approach to detecting surprising events in video. *In CVPR (1)*, pages 631–637. IEEE Computer Society, 2005. ISBN 0-7695-2372-2. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2005-1.html#IttiB05`.

I. T. Jolliffe : *Principal Component Analysis*. Springer, second édition, oct 2002. ISBN 0387954422.

N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian et M. E. Munich : The vSLAM algorithm for robust localization and mapping. *In 2005 IEEE International Conf. on Robotics and Automation, ICRA 2005*, 2005.

S. Koenig et R. G. Simmons : Unsupervised learning of probabilistic models for robot navigation. *In in Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2301–2308, 1996.

K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit et P. Fua : View-based maps. *In Proceedings of Robotics : Science and Systems*, Seattle, USA, June 2009.

J. Košecká, F. Li et X. Yang : Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27 – 38, 2005. ISSN 0921-8890. URL `http://www.sciencedirect.com/science/article/pii/S092188900500062X`. ¡ce :title¿Advances in Robot Vision¡/ce :title¿.

G. Krishnan et S. K. Nayar : Towards A True Spherical Camera. *In SPIE Human Vision and Electronic Imaging*, January 2009.

B. Kröse, N. Vlassis et R. Bunschoten : Omnidirectional vision for appearance-based robot localization. *In* Gregory D. Hager, Henrik I. Christensen, Horst Bunke et Rolf Klein, éditeurs : *Sensor Based Intelligent Robots*, volume 2238 de *Lecture Notes in Computer Science*, pages 39–50. Springer, 2000. ISBN 3-540-43399-6. URL `http://dblp.uni-trier.de/db/conf/dagstuhl/sbir2000.html#KroseVB00`.

A. Kumar, J.-P. Tardif, R. Anati et K. Daniilidis : Experiments on visual loop closing using vocabulary trees. *In Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –8, June 2008.

T. Lemaire, C. Berger, I. Jung et S. Lacroix : Vision-based slam : Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343–364, 2007. URL `http://dblp.uni-trier.de/db/journals/ijcv/ijcv74.html#LemaireBJL07`.

T. Lemaire et S. Lacroix : Long term slam with panoramic vision. *Journal of Field Robotics*, 24:91–111, 2007.

J. Leonard et D. Whyte : Simultaneous map building and localization for an autonomous mobile robot. *In IEEE International Conference on Intelligent Robot Systems, Osaka, Japan*, 1991.

V. Lepetit et P. Fua : Keypoint recognition using randomized trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1465–1479, 2006. URL http://dblp.uni-trier.de/db/journals/pami/pami28.html#LepetitF06.

S. Li : Full-view spherical image camera. *In ICPR (4)*, pages 386–390. IEEE Computer Society, 2006. ISBN 0-7695-2521-0. URL http://dblp.uni-trier.de/db/conf/icpr/icpr2006-4.html#Li06b.

T. Lindeberg : Scale-space theory : A basic tool for analysing structures at different scales. *J. of Applied Statistics*, 21(2):224–270, 1994. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.4689.

S. Lovegrove et A. Davison : Real-time spherical mosaicing using whole image alignment. *In* Kostas Daniilidis, Petros Maragos et Nikos Paragios, éditeurs : *ECCV (3)*, volume 6313 de *Lecture Notes in Computer Science*, pages 73–86. Springer, 2010. ISBN 978-3-642-15557-4. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2010-3.html#LovegroveD10.

D. Lowe : Object recognition from local scale-invariant features. *Computer Vision, IEEE International Conference on*, 2:1150–1157 vol.2, August 1999. URL http://dx.doi.org/10.1109/ICCV.1999.790410.

D. Lowe : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91, November 2004.

F. Lu et E. Milios : Robot pose estimation in unknown environments by matching 2d range scans. *In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 935 –938, jun 1994.

F. Lu et E. Milios : Globally consistent range scan alignment for environment mapping. *Auton. Robots*, 4(4):333–349, octobre 1997. ISSN 0929-5593. URL http://dx.doi.org/10.1023/A:1008854305733.

J. Luo, A. Pronobis, B. Caputo et P. Jensfelt : Incremental learning for place recognition in dynamic environments. *In IROS*, pages 721–728. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/iros/iros2007.html#LuoPCJ07.

T. MacRobert : *Spherical Harmonics, an Elementary Treatise on Harmonic Functions, with Application,...* Dover, 1948. URL http://books.google.fr/books?id=8TBLPgAACAAJ.

R. V. Martín, P. Núñez et A. Bandera : Less-mapping : Online environment segmentation based on spectral mapping. *Robotics and Autonomous Systems*, 60(1):41–54, 2012. URL http://dblp.uni-trier.de/db/journals/ras/ras60.html#MartinNB12.

J. Matas, O. Chum, M. Urban et T. Pajdla : Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.*, 22(10):761–767, 2004. URL http://dblp.uni-trier.de/db/journals/ivc/ivc22.html#MatasCUP04.

C. Mei et P. Rives : Single view point omnidirectional camera calibration from planar grids. *In ICRA*, pages 3945–3950. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/icra/icra2007.html#MeiR07.

M. Meilland : *Cartographie RGB-D dense pour la localisation visuelle temps-réel et la navigation autonome*. Thèse de doctorat, École nationale supérieure des mines de Paris, 2012.

M. Meilland, A.I. Comport et P. Rives : A Spherical Robot-Centered Representation for Urban Navigation. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'10*, Taipei, Taiwan, October 2010. URL `http://www.i3s.unice.fr/~comport/publications/2010_IROS_Meilland.pdf`.

M. Meilland, A.I. Comport et P. Rives : Dense visual mapping of large scale environments for real-time localisation. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, California, September 2011. URL `http://www.i3s.unice.fr/~comport/publications/2011_IROS_Meilland.pdf`.

E. Menegatti, M. Zoccarato, E. Pagello et H. Ishiguro : Image-based monte carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004. URL `http://dblp.uni-trier.de/db/journals/ras/ras48.html#MenegattiZPI04`.

K. Mikolajczyk et C. Schmid : A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1498756`.

M. Montemerlo, S. Thrun, D. Koller et B. Wegbreit : Fastslam 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *In* Georg Gottlob et Toby Walsh, éditeurs : *IJCAI*, pages 1151–1156. Morgan Kaufmann, 2003. URL `http://dblp.uni-trier.de/db/conf/ijcai/ijcai2003.html#MontemerloTKW03`.

G. Mori, X. Ren, A. Efros et J. Malik : Recovering human body configurations : Combining segmentation and recognition. *In CVPR (2)*, pages 326–333, 2004. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2004-2.html#MoriREM04`.

A. C. Murillo et J. Košecká : Experiments in place recognition using gist panoramas. *In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 2196 –2203, October 2009.

S. K. Nayar : Catadioptric omnidirectional camera. pages 482 –488, June 1997.

P. Newman, D. Cole et K. Ho : Outdoor slam using visual appearance and laser ranging. *In ICRA*, pages 1180–1187. IEEE, 2006. URL `http://dblp.uni-trier.de/db/conf/icra/icra2006.html#NewmanCH06`.

J. Neyman et E. Pearson : On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933. ISSN 02643952. URL `http://www.jstor.org/stable/91247`.

M.-E. Nilsback et A. Zisserman : A visual vocabulary for flower classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

D. Nistér et H. Stewénius : Scalable recognition with a vocabulary tree. *In CVPR (2)*, pages 2161–2168. IEEE Computer Society, 2006. ISBN 0-7695-2597-0. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2006-2.html#NisterS06`.

A. Oliva et A. Torralba : Modeling the shape of the scene : A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, May 2001. ISSN 09205691. URL http://dx.doi.org/10.1023/A:1011139631724.

A. Oliva et A. Torralba : Building the gist of a scene : the role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006. ISSN 0079-6123. URL http://dx.doi.org/10.1016/S0079-6123(06)55002-2.

E. Page : Continuous Inspection Schemes. *Biometrika*, 41(1/2):100–115, 1954. ISSN 00063444. URL http://dx.doi.org/10.2307/2333009.

E. Parzen : On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):pp. 1065–1076, 1962. ISSN 00034851. URL http://www.jstor.org/stable/2237880.

J. Pfeil, K. Hildebrand, C. Gremzow, B. Bickel et M. Alexa : Throwable panoramic ball camera. *In SIGGRAPH Asia 2011 Emerging Technologies*, SA '11, pages 4 :1–4 :1, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1136-6. URL http://doi.acm.org/10.1145/2073370.2073373.

A. Pronobis et B. Caputo : Confidence-based cue integration for visual place recognition. *In IROS*, pages 2394–2401. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/iros/iros2007.html#PronobisC07.

A. Pronobis, B. Caputo, P. Jensfelt et H. Christensen : A discriminative approach to robust visual place recognition. *In IROS*, pages 3829–3836. IEEE, 2006. URL http://dblp.uni-trier.de/db/conf/iros/iros2006.html#PronobisCJC06a.

M. Pupilli et A. Calway : Real-time visual slam with resilience to erratic motion. *In CVPR (1)*, pages 1244–1249. IEEE Computer Society, 2006. ISBN 0-7695-2597-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2006-1.html#PupilliC06.

A. Ranganathan : PLISS : Detecting and Labeling Places Using Online Change-Point Detection. *In Robotics : Science and Systems VI*, 2010. URL http://www.roboticsproceedings.org/rss06/p24.html.

A. Ranganathan et F. Dellaert : Bayesian surprise and landmark detection. *In ICRA*, pages 2017–2023. IEEE, 2009. URL http://dblp.uni-trier.de/db/conf/icra/icra2009.html#RanganathanD09.

A. Ranganathan, E. Menegatti et F. Dellaert : Bayesian inference in the space of topological maps. *IEEE Transactions on Robotics*, 22:92–107, 2006.

X. Ren et J. Malik : Learning a classification model for segmentation. *In ICCV*, pages 10–17. IEEE Computer Society, 2003. ISBN 0-7695-1950-4. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2003-1.html#RenM03.

S. Roberts : Control chart tests based on geometric moving averages. 1959.

E. Rosten et T. Drummond : Machine learning for high-speed corner detection. *In* Ales Leonardis, Horst Bischof et Axel Pinz, éditeurs : *ECCV (1)*, volume 3951 de *Lecture Notes in Computer Science*, pages 430–443. Springer, 2006. ISBN 3-540-33832-2. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2006-1.html#RostenD06.

V. Schönefeld : Spherical harmonics, July 2005.

S. Se, D. Lowe et J. Little : Global localization using distinctive visual features. *In Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 226 – 231 vol.1, 2002.

G. Shafer : *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

H. Shatkay et L. P. Kaelbling : Learning topological maps with weak local odometric information. *In Proceedings of the Fifteenth international joint conference on Artifical intelligence - Volume 2*, pages 920–927, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-555860-480-4. URL http://portal.acm.org/citation.cfm?id=1622289.

W. Shewart : *Economic control of Quality of Manufactured Product*. Van Nostrand Reinhold Co., New York, 1931.

B. Silverman : *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, April 1986. ISBN 0412246201.

R. Sim et G. Dudek : Learning and evaluating visual features for pose estimation. *In ICCV*, pages 1217–1222, 1999. URL http://dblp.uni-trier.de/db/conf/iccv/iccv1999-2.html#SimD99.

R. Sim et G. Dudek : Learning generative models of invariant features. *In in Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS*, pages 3481–3488, 2004.

R. Sim, M. Griffin, A. Shyr et J. Little : Scalable real-time visionbased slam for planetary rovers. *In in IEEE IROS Workshop on Robot Vision for Space Applications, IEEE*, pages 16–21. IEEE Press, 2005.

G. Singh et J. Košecká : Visual loop closing using gist descriptors in manhattan world. 2010.

J. Sivic et A. Zisserman : Video google : A text retrieval approach to object matching in videos. *In ICCV*, pages 1470–1477. IEEE Computer Society, 2003. ISBN 0-7695-1950-4. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2003-2.html#SivicZ03.

R. Smith et P. Cheeseman : On the Representation and Estimation of Spatial Uncertainty, 1986.

R. Smith, M. Self et P. Cheeseman : Estimating uncertain spatial relationships in robotics. *In Robotics and Automation. Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, page 850, mar 1987.

S. Smith et J. Brady : Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997. URL http://dblp.uni-trier.de/db/journals/ijcv/ijcv23.html#SmithB97.

T. Svoboda, D. Martinec et T. Pajdla : A convenient multicamera self-calibration for virtual environments. *Presence : Teleoper. Virtual Environ.*, 14(4):407–422, août 2005. ISSN 1054-7460. URL http://dx.doi.org/10.1162/105474605774785325.

R. Szeliski : Image alignment and stitching : a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2 (1):1–104, 2006. ISSN 1572-2740.

T. T. Tanimoto : IBM Internal Report, 1957.

A. Tapus et R. Siegwart : Incremental Robot Mapping with Fingerprints of Places. *In International Conference on Intelligent Robots and Systems*, pages 172–177, 2005. URL http://dx.doi.org/10.1109/IROS.2005.1544977.

S. Thrun, J. Gutmann, D. Fox, W. Burgard et B. Kuipers : Integrating topological and metric maps for mobile robot navigation : A statistical approach. *In In Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

E. Tola, V. Lepetit et P. Fua : Daisy : An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1), 2009. ISSN 0162-8828. URL http://dx.doi.org/10.1109/TPAMI.2009.77.

G. Tsechpenakis, D. N. Metaxas, C. Neidle et O. Hadjiliadis : Robust online change-point detection in video sequences. *In Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, CVPRW '06, pages 155–, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2646-2. URL http://dx.doi.org/10.1109/CVPRW.2006.176.

T. Tuytelaars et K. Mikolajczyk : *Local Invariant Feature Detectors : A Survey*. Now Publishers Inc., Hanover, MA, USA, 2008. ISBN 1601981384, 9781601981387. URL http://portal.acm.org/citation.cfm?id=1481563.

I. Ulrich et I. Nourbakhsh : Appearance-based place recognition for topological localization. *In ICRA'00*, pages 1023–1029, 2000.

C. Valgren, T. Duckett et A. Lilienthal : Incremental spectral clustering and its application to topological mapping. *In ICRA*, pages 4283–4288. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/icra/icra2007.html#ValgrenDL07.

C. Valgren et A. Lilienthal : SIFT, SURF and seasons : Long-term outdoor localization using local features. *In Proceedings of the European Conference on Mobile Robots (ECMR)*, pages 253–258, September 19–21 2007.

P. Viola et M. Jones : Rapid object detection using a boosted cascade of simple features. *In CVPR (1)*, pages 511–518. IEEE Computer Society, 2001. ISBN 0-7695-1272-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2001-1.html#ViolaJ01.

J. Wang, H. Zha et R. Cipolla : Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(2):413–422, 2006. URL http://dblp.uni-trier.de/db/journals/tsmc/tsmcb36.html#WangZC06.

C. Weiss, H. Tamimi, A. Masselli et A. Zell : A hybrid approach for vision-based outdoor robot localization using global and local image features. *In IROS*, pages 1047–1052. IEEE, 2007. URL http://dblp.uni-trier.de/db/conf/iros/iros2007.html#WeissTMZ07.

B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid et J. Tardos : An image-to-map loop closing method for monocular slam. *In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2053 –2059, sept. 2008.

B. Williams, G. Klein et I. Reid : Real-time slam relocalisation. *In ICCV'07*, pages 1–8, 2007a.

B. Williams, P. Smith et I. Reid : Automatic relocalisation for a single-camera simultaneous localisation and mapping system. *In Robotics and Automation, 2007 IEEE International Conference on*, pages 2784 –2790, april 2007b.

J. Wolf, W. Burgard et H. Burkhardt : Robust vision-based localization by combining an image-retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005. URL `http://dblp.uni-trier.de/db/journals/trob/trob21.html#WolfBB05`.

G. Yu et J.-M. Morel : A fully affine invariant image comparison method. *In ICASSP '09 : Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1597–1600, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-1-4244-2353-8.

A. Zaharescu, R. Horaud, R. Ronfard et L. Lefort : Multiple camera calibration using robust perspective factorization. *In 3DPVT*, pages 504–511. IEEE Computer Society, 2006. ISBN 978-0-7695-2825-0. URL `http://dblp.uni-trier.de/db/conf/3dpvt/3dpvt2006.html#ZaharescuHRL06`.

Z. Zivkovic, B. Bakker et B. Krose : Hierarchical map building using visual landmarks and geometric constraints. *In Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2480 – 2485, August 2005.

Z. Zivkovic, O. Booij et B. J. A. Kröse : From images to rooms. *Robotics and Autonomous Systems*, 55(5):411–418, 2007. URL `http://dblp.uni-trier.de/db/journals/ras/ras55.html#ZivkovicBK07`.

Résumé :

Dans le contexte de la localisation globale et, plus largement, dans celui de la Localisation et Cartographie Simultanées, il est nécessaire de pouvoir déterminer si un robot revient dans un endroit déjà visité. Il s'agit du problème de la détection de fermeture de boucle. Dans un cadre de reconnaissance visuelle des lieux, les algorithmes existants permettent une détection en temps-réel, une robustesse face à l'aliasing perceptuel ou encore face à la présence d'objets dynamiques. Ces algorithmes sont souvent sensibles à l'orientation du robot rendant impossible la fermeture de boucle à partir d'un point de vue différent. Pour pallier ce problème, des caméras panoramiques ou omnidirectionnelles sont employées. Nous présentons ici une méthode plus générale de représentation de l'environnement sous forme d'une vue sphérique ego-centrée. En utilisant les propriétés de cette représentation, nous proposons une méthode de détection de fermeture de boucle satisfaisant, en plus des autres propriétés, une indépendance à l'orientation du robot.

Le modèle de l'environnement est souvent un ensemble d'images prises à des instants différents, chaque image représentant un lieu. Afin de grouper ces images en lieux significatifs de l'environnement, des lieux topologiques, les méthodes existantes emploient une notion de covisibilité de l'information entre les lieux. Notre approche repose sur l'exploitation de la structure de l'environnement. Nous définissons ainsi un lieu topologique comme ayant une structure qui ne varie pas, la variation engendrant le changement de lieu. Les variations de structure sont détectées à l'aide d'un algorithme efficace de détection de rupture de modèle.

Abstract :

In the context of global localization and, more widely, in Simultaneous Localization And Mapping, it is mandatory to be able to detect if a robot comes to a previously visited place. It is the loop closure detection problem. Algorithms, in visual place recognition, usually allow detection in real-time, are robust to perceptual aliasing or even to dynamic objects. Those algorithms are often sensitive to the robot orientation involving an impossibility to detect a loop closure from a different point of view. In order to alleviate this drawback, panoramic or omnidirectional cameras are often used. We propose a more general representation of the environment with an ego-centric spherical view. Using this representation properties, we elaborate a loop closure detection algorithm that satisfies, in addition to other properties, a robot orientation independence.

The environment model is often a set of images taken at various moments, each image corresponding to a place. Existing methods cluster those images in meaning places of the environment, the topological places, using the concept of covisibility of information between places. Our approach relies on the utilization of the environment structure. We hence define a topological place as having a structure which does not change, variation leading to a place change. The structure variations are detected with an efficient change-point detection algorithm.